Lecture Notes in Electrical Engineering 736

Sabu M. Thampi · Erol Gelenbe ·
Mohammed Atiquzzaman ·
Vipin Chaudhary ·
Kuan-Ching Li *Editors*

Advances in Computing and Network Communications

Proceedings of CoCoNet 2020, Volume 2

Springer

Lecture Notes in Electrical Engineering

Volume 736

The book series *Lecture Notes in Electrical Engineering* (LNEE) publishes the latest developments in Electrical Engineering - quickly, informally and in high quality. While original research reported in proceedings and monographs has traditionally formed the core of LNEE, we also encourage authors to submit books devoted to supporting student education and professional training in the various fields and applications areas of electrical engineering. The series cover classical and emerging topics concerning:

- Communication Engineering, Information Theory and Networks
- Electronics Engineering and Microelectronics
- Signal, Image and Speech Processing
- Wireless and Mobile Communication
- Circuits and Systems
- Energy Systems, Power Electronics and Electrical Machines
- Electro-optical Engineering
- Instrumentation Engineering
- Avionics Engineering
- Control Systems
- Internet-of-Things and Cybersecurity
- Biomedical Devices, MEMS and NEMS

For general information about this book series, comments or suggestions, please contact leontina.dicecco@springer.com.

To submit a proposal or request further information, please contact the Publishing Editor in your country:

China

Jasmine Dou, Editor (jasmine.dou@springer.com)

India, Japan, Rest of Asia

Swati Meherishi, Editorial Director (Swati.Meherishi@springer.com)

Southeast Asia, Australia, New Zealand

Ramesh Nath Premnath, Editor (ramesh.premnath@springernature.com)

USA, Canada:

Michael Luby, Senior Editor (michael.luby@springer.com)

All other Countries:

Leontina Di Cecco, Senior Editor (leontina.dicecco@springer.com)

**** This series is indexed by EI Compendex and Scopus databases. ****

More information about this series at http://www.springer.com/series/7818

Sabu M. Thampi · Erol Gelenbe ·
Mohammed Atiquzzaman · Vipin Chaudhary ·
Kuan-Ching Li
Editors

Advances in Computing and Network Communications

Proceedings of CoCoNet 2020, Volume 2

Springer

Editors
Sabu M. Thampi
School of CSE, Indian Institute
of Information Technology
and Management-Kerala
Trivandrum, Kerala, India

Mohammed Atiquzzaman
School of Computer Science
University of Oklahoma
Norman, OK, USA

Kuan-Ching Li
Department of Computer Science
and Information Engineering
Providence University
Taichung, Taiwan

Erol Gelenbe
Institute of Theoretical and Applied
Informatics
Polish Academy of Sciences
Gliwice, Poland

Vipin Chaudhary
Department of Computer Science
University at Buffalo
State University
Buffalo, NY, USA

Case Western Reserve University
Cleveland, OH, USA

ISSN 1876-1100 ISSN 1876-1119 (electronic)
Lecture Notes in Electrical Engineering
ISBN 978-981-33-6986-3 ISBN 978-981-33-6987-0 (eBook)
https://doi.org/10.1007/978-981-33-6987-0

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Preface

The Fourth International Conference on Computing and Network Communications (CoCoNet'20) provided a forum for sharing original research outcomes and practical development experiences among experts in the emerging areas of computing and communications. This conference was organized by Vellore Institute of Technology (VIT), Chennai, India. Due to the recent pandemic situation, the conference was conducted as a virtual event during October 14–17, 2020.

The material was presented in a program that consisted of keynote talks, technical sessions, lightning talks, tutorials, symposiums, parallel sessions, workshops and hot off the press. CoCoNet'20 convened a well-tailored and handpicked collection of eminent speakers from renowned universities and industries in and outside India. The array of speakers included Dr. Ian T. Foster, Dr. Arumugam Nallanathan, Dr. Mohammed Atiquzzaman, Dr. Vipin Chaudhary, Dr. Dilip Krishnaswamy and Dr. Ljiljana Trajkovic. This conference received 181 submissions this year, out of which 98 papers (58 regular papers, 40 short papers) had been accepted. The papers were subjected to a rigorous review process that examined the significance, novelty and technical quality of the submission. The papers were presented in different sessions, namely the best paper sessions, regular paper sessions and short paper sessions.

The proceedings of the conference is organized into two volumes. This volume is comprised of 47 papers, and the topical sections include machine learning, visual computing and signal processing, and natural language processing. This volume also includes selected papers from two co-affiliated symposiums: the International Symposium on Computer Vision and the Internet (VisionNet'20) and the Symposium on Emerging Topics in Computing and Communications (SETCAC'20).

The success of the conference depends ultimately on the numerous people who have worked with us to plan and organize both the technical program and local arrangements. In particular, the wise advice and brilliant ideas of the program chairs, workshop and symposium chairs, and industry track chairs in the organization of the technical program are gratefully appreciated. We would like to extend our heartfelt gratitude to the organizing committee and advisory committee members.

The accomplishment of the motives of the conference is powered by the tireless efforts of many individuals. We would like to express our sincere gratitude to the TPC Chairs, TPC members and additional reviewers who shared their technical expertise and assisted us in reviewing all the submitted papers. We would like to thank the general chairs, organizing committee members, steering committee, keynote speakers, session chairs and conference attendees. We are thankful to all the authors for choosing this conference as a venue for presenting their research works. We express our wholehearted appreciation for the contributions of all those who apportioned their valuable time for the success of CoCoNet'20.

We are grateful to Vellore Institute of Technology (VIT), Chennai, for organizing this conference. Recognition should go to the local organizing committee members who all have worked extremely hard for the details of important aspects of the conference programs. We appreciate the contributions of all the faculty and staff of VIT and the student volunteers who have selflessly contributed their time to make this virtual conference successful. We would like to express our gratitude to the senior editor of Springer Nature, Aninda Bose, for his help and cooperation.

We sincerely hope that the CoCoNet'20 turned out to be a forum for excellent discussions that enabled new ideas to come about, promoting collaborative research. We are confident that the proceedings will serve as a momentous source of research references and knowledge, which will lead not only to the scientific and engineering findings but also to the development of new products and technologies.

Trivandrum, India Sabu M. Thampi
Gliwice, Poland Erol Gelenbe
Norman, USA Mohammed Atiquzzaman
Buffalo\Cleveland, USA Vipin Chaudhary
Taichung, Taiwan Kuan-Ching Li
October 2020

Organized by

Vellore Institute of Technology (VIT), Chennai, India

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Conference Organization

Chief Patron

Dr. G. Viswanathan, Chancellor, VIT

Patrons

Sankar Viswanathan, Vice-President, Vellore Institute of Technology
Sekar Viswanathan, Vice-President, Vellore Institute of Technology
G. V. Selvam, Vice-President, Vellore Institute of Technology
Sandhya Pentareddy, Executive Director, Vellore Institute of Technology
Kadhambari S. Viswanathan, Assistant Vice-President, Vellore Institute of Technology
Rambabu Kodali, Vice Chancellor, Vellore Institute of Technology

S. Narayanan, Pro-VC, Vellore Institute of Technology, Vellore
V. S. Kanchana Bhaaskaran, Pro-VC, Vellore Institute of Technology, Chennai
P. K. Manoharan, Additional Registrar, VIT, Chennai

Honorary General Chair

Raj Jain, Barbara J. and Jerome R. Cox, Jr., Professor of Computer Science and Engineering, Washington University in St. Louis, USA

General Chairs

Erol Gelenbe, Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Poland
Jayanta Mukhopadhyay, Indian Institute of Technology, Kharagpur, India
Jagadeesh Kannan R., VIT, Chennai

General Executive Chair

Sabu M. Thampi, Indian Institute of Information Technology and Management-Kerala (IIITM-K), Trivandrum

Program Chairs

Mohammed Atiquzzaman, University of Oklahoma, USA
Vipin Chaudhary, University at Buffalo (UB), SUNY, USA
Kuan-Ching Li, Providence University, Taiwan
Peter Mueller, IBM Zurich Research Laboratory, Switzerland

Workshop and Symposium Chairs

Al-Sakib Khan Pathan, Independent University, Bangladesh
Pradeep K. Atrey, State University of New York (SUNY), Albany, USA

Industry Track Chairs

Dilip Krishnaswamy, Reliance Industries Ltd., India
Arpan Pal, TCS Innovation Lab, Kolkata, India
Sougata Mukherjea, IBM India Research Lab, New Delhi, India
Anindita Banerjee, QuNu Labs, Bengaluru, India

Organizing Chair

Geetha S., VIT, Chennai

Organizing Secretaries

Sweetlin Hemalatha C., VIT, Chennai
Suganya G., VIT, Chennai
Kumar R., VIT, Chennai

Organizing Co-chairs

Asha S., VIT, Chennai
Pattabiraman R., VIT, Chennai
Viswanathan V., VIT, Chennai

Tutorial Chairs

Domenico Ciuonzo, University of Naples Federico II, Italy
Maheshkumar H. Kolekar, Indian Institute of Technology Patna, India
Pascal Lorenz, University of Haute Alsace, France

Demo/Poster Chair

Sergey Mosin, Kazan Federal University, Russia

Advisory Committee

Mukesh Mohania, Indraprastha Institute of Information Technology, Delhi (IIIT-D), India
Oge Marques, Florida Atlantic University (FAU) (Boca Raton, Florida), USA
Ajith Abraham, Machine Intelligence Research Labs (MIR Labs), USA
Axel Sikora, University of Applied Sciences Offenburg, Germany
Madhukar Pitke, Professor (Retired) at Tata Institute of Fundamental Research, Mumbai
Selwyn Piramuthu, University of Florida, USA
Juan Manuel Corchado Rodríguez, University of Salamanca, Spain
Mauro Conti, University of Padua, Italy
Nallanathan Arumugam, King's College London, UK
Raj Kumar Buyya, University of Melbourne, Australia
Sameep Mehta, IBM Research—India
Bharat Bhargava, Purdue University, USA
Debabrata Das, International Institute of Information Technology, Bangalore
Schahram Dustdar, The TU Wien, Austria
Sherali Zeadally, University of Kentucky, USA
V. N. Venkatakrishnan, University of Illinois at Chicago, USA
Jorge Sá Silva, Universidade de Coimbra, Portugal
Jiankun Hu, University of New South Wales, Australia
Bharat Jayaraman, University at Buffalo, The State University of New York, USA
Rajendra Boppana, The University of Texas at San Antonio (UTSA), USA
Jalel Ben-Othman, University of Paris 13, France
Stefan Fischer, University of Luebeck, Germany
Shekar Babu, Amrita Vishwa Vidyapeetham, Bangalore

TPC Members

http://coconet-conference.org/2020/?q=committee

Contents

Machine Learning, Visual Computing and Signal Processing

Natural Language Processing

Seventh International Symposium on Computer Vision and the Internet (VisionNet'20)

Symposium on Emerging Topics in Computing and Communications (SETCAC'20)

About the Editors

Sabu M. Thampi is Professor at the Indian Institute of Information Technology and Management-Kerala (IIITM-K), Technopark Campus, Trivandrum, India. His current research interests include cognitive computing, Internet of Things (IoT), authorship analysis, trust management, biometrics, social networks, nature-inspired computing and video surveillance. He has published papers in book chapters, journals and conference proceedings. He has authored and edited a few books. Sabu has served as Guest Editor for special issues in few journals and a program committee member for many international conferences and workshops. He has co-chaired several international workshops and conferences. He has initiated and is also involved in the organization of several annual conferences/symposiums. Sabu is currently serving as Editor for Elsevier *Journal of Network and Computer Applications* (JNCA), *Connection Science*, Taylor Francis, Associate Editor for *IEEE Access* and *International Journal of Embedded Systems*, Inderscience, UK, and Reviewer for several reputed international journals. Sabu is a senior member of IEEE and ACM. He is Founding Chair of ACM Trivandrum Professional Chapter.

Erol Gelenbe is a Turkish-French computer scientist, electronic engineer and applied mathematician who is professor in Computer-Communications at Imperial College. Known for pioneering the field of modelling and performance evaluation of computer systems and networks throughout Europe, he invented the random neural network and the eponymous G-networks. His many awards include the ACM SIGMETRICS Life-Time Achievement Award, and the in Memoriam Dennis Gabor Award of the Hungarian Academy of Sciences. Working as a foreigner everywhere, Gelenbe was born in Istanbul in 1945, to Yusuf Ali Gelenbe, a descendant of the 18th-century Ottoman mathematician Gelenbevi Ismail Efendi, and to Maria Sacchet Gelenbe from Cesiomaggiore, Belluno, Italy. After a childhood spent in Istanbul and Alexandria (Egypt), He graduated from Ankara Koleji in 1962 and the Middle East Technical University in 1966, winning the K. K. Clarke Research Award for work on "partial flux switching magnetic memory systems". Awarded a Fulbright Fellowship, he continued his studies at Polytechnic University, where he completed a master's degree and a Ph.D. thesis on "Stochastic automata with structural restrictions", under the supervision of Edward J. Smith. After graduation he joined the University of Michigan as an assistant professor. In 1972, and then on leave from Michigan, he founded the Modeling and Performance Evaluation of Computer Systems research group at INRIA (France), and was a visiting lecturer at the University of Paris 13 University. In 1971 he was elected to the second chair in Computer Science at the University of Liège, where he joined Prof. Danny Ribbens in 1973, while remaining a research director at INRIA. In 1973, he was awarded a Doctorat d'État ès Sciences Mathématiques from the Paris VI University with a thesis on "Modèlisation des systèmes informatiques", under Jacques-Louis Lions. He remained a close friend of Prof. Ribbens and of the University of Liège, and in 1979, he moved to the Paris-Sud 11 University, where he co-founded the Laboratoire de Recherche en Informatique and its Ph.D. Program, before joining Paris Descartes University in 1986 to found the Ecole des Hautes Etudes en Informatique.Gelenbe became New Jersey State Endowed Professor at the New Jersey Institute of Technology from 1991 to 1993, and from 1993 and 1998 he was chaired professor and head of Electrical and Computer Engineering at Duke University. From 1998 to 2003 at the University of Central Florida, he founded the Department (School) of Electrical Engineering and Computer Science and developed the Harris Corporation Engineering Centre. In 2003, Gelenbe joined Imperial College London as Dennis Gabor Professor in Computer and Communication Networks and Head of Intelligent Systems and Networks. In 2016 he joined Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Gliwice Poland.

Gelenbe has contributed pioneering research concerning the performance of multiprogramming computer systems, virtual memory management, data base reliability optimisation, distributed systems and network protocols. He formed, led, and trained the team that designed the commercial QNAP Computer and Network Performance Modeling Tool. He introduced the Flexsim Object Oriented approach for the simulation in manufacturing systems. He carried out some of the first work on adaptive control of computer systems, and published seminal papers on the performance optimisation of computer network protocols and on the use of diffusion approximations for network performance. He developed new product form queueing networks with negative customers and triggers known as G-networks. He also introduced a new spiked stochastic neural network model known as the random neural network, developed its mathematical solution and learning algorithms, and applied it to both engineering and biological problems. His inventions include the design of the first random access fibre-optics local area network, a patented admission control technique for ATM networks, a neural network based anomaly detector for brain magnetic resonance scans, and the cognitive packet network routing protocol to offer quality of service to users. From 1984 to 1986 he served as the Science and Technology Advisor to the French Secretary of State for Universities. He founded the ISCIS (International Symposium on Computer and Information Sciences) series of conferences that since 1986 are held annually in Turkey, the USA and Europe to bring together Turkish computer scientists with their international counterparts. According to the Mathematics Genealogy project, Gelenbe has graduated over 72 Ph.D. students, placing him in the Top50 worldwide—all time—Ph.D. supervisors in the mathematical sciences.

Mohammed Atiquzzaman (senior member, IEEE) obtained his M.S. and Ph.D. in Electrical Engineering and Electronics from the University of Manchester (UK) in 1984 and 1987, respectively. He currently holds the Edith J. Kinney Gaylord Presidential professorship in the School of Computer Science at the University of Oklahoma. Dr. Atiquzzaman is Editor-in-Chief of Journal of Networks and Computer Applications and Founding Editor-in-Chief of *Vehicular Communications* and serves/served on the editorial boards of many journals including *IEEE Communications Magazine*, *Real Time Imaging Journal*, *International Journal of Communication Networks and Distributed Systems* and *Journal of Sensor Networks* and *International Journal of Communication Systems*. He co-chaired the IEEE High Performance Switching and Routing Symposium (2003, 2011), IEEE GLOBECOM and ICC (2014, 2012, 2010, 2009, 2007, 2006), IEEE VTC (2013) and the SPIE Quality of Service over Next-Generation Data Networks Conferences (2001, 2002, 2003). He was Panel Co-Chair of INFOCOM'05, is/has been in the program committee of many conferences such as INFOCOM, GLOBECOM, ICCCN, ICCIT and Local Computer Networks and serves on the review panels at the National Science Foundation. He is Current Chair of IEEE Communication Society Technical Committee on Communications Switching and Routing. Dr. Atiquzzaman received IEEE Communication Society's Fred W. Ellersick Prize and NASA Group Achievement Award for "outstanding work to further NASA Glenn Research Center's effort in the area of Advanced Communications/Air Traffic Management's Fiber Optic Signal Distribution for Aeronautical Communications" project. He is Co-author of the book *Performance of TCP/IP over ATM Networks* and has over 270 refereed publications. His current research interests are in areas of transport protocols, wireless and mobile networks, ad hoc networks, satellite networks, power-aware networking and optical communications. His research has been funded by National Science Foundation (NSF), National Aeronautics and Space Administration (NASA), and US Air Force, Cisco and Honeywell.

Vipin Chaudhary is the Kevin J. Kranzusch Professor and Chair of the Department of Computer and Data Sciences at Case Western Reserve University. Most recently, he was a Program Director at the National Science Foundation where he was involved in many national initiatives and the Empire Innovation Professor of Computer Science and Engineering at SUNY Buffalo. He cofounded Scalable Informatics, a leading provider of pragmatic, high performance software-defined storage and compute solutions to a wide range of markets, from financial and scientific computing to research and big data analytics. From 2010 to 2013, Dr. Chaudhary was the Chief Executive Officer of Computational Research Laboratories (CRL), a wholly owned Tata Sons company, where he grew the company globally to be an HPC cloud and solutions leader before selling it to Tata Consulting Services. Prior to this, as Senior Director of Advanced Development at Cradle Technologies, Inc., he was responsible for advanced programming tools for multi-processor chips. He was also the Chief Architect at Corio Inc., which had a successful IPO in July, 2000. Dr. Chaudhary was awarded the prestigious President of India Gold Medal in 1986 at the Indian Institute of Technology (IIT) Kharagpur where he received the B.Tech.

(Hons.) degree in Computer Science and Engineering and a Ph.D. degree from The University of Texas at Austin.

Kuan-Ching Li is Professor in the Department of Computer Science and Information Engineering at Providence University, Taiwan. Dr. Li is the recipient of awards from Nvidia, AWS, Intel, Ministry of Education (MOE)/Taiwan and Ministry of Science and Technology (MOST)/Taiwan, as also distinguished chair professorships from universities in China and other countries. He has been involved actively in conferences and workshops as a program/general/steering conference chairman positions and numerous conferences and workshops as a program committee member and has organized numerous conferences related to high-performance computing and computational science and engineering. Dr. Li is Editor-in-Chief of technical publications *Connection Science* (Taylor and Francis), *International Journal of Computational Science and Engineering* (Inderscience) and *International Journal of Embedded Systems* (Inderscience), also serving a number of journal's editorial boards and guest editorships. In addition, he has been acting as Co-author and Co-editor of several technical professional books, published by CRC Press, Springer, McGraw-Hill and IGI Global. His topics of interest include GPU/cloud/edge computing, parallel software design, performance evaluation and benchmarking. Dr. Li is a member of AAAS, a senior member of the IEEE and Fellow of the IET.

Machine Learning, Visual Computing and Signal Processing

Computational Reconstructions of Extracellular Action Potentials and Local Field Potentials of a Rat Cerebellum Using Point Neurons

Arathi Rajendran, Naveen Kumar Sargurunathan, Varadha Sasi Menon, Sneha Variyath, Satram Dayamai Sai, and Shyam Diwakar

Abstract One of the main challenges in computational modeling of neurons is to reconstruct the realistic behavior of the neurons of the brain under different functional conditions. At the same time, simulation of large networks is time-consuming and requires huge computational power. The use of spiking neuron models could reduce the computational cost and time. In this study, the extracellular potentials were reconstructed from a single point neuron model of cerebellum granule neuron, and local field potentials (LFP) were modeled. Realistic reconstruction of cerebellum Crus II evoked post-synaptic local field potentials using simple models of granule neurons help to explore emergent behavior attributing patterns of information flow in the granule layer of the cerebellum. The modeling suggests that the evoked extracellular action potential (EAP) arises from the transmembrane currents correlating spiking activity and conductive properties of the extracellular medium to the LFP. The computation study reproduces experimentally observed in vitro N2a and N2b evoked LFP waves and can be used to test the scaling of models developed from a bottom-up approach.

Keywords Neuron · Modeling · Computational biology · Cerebellum · Extracellular · Local field potential

1 Introduction

Neurons are modeled as stochastic communication channels, and the information processing through the neurons depends on numerous characteristics and to the excitatory and inhibitory stimuli from the synapses. Using mathematical modeling, the nonlinear neuronal properties may be implemented using abstractions like the Hodgkin–Huxley model, which highlights the contribution of a single cell and its dynamic properties [1]. However, integrate and fire models [2] provide a simpler

A. Rajendran · N. K. Sargurunathan · V. S. Menon · S. Variyath · S. D. Sai · S. Diwakar (✉)
Amrita School of Biotechnology, Amrita Vishwa Vidyapeetham, Amritapuri Campus, Kollam 690525, India
e-mail: shyam@amrita.edu

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_1

mechanistic explanation for basic neural operations. In studying circuit or behavioral signals, single neuron contributions to the information processing and the ensemble behavior in complex neural circuits, it is important to attribute a choice of quantitative models, which are optimal for understanding the electro-responsiveness properties that includes both intracellular and extracellular action potentials (IAP and EAP).

EAP is the extracellular potential generated from a neuron composed of several contributions from transmembrane currents, action potential and ion channel currents [3, 4]. The voltage fluctuations can be recorded using an extracellular electrode [5] as a low-frequency signal (<500 Hz) from the extracellular medium. The extracellular signals reflect the underlying activity of neurons and its synapses from a local population surrounding the electrode, and yet the actual number of single neurons contributing to this activity is still unclear [6]. However, computing EAP from spiking model neurons poses a challenge because individual point neuron models do not produce spatially distributed currents, which are necessary to compute LFPs. Whereas, LFPs are generated by the transmembrane currents in the vicinity of the recording electrode and mainly depend on the morphological features of the neurons in the region of interest, the positioning of synapses and also the spatio-temporal variety of synaptic inputs.

The activity of a single granule neuron of the cerebellum during in vitro condition is observed as a negative N_{2a} and N_{2b} waves when a single pulse stimulation applied to MFs [7]. Likewise, in vivo LFP waves are denoted in the form of T (Trigeminal) and C (Cortical) waves, which are produced from the trigeminal cortical pathway from the burst input to the cerebellum during whisker pad air puff stimulation [8]. Modeling circuit functions using a detailed network of biophysical models is computationally expensive and complex in terms of number of equations to be solved and number of synapses to be simulated, so the use of point neuron models may be optimal for predictions of large-scale behavioral predictions. Such models may need methods to explore population activity. In recent years, new technologies have suggested methods to analyze extracellular potentials for larger and spatially diverse neurons in addition, and EAP and LFP reconstructions have been developed using biophysical models [9–12]. In an early study, biophysical neuron models were used to estimate the LFP from the transmembrane ionic current [13], and recently, the technique has been used to validate single neuron as well as neuronal network from the cerebellum [14]. Point source approximation (PSA) method has been employed to record the electrical field generated in the biophysical model of cerebellum granular layer network [15]. The recent studies have shown that [16], the LFP signals can be reconstructed using spiking neuron network models for morphologically detailed neuronal networks such as hippocampal pyramidal neurons by kernel-based method; these signals are also the indicators of disease conditions [17]. Even though the signal reconstruction happens for morphologically detailed neurons, extracellular potentials are much more complex to interpret for small neurons like cerebellum granule neurons using spiking point neuron models, and it should be possible to explain LFP by using the knowledge of granule neuron physiology. As observed in experiments, the evoked extracellular potentials attribute to changes in mossy fiber inputs in both in vitro and in vivo like conditions.

To generate any kind of extracellular potential, the mathematical model of a neuron is needed to have a minimum of a few spatially separated compartments. However, in point neurons, approximation for the entire neuron relies on single values reconstructed representing a point. Cerebellum's granule neurons are small, numerous and compact making them difficult for modeling local field potentials unlike using larger neurons. In this study, the reconstruction of a single granule cell evoked EAP and LFP was performed to model a cerebellum evoked local field, maintaining some of neuronal dynamics using a combination of the point source approximation (PSA) technique and convolution-based methods, to understand the neuronal activity of the motor function related rat cerebellum granular layer.

2 Methodology

This study employs a single cerebellum granule neuron spiking model to reconstruct the EAP during in vitro condition and LFP in both in vivo and in vitro conditions. The extracellular potential evoked by the action potentials in a single cerebellar granule neuron was computed in two distinct stages; first, the granule neuron model [18, 19] by varying input connections has been simulated and recorded the transmembrane currents and then used to compute EAP. The parameters used for the reconstruction of a single cerebellar granule neuron were borrowed from our previous study [20] (see Table 1). All the simulations were done on the NEURON platform, [8], and all the simulations were performed on a workstation with Intel(R) Core(TM) i5-8265U CPU@ 1.80 GHz processor with 8.00 GB RAM.

The cerebellar granule neuron here modeled was based on our previous study [20]. In this study, both in vitro and in vivo style inputs were provided to evoke both intracellular and extracellular action potentials. Both in vitro and in vivo inputs were given by using an artificial spike generator, mimicked the mossy fiber input patterns in the cerebellum. The granule neuron model and the virtual electrode were placed (Fig. 1) in a 3D space with 111.803 µm apart, which was allowed optimal signal

Table 1 Parametric values for granule neuron AdEx model

Parameters	Values
C	150
gL	10
eL	−70
del$T$	4
Vt	−50
τw	13
I	350
A	9
B	250

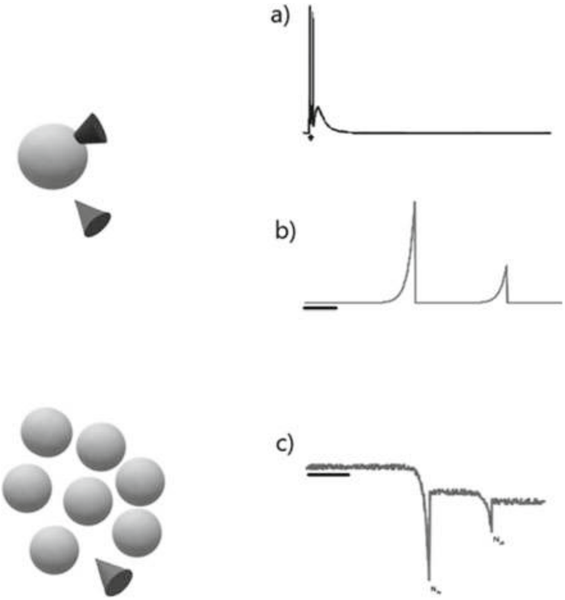


Fig. 1 Cartoon of point neuron models to record intracellular and extracellular action potentials (EAP and LFP). The arrow mark indicating the electrodes for intracellular (**a**), extracellular (**b**) and population activities (**c**), respectively

reconstruction. The dynamics of a single granule neuron was simulated using a point neuron simulation, and the resulting transmembrane currents are recorded for each time steps dt. Based on the resulting transmembrane currents, EAP is calculated. The simulations were performed for varying numbers of mossy fiber input connections and recorded the EAPs, and convolution-based LFP has been reconstructed. There are different approaches such as line source approximation [6], point source approximation (PSA) [2], ReConv [6] and low-pass filtering to quantify extracellular potentials. Here, we have used a combination of PSA and convolution-based LFP reconstruction.

2.1 Point Neuron Model of Cerebellum Granule Neuron

Adaptive exponential integrate and fire model (AdEx) was used to reconstruct the firing dynamics of the cerebellar granule neuron [21]. The model included two equations, one for regulating the state variables and the other for regulating the adaptation constant.

$$\begin{aligned} C * \mathrm{d}V/\mathrm{d}t = &- \mathrm{gL} * (V - \mathrm{eL}) + \mathrm{gL} * \mathrm{del}T \\ &* \exp((V - Vt)/\mathrm{del}T) + I - w \end{aligned} \tag{1}$$

$$\tau w * \mathrm{d}w/\mathrm{d}t = a * (V - eL) - w \tag{2}$$

where (V) was the membrane potential, (w) was the adaptation current, (C) was the total capacitance, total leak conductance (gL), effective rest potential (eL), threshold slope factor (delT), effective threshold potential (Vt) and combining the parameters C and gL into time scale $\tau m = C/\mathrm{gL}$ and the rest four parameters were directly proportional to the conductance (a), the time constant (τw), the spike-triggered adaptation (b) and the resting potential Vr.

2.2 Estimation of EAP Using Point Source Approximation

EAP generated from the neuron was approximated to a point source in the center where the transmembrane current was assumed to be generated unlike in our biophysically detailed multi-compartmental granule neuron model [20]. The electrical potential from the extracellular medium was estimated using Laplace's equation. For a point source in extracellular space, extracellular potential (ø) at a point r is expressed as

$$ø = \rho I / 4\pi \mathrm{r} \tag{3}$$

where I was the current generated from the source, ρ was the conductivity of the medium and r was the distance from source to the point of measurement. Single neuron EP was constructed by the extracellular voltage from the point source. The extracellular potential for the neuron was computed by validating the transmembrane current from the single granule neuron cell. The transmembrane current was reconstructed as

$$I = C\,\mathrm{d}V/\mathrm{d}t \tag{4}$$

where I was the transmembrane current from the neuron, C the capacitive current, dV the extracellular voltage, then EAP from the estimated transmembrane current point source of single neuron was expressed as

$$\mathrm{EP}_{\mathrm{point}} = I * d_{\mathrm{EP}} \tag{5}$$

where I was the transmembrane current at a point source and $d_{\mathrm{EP},}$ the distance from the electrode to the point of calculation of the EAP.

2.3 Convolution-Based Method for the Reconstruction of LFP

The extracellular responses were convolved to an averaged extracellular action potential. Single pulse as well as burst mossy fiber inputs were given to granule neuron to mimic the distant electrode stimulation on a mossy fiber bundle on an acute brain slice and anesthetized animal, respectively. As seen in experiments, the LFP was modeled as a temporal population code made by various signals with noise (jitter). To calculate the LFP from multiple signals from a single spiking neuron model, 25 average trials were performed with a jitter delay of <3 ms. This method matched our previous study [6]; however, convolution was performed only once. The convolution averages multiple possible activation patterns and incorporates temporal jitter.

3 Results

3.1 Intracellular Potentials Shape Extracellular Potentials

Extracellular potentials that result from transmembrane current distributed across the cell surface. Since the point neuron models lack the spatial properties, the contribution of intracellular potential to the EAP has important implications.

The input through mossy fiber stimuli activated the granule cells in the cerebellum. Stimuli were varied based on the number of given mossy fiber input. The granule neuron responses matched experimental data[7], and the changes in mossy fiber inputs vary from 1 to 4. The mossy fiber stimuli were presented to the granule neuron at 20 ms.

In the case of point neurons, the generated patterns (Fig. 2 lines.) represented the evoked EAP in the Crus IIa rat somatosensory cerebellar cortex for the corresponding local field potentials observed in experimental studies [22]. The result indicated that the EAP may include spiking information attributed to the field by a single granule neuron.

The point neuron-based modeling of LFP includes a temporal delay attributed to the latency from multiple synaptic inputs in generating ionic current in the granular layer, especially due to the channel mechanisms of both sodium and potassium ions [23]. The latency of extracellular potential was considered due to the time taken for synaptic transmission and the action potential propagation along the axon [24]. In the model, it was observed that the maximum time taken for EPSP signals to reach the electrode was 27 ms while the maximum time taken by the action potential to reach the electrodes was 92 ms. It was noted that changes in synaptic connections led to changes in frequency and position of the spikes and it was also observed across the modeled extracellular potential. The spike shape and amplitude are known to be depended, and the amplitude decays with distance of electrode from the neuron [25].

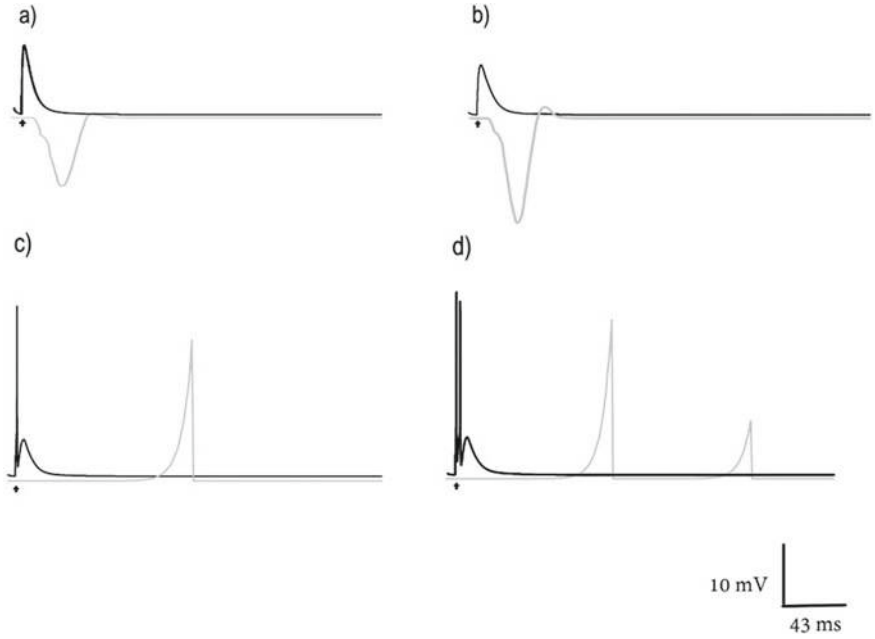


Fig. 2 Modeling intra and extracellular potentials of a single cerebellum granule neuron. **a**, **b**, **c** and **d** (black) are representing the spiking activity of a single cerebellum granule neuron during 1, 2, 3 and 4 mossy fiber stimuli, while the extracellular action potential of a single granule neuron corresponding to the respective spiking activities is represented in gray color. Due to small size, EAP of single granule neurons have not be recorded

3.2 Convolution-Based LFP Reconstruction

Single pulse (in vitro) mossy fiber input was applied to a single spiking granule neuron by varying mossy fiber excitatory connections from 1 to 4, and negative N_{2a} and N_{2b} waves were generated from various signals from a single neuron. Similarly, the T and C waves were reconstructed during in vivo like conditions by giving burst inputs; 5 spikes/burst (short burst) and 9 spikes/burst (long burst).

Since the shape of the wave was attributed to the temporal jitter convolution of several individual responses, it was observed that the N_{2a} and N_{2b} waves may be related to the spiking activity in the granular layer [7]; in the model (Fig. 3a), N_{2a} represented the first spike and N_{2b}, the second spike in the doublet of the granule neuron, respectively. The spike doublets contributed the N_{2a} and N_{2b} in the LFP, and it was observed that the NMDA receptor mediated depolarization was responsible for the generation of N2b spikes in the neuron. This matched experimental observations of [7]. Our spiking neuron reconstruction of the cerebellum Crus IIa evoked LFP suggests reduced excitability modified the amplitude of the waveforms and without spike doublet which perhaps are different from larger neurons such as pyramidal or Purkinje neurons.

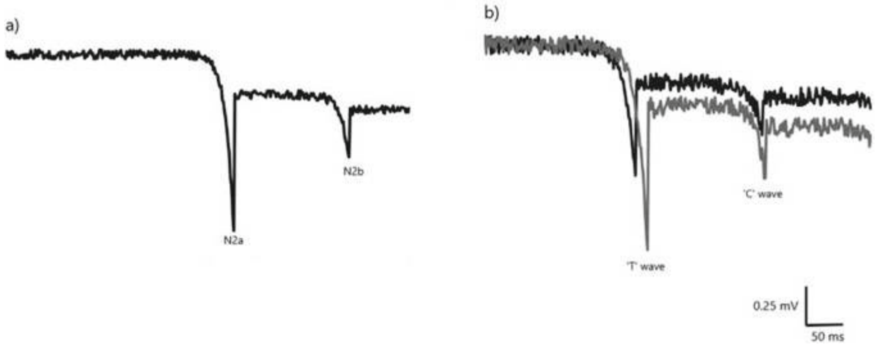


Fig. 3 LFP traces from a single granule neuron using convolution-based methods; **a** N_{2a} and N_{2b} waveform of single granule neuron in in vitro like condition. **b** Simulated in vivo LFP waves indicating T and C waves during short (black) and long (grey) spiking burst inputs

In vivo extracellular potentials were reconstructed by activating MF with long and short spike bursts like inputs. The relationship between LFP and spike is complex to understand since, the recorded LFPs reflect some but not all input spikes. However, the average LFP estimation using convolution-based method indicated that the LFP associated with each spike of single neuron. Here, the convolution-based method generated T and C waves with appropriate shape and timing were observed (Fig. 3b).

4 Discussion

In this study, a reconstruction of extracellular potential based on modeled intracellular properties was performed in the context of cerebellum's granular layer microcircuitry. The match across temporal scales of neuronal activity may indicate the differences in information coding in the underlying neurons. Modulation of signal behavior may act as markers for better network modeling. In spiking models, the shape of the LFP may be attributed to the contribution of delays modeled in perceiving the intracellular action potentials by the electrode.

To explore behavioral reconstruction of large-scale neural circuits in simpler computing devices, point neuron models may be employed, and extracellular potentials like EAP and field potentials like LFP may be computed also on electrotonically small or compact neurons. This needs approximate modeling of transmembrane currents and their transformation as summed averaged potentials. The main aim of this study was to compute extracellular potentials accurately using a suitable approximation technique from a single point neuron and convolution-based approach, avoiding a realistic spatial structure that could add computational complexity.

In our reconstructions, the peak of EAP resembled the timeline of Na^+ conductance as in the biophysical models. The variation in the action potential shape could be due to the number of excitatory synaptic connections to the neuron. Since the point neurons do not have any spatial properties, here only the transmembrane current has been considered and excluded other membrane properties like membrane resistance and membrane capacitance.

The strategy used here to reconstruct evoked post-synaptic cerebellum local field potentials employs the intrinsic spiking dynamics. This allows a unique window to infer the properties of the point neuron and also can be used to estimate synaptic connections underlying the point source. For bottom-up reconstructions of entire brain circuits, this style of extracellular potential reconstruction allows interpretations of neural network activity and also to constrain the detailed models. Through this model, the EAP captures a multitude of processes like synchronized synaptic potentials and voltage gated membrane oscillations and gives insights to independent components of a LFP. A combined analysis of EAPs and spikes may offer novel insights into the circuit mechanism generating neural representation of various information processing that may not be obtained by just examining the spikes.

Unlike other published studies, this convolution-based reconstruction on the spiking cerebellum granule neuron model allows template-based reconstructions of the independent components of post-synaptic LFP. Since the granule neurons are similar one to each other, it may help exploring the redundancy of the code in the cerebellum granular layer and an outlook into the averaged approximations of granular layer activity.

5 Conclusion

This point neuron model may be used to estimate network function through estimations of local field potential and other neuronal activities as it greatly reduces the computational cost associated with simulating neuronal networks of morphologically accurate neurons. We are yet to scale models that allow both spike and burst-based activity reconstructions and the effects of realistic spatial localization of the point neurons. This study will enhance studies attributed to neuronal losses and consequent circuit functions and dysfunctions.

Acknowledgements This work derives direction and ideas from the Chancellor of Amrita Vishwa Vidyapeetham, Sri Mata Amritanandamayi Devi. This study was partially supported by Amrita School of Biotechnology, Department of science and technology grant DST/CSRI/2017/31, Government of India, and embracing the World Research-for-a-Cause initiative.

References

1. Hodgkin, A.L., Huxley, A.F.: A quantitative description of membrane current and its application to conduction and excitation in nerve. J Physiol. **117**(4), 500–544 (1952)

2. Naud, R., Marcille, N., Clopath, C., Gerstner, W.: Firing patterns in the adaptive exponential integrate-and-fire model. Biol. Cybern. **99**, 335–347 (2008). https://doi.org/10.1007/s00422-008-0264-7
3. Buzsáki, G., Anastassiou, C.A., Koch, C.: The origin of extracellular fields and currents—EEG, ECoG, LFP and spikes. Nat. Rev. Neurosci. **13**(6), 407–420 (2012). https://doi.org/10.1038/nrn3241
4. Eccles, J.C.: Interpretation of action potentials evoked in the cerebral cortex. Electroencephalogr. Clin. Neurophysiol. **3**, 449–464 (1951). https://doi.org/10.1016/0013-4694(51)90033-8
5. Rall, W., Shepherd, G.M.: Theoretical reconstruction of field potentials and dendrodendritic synaptic interactions in olfactory bulb. J. Neurophysiol. **31**, 884–915 (1968). https://doi.org/10.1152/jn.1968.31.6.884
6. Diwakar, S., Lombardo, P., Solinas, S., Naldi, G., D'Angelo, E.: Local field potential modeling predicts dense activation in cerebellar granule cells clusters under LTP and LTD control. PLoS ONE **6**, e21928 (2011). https://doi.org/10.1371/journal.pone.0021928
7. Mapelli, J., D'Angelo, E.: The spatial organization of long-term synaptic plasticity at the input stage of cerebellum. J. Neurosci. **27**, 1285–1296 (2007). https://doi.org/10.1523/JNEUROSCI.4873-06.2007
8. Roggeri, L., Rivieccio, B., Rossi, P., D'Angelo, E.: Tactile stimulation evokes long-term synaptic plasticity in the granular layer of cerebellum. J. Neurosci. **28**, 6354–6359 (2008). https://doi.org/10.1523/JNEUROSCI.5709-07.2008
9. Parasuram, H., Nair, B., Naldi, G., D'Angelo, E., Diwakar, S.: Understanding cerebellum granular layer network computations through mathematical reconstructions of evoked local field potentials. Ann. Neurosci. **25**, 11–24 (2018). https://doi.org/10.1159/000481905
10. Gold, C., Henze, D.A., Koch, C.: Using extracellular action potential recordings to constrain compartmental models. J. Comput. Neurosci. **23**, 39–58 (2007). https://doi.org/10.1007/s10827-006-0018-2
11. Pettersen, K.H., Hagen, E., Einevoll, G.T.: Estimation of population firing rates and current source densities from laminar electrode recordings. J. Comput. Neurosci. **24**, 291–313 (2008). https://doi.org/10.1007/s10827-007-0056-4
12. Einevoll, G., Lindén, H., Tetzlaff, T., Ski, S., Pettersen, K.: Local field potentials. Princ. Neural Coding, 37–60 (2013). https://doi.org/10.1201/b14756-5
13. Pettersen, K.H., Lindén, H., Dale, A.M., Einevoll, G.T.: Extracellular spikes and CSD. In: Handbook of Neural Activity Measurement, pp. 92–135. Cambridge University Press (2012)
14. D'Angelo, E., Solinas, S., Garrido, J., Casellato, C., Pedrocchi, A., Mapelli, J., Gandolfi, D., Prestori, F.: Realistic modeling of neurons and networks: towards brain simulation. Funct. Neurol. **28**, 153–166 (2013). https://doi.org/10.11138/FNeur/2013.28.3.153
15. Parasuram, H., Nair, B., D'Angelo, E., Hines, M., Naldi, G., Diwakar, S.: Computational modeling of single neuron extracellular electric potentials and network local field potentials using LFPsim. Front. Comput. Neurosci. **10**, 1–13 (2016). https://doi.org/10.3389/fncom.2016.00065
16. Telenczuk, B., Telenczuk, M., Destexhe, A.: A kernel-based method to calculate local field potentials from networks of spiking neurons. J. Neurosci. Methods **344**, 108871 (2020). https://doi.org/10.1016/j.jneumeth.2020.108871
17. Terlau, J., Yang, J., Khastkhodaei, Z., Seidenbecher, T., Luhmann, H.J., Pape, H., Lüttjohann, A.: Spike-wave discharges in absence epilepsy: segregation of electrographic components reveals distinct pathways of seizure activity. J. Physiol. **598**, 2397–2414 (2020). https://doi.org/10.1113/JP279483
18. Vijayan, A., Medini, C., Palolithazhe, A., Muralidharan, B., Nair, B., Diwakar, S.: Modeling pattern abstraction in cerebellum and estimation of optimal storage capacity. In: 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 283–289. IEEE, Kochi (2015)
19. Medini, C., Nair, B., D'Angelo, E., Naldi, G., Diwakar, S.: Modeling spike-train processing in the cerebellum granular layer and changes in plasticity reveal single neuron effects in neural ensembles. Comput. Intell. Neurosci. **2012** (2012). https://doi.org/10.1155/2012/359529

20. Diwakar, S., Magistretti, J., Goldfarb, M., Naldi, G., D'Angelo, E.: Axonal Na+ channels ensure fast spike activation and back-propagation in cerebellar granule cells. J. Neurophysiol. **101**, (2009)
21. Medini, C., Vijayan, A., D'Angelo, E., Nair, B., Diwakar, S.: Computationally efficient bio-realistic reconstructions of cerebellar neuron spiking patterns. In: ACM International Conference Proceeding Series (2014)
22. Telenczuk, B., Dehghani, N., Le Van Quyen, M., Cash, S.S., Halgren, E., Hatsopoulos, N.G., Destexhe, A.: Local field potentials primarily reflect inhibitory neuron activity in human and monkey cortex. Sci. Rep. **7**, 1–10 (2017). https://doi.org/10.1038/srep40211
23. Parasuram, H., Nair, B., Naldi, G., D'Angelo, E., Diwakar, S.: Exploiting point source approximation on detailed neuronal models to reconstruct single neuron electric field and population LFP. In: Proceedings of the International Joint Conference on Neural Networks. Institute of Electrical and Electronics Engineers Inc. (2015)
24. Mazzoni, A., Logothetis, N.K., Panzeri, S.: Information content of local field potentials: experiments and models. In: Principles of Neural Coding, pp. 411–430 (2013)
25. Pettersen, K.H., Einevoll, G.T.: Amplitude variability and extracellular low-pass filtering of neuronal spikes. Biophys. J. **94**, 784–802 (2008). https://doi.org/10.1529/biophysj.107.111179

Iris Recognition Using Integer Wavelet Transform and Log Energy Entropy

Jincy J. Fernandez and Nithyanandam Pandian

Abstract As the technology is reaching its next level day by day, the concerns over information or data security are also creeping up. Biometric systems have been widely used in many real-world applications in order to provide more security to the data. Iris recognition system has become a widely used system for human identification from the last few decades. In this paper, an efficient iris recognition system is proposed where iris localization is carried out by first finding the pupil-iris boundary using the connected component analysis approach. And then by considering the pupil center as the reference point, it traverses through the virtual outer boundary to detect the iris-sclera boundary. After applying normalization on the iris region, the iris region is partitioned into non-overlapping blocks. Further, a combination of integer wavelet transform (IWT) with log energy entropy (LEE) is applied on each block to extract the unique iris code as the feature vector. The experiments have been conducted using the multimodal biometric database, SDUMLA-HMT. The proposed system has succeeded in achieving a low false acceptance rate and a very low false rejection rate. Also, the uniqueness of the iris patterns is evaluated in terms of degrees of freedom and is found to be a promising one.

Keywords Iris recognition · Connected component analysis · Integer wavelet transform · Log energy entropy

1 Introduction

With the movement of the present world toward a complete digitization phase, where important personal and private information are being converted into digital format

J. J. Fernandez (✉)
Research Scholar, VIT University, Chennai, India
e-mail: jincyj.fernandez2015@vit.ac.in

N. Pandian
Professor, VIT University, Chennai, India
e-mail: nithyanandam.p@vit.ac.in

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_2

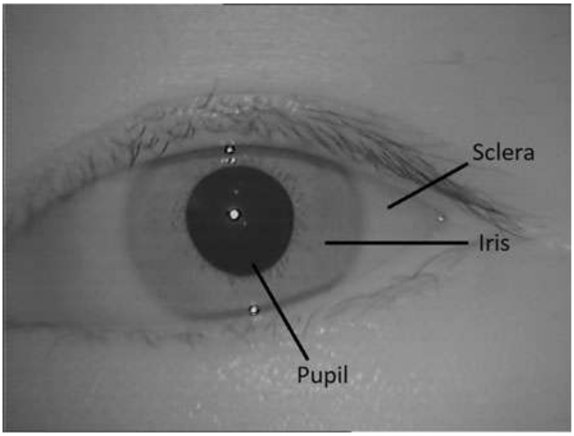


Fig. 1 Sample iris image. Image taken from SDUMLA-HMT database [31]

for more convenience and accessibility. In situations where it is required to recognize a person at a large scale and speed, it could be a challenging and tedious process. The idea of using biometrics comes into existence in such testing times. Biometrics has the capability to make authentication dramatically faster, easier, and more secure than traditional passwords. Each new device developed, an organization formed or technology used today, i.e., from mobile devices to laptops, government agencies to multinational organizations, voter registrations to national ID cards, and biometric watermarking systems [14], and biometric is evolving everywhere in every phase and aspect of our lives. Among all the biometrics, iris recognition system has become a widely used system for human identification. The human iris is the annular part seen between the black colored pupil and white-colored sclera as shown in Fig. 1. The extraordinary structure of the iris includes freckles, coronas, stripes, furrows, and crypts and it makes iris biometric to uniquely identify the individual. Another advantage of using iris biometric is that iris is an internal organ that is externally visible; so less damage occurs to iris, unlike fingerprint [15].

1.1 Main Contribution of the Work

- The use of block-based integer wavelet transform, which helps to extract more texture information from the iris.
- The novel use of log energy entropy in the iris recognition system.
- There have been no previous applications of the use of a combination of integer wavelet transform (IWT) with log energy entropy (LEE) to iris recognition system problems. With respect to this, a novel system based on IWT and LEE is proposed.

The rest of the paper is organized as follows: Sect. 2 contains the review on recent iris recognition algorithms. Section 3 gives an overview of the proposed system. Section 4 describes, evaluates, and discusses the performance of the proposed approach. Finally, Sect. 5 contains a conclusion on the proposed approach and future works.

2 Related Study

The first iris recognition system developed by Daugman [8] operates as follows: the edges of the pupil and the iris are detected using integro-differential operator. The iris region is then normalized using Daugman's rubber-sheet model. Finally, 1D Gabor wavelets are used to extract the iris features and then get encoded. Hamming distance is used for human identification and verification based on a predefined threshold value. Wildes [33] used Hough transform to find the pupillary and limbic boundary as part of iris localization. Further, a normalized correlation and Laplacian of Gaussian filters [4] at different resolutions are used for pattern matching. Chen and Chu [5] suggested an iris feature extraction technique with intelligent classifier. The iris features are represented using 1D circular profile. An optimized PNN model is considered, which is the combined form of probabilistic neural network [28] and particle swarm optimization [20]. Hybrid use of scattering transform-based features and textural features is suggested by Minaee et al. [22]. PCA is used to reduce the dimensionality of the extracted feature vector and a minimum distance classifier is used for recognition.

The first iris recognition system developed by Daugman [8] operates as follows: the edges of the pupil and the iris are detected using integro-differential operator. The iris region is then normalized using Daugman's rubber-sheet model. Finally, 1D Gabor wavelets are used to extract the iris features and then get encoded. Hamming distance is used for human identification and verification based on a predefined threshold value. Wildes [33] used Hough transform to find the pupillary and limbic boundary as part of iris localization. Further, a normalized correlation and Laplacian of Gaussian filters [4] at different resolutions are used for pattern matching. Chen and Chu [5] suggested an iris feature extraction technique with intelligent classifier. The iris features are represented using 1D circular profile. An optimized PNN model is considered, which is the combined form of probabilistic neural network [28] and particle swarm optimization [20]. Hybrid use of scattering transform-based features and textural features is suggested by Minaee et al. [22]. PCA is used to reduce the dimensionality of the extracted feature vector and a minimum distance classifier is used for recognition.

An iris recognition system based on multiscale morphological features is developed by Umer et al. [30]. In this approach, a restricted circular Hough transform [26] is used to localize the iris region from the input eye image. The normalized image undergoes image sharpening by the toggle filter. Further, features are extracted from the sharpened image by using morphologic top-hat transform. The advantage of the morphological filter is that it highlights the texture of the iris image. An SVM classifier is used for recognition purposes. The main disadvantage is that it works well for some of the databases only. The use of human interpretable features is introduced in [6] to promote iris recognition in forensics. The developed system involves two steps: iris crypts detection and crypts matching. Morphological operations are employed to detect crypts, whereas Earth Mover's distance [27] is used for crypt matching. The disadvantage of the system is that all the crypts are not captured from an image due to various physical conditions. The iris recognition system suggested by [2] involves the generation of a dictionary of feature vectors, basis pursuit de-noising optimization technique for finding the sparse vector, use of three classifiers such as K-nearest space-based, sector-based, and CSCI-based classifiers. Further, a final decision is taken by using the genetic algorithm.

A cross-spectral matching-based iris recognition is suggested in [23] where the input iris image in cross-spectral or cross-sensor form undergoes preprocessing and feature extraction process to generate binary feature codes. This feature code is transformed into a real-valued feature descriptor to provide a robust and meaningful adaptation strategy. This is followed by computing a similarity measure based on real-valued representation of log-Gabor filter responses. Finally, a Naive Bayes nearest neighbor (NBNN)-based domain adaptation framework is used to recognize both cross-spectral and cross-sensor images. An iris recognition system using fuzzy edge processing is suggested in [19]. After obtaining a noise reduced eye image by applying Gaussian filtering, edges are estimated from it. Edge estimation traces the high-frequency contents in the images and the challenge lies in effective edge estimation in noisy and blurred images. The fuzzy membership functions for the edge estimation model are defined in terms of the gradient of the image. Circular Hough transform is applied on edge detected image to localize the iris portion from the eye image. 1-level Haar wavelet is then applied to the enhanced iris image. The statistical parameters such as mean and variance are further obtained from the approximation and detail coefficient bands (LL, HL, LH, and HH subbands) which act as enhanced feature elements.

The scale and rotation invariant SIFT features are used in [25]. For recognition, the image pairs containing similar patterns are identified by finding the number of matching SIFT features using Euclidean distance. The drawback of the approach is that some of the iris textures near the pupillary boundary are removed during the segmentation process. And it is already proved that the presence of iris texture is more near to the pupillary boundary [11]. So the system failed to give a promising recognition rate. A deep learning-based approach is developed by Zhao et al. [35] where a capsule network is used for iris recognition. With the use of normalizing the ROI image and further resizing, the performance of the system gets degraded as the uniqueness of the iris patterns cannot be proved. Dua et al. suggested an iris recognition system where a radial basis function neural network is used for classification [9]. For that, the eye image undergoes preprocessing and feature extraction process where 1D Gabor features are considered. The obtained accuracy is not a promising one compared to other existing approaches.

Geometric invariant feature representation is used in [24], which considers a method of using Zernike moments and Maitra's moment. Initially, Zernike moments are used to represent the global texture feature. It is followed by extracting local features using HU and Maitra's moments. Further, the local and global features are combined to form the feature vector. The accuracy achieved is 99.03% with the use of the K-nearest neighbor classifier. The main disadvantage is the uniqueness of the iris patterns generated using the approach is not stated. In [1], a base64 encoder is used to extract a set of stylometric-based features from the normalized iris image. This is followed by oversampling the feature set and use the random forest [16] machine learning algorithm for identification purposes. The main advantage of the approach is the absence of a false acceptance rate. The use of an iris recognition system specifically for postmortem samples is developed by Trokielewicz et al. [29] where a deep learning-based segmentation is employed to extract irregular iris texture. The main disadvantage is that the pupillary and limbic boundaries are assumed to be circular, so that good iris features near to the pupillary boundary are not extracted.

3 Proposed System

The proposed system involves two phases: the registration phase and identification or verification phase. The registration phase includes six main steps: iris localization, iris normalization, block partitioning, integer wavelet transformation, computation of log energy entropy, and iris code generation. This iris code is further stored in the database. During the identification phase, an individual is recognized by the system based on the comparison between individual's iris code with all the iris codes stored in the database, whereas during verification, an individual is recognized with n samples of the claimer. The verification phase is a 1 to 1 matching process, whereas the identification phase is a 1 to n matching process is done. The framework of the proposed system is shown in Fig. 2.

3.1 Iris Localization

Iris localization aims to isolate the iris region from other regions of the eye image such as pupil, sclera, eyelids, and eyelashes. In general, localization of the iris region is done from the input eye image based on the assumption that the iris is circular in shape. But it is proved that the pupil boundary is not always circular [11] as shown in Fig. 3a. Such a simple assumption leads to inaccurate iris localization, thereby the generation of unique iris code is affected much. The proposed work finds the pupil-iris boundary using the connected component labeling approach [11] and it accurately finds even if the pupillary boundary is non-circular in shape as shown in

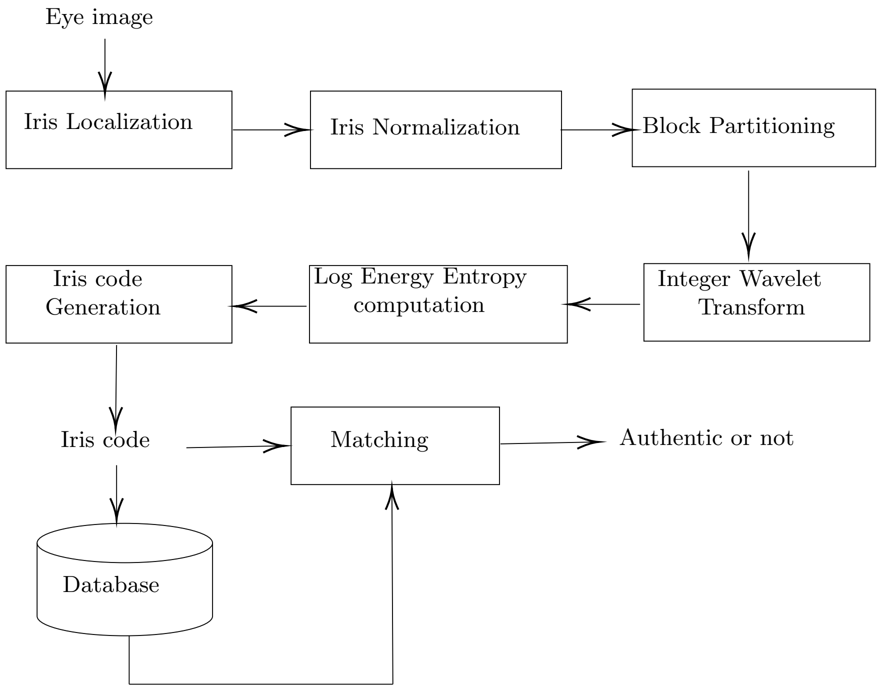
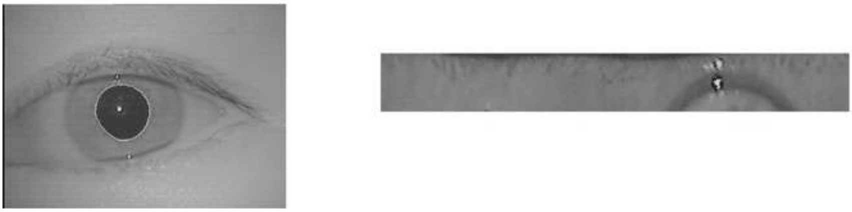


Fig. 2 Framework of the proposed system



(a) Image after pupil boundary detection

(b) Image after normalization process

Fig. 3 Results after pupil boundary detection and normalization

Fig. 3a. By considering the pupil center as the reference point, it traverses through the virtual outer boundary to detect the limbic boundary. The virtual outer boundary is found by extending the radius of the pupillary boundary.

3.2 Iris Normalization

Due to the image acquisition issues, the captured irises may not be of the same size. Such elastic deformation in the iris texture may result in incorrect recognition. This issue can be solved by the iris normalization process, which is the unwrapping of the iris region into a rectangular form of predefined size by using Cartesian to polar transformation. The most widely used normalization method, Daugman's rubber-sheet method [8] is used in the work. Daugman's rubber-sheet model maps each point in the iris region to a pair of polar coordinates (r, θ) where r is in the interval $[0, 1]$ and θ is the angle in $[0, 2\pi]$ so that a translation-invariant representation of the iris image is obtained.

3.3 Block Partitioning

To extract more iris features from the image, the normalized iris image is partitioned into non-overlapping blocks of size 10×10. Block partitioning leads to view the image as a combination of different blocks and each block is perceived as one source of the whole image. This partition scheme is not only simple, and fast, but also resist the changes that occurred in other regions, while the global approach is more likely to corrupt by those changes. That is, all the small blocks of different iris images will not share the same distribution.

3.4 Integer Wavelet Transform

The wavelet transform is a widely used method, which gives a multi-resolution analysis of the signal. When a wavelet transform is applied on a discrete signal $x(n)$, it is analyzed in terms of high frequency component coefficient, $D_j(k)$ and low frequency component coefficient, $A_j(k)$ at instant k and scale j. The input signal $x(n)$ can be represented by the sum of all these components, as given in Eq. (1).

$$\begin{aligned} x(n) &= D_1(n) + A_1(n) = D_1(n) + [D_2(n) + A_2(n)] \\ &= D_1(n) + D_2(n) + [D_3(n) + A_3(n)] = \cdots \\ &= \sum_{j=1}^{J} D_j(n) + A_J(n) \end{aligned} \tag{1}$$

where J is the maximal scale. One of the problems associated with the conventional wavelet filters is that when the input data is an integer, the output data may not be an integer so that the perfect reconstruction of the image is not possible. This happens due to the truncation or rounding-off process used in conventional wavelets. As a solution to this, integer wavelet transform (IWT) can be used which maps input integer data to output integer data. This results in no loss of information through forward and inverse transform. Lifting scheme [3, 17] is used in IWT where it involves three steps:

1. Split: split process deals with partitioning the input data set into odd and even polyphase components.
2. Predict (dual lifting): the odd polyphase component is predicted based on a linear combination of samples of the even polyphase component. While performing predict operation, the component contains much less information than the original data set. Some basic properties of the image such as the mean value of the image are lost during the predict step. The new odd polyphase component is computed as given in Eq. (2).

$$\text{New\_oddset} = (\text{Evenset} - \text{Oddset}) \tag{2}$$

3. Update: the even polyphase component is updated based on a linear combination of difference samples obtained from the predict step. The update process helps to restore the information loss that occurred in predict step, so that the global properties of the original image in the subsets are preserved. The new even polyphase component is computed as given in Eq. (3).

$$\text{New\_evenset} = \text{Oddset} + \lfloor \frac{\text{New\_oddset}}{2} \rfloor \tag{3}$$

Equations (2) and (3) depict the forward transform. The inverse transform is done in reverse way as given in Eqs. (4) and (5). The steps involved in inverse transform are:

1. Inverse primal lifting: The original even set data is computed.

$$\text{Original\_evenset} = \text{New\_evenset} + \lceil \frac{\text{New\_oddset}}{2} \rceil \tag{4}$$

2. Inverse dual lifting: Computes the original odd set.

$$\text{Original\_oddset} = \text{Original\_evenset} - \text{New\_oddset} \tag{5}$$

3. Merge: The Original_oddset and Original_evenset are combined to get the input data back.

The main advantages of IWT using lifting schemes are:

- High computational efficiency.
- Auxiliary memory is not required and the input data is replaced by its wavelet coefficients.
- Ensures an ideal reconstruction for lossless compression.

After the partitioning of the normalized iris image, IWT is applied on each block to extract more textural information from the iris. The block-based IWT results in the generation of one approximation subband and three detail subbands per block.

3.5 Computation of Log Energy Entropy

Entropy is a higher-order invariant statistic and is widely used in information theory and signal processing areas. Inspired by local matching approaches for iris recognition, the novel use of log energy entropy features is done in this work for iris recognition. The log energy entropy is a type of wavelet entropy, which gives a time frequency domain characteristics and it is used to quantify the reliability and accuracy of a feature. The main advantages of log energy entropy are:

- Detection of fine changes in a non-stationary signal is easier.
- Computationally simple.
- Performance does not depend on any parameters.

The applications of log energy entropy involve locating fault on transmission lines [10], ground moving target recognition [12], BCI-oriented EEG analysis [13], breast cancer detection [18], automated diagnosis of atrial fibrillation ECG signals [21], and face recognition [34]. Among the various entropies, log energy entropies are more representative to blocks of iris images as it has more significant and better discrimination capability [7]. The log energy entropy of the coefficients of the detail subbands such as LH, HL, and HH subbands are computed and are considered as the feature vector in the proposed work. The log energy entropy is computed as given in Eq. (6).

$$E(s) = \sum_i \log(s_i^2) \tag{6}$$

where s is the image and s_i is the ith wavelet coefficients of s. An adaptive threshold is computed based on the log energy entropies (LEE) computed from LH, HL, and HH subbands of all the blocks. The computation of the adaptive threshold is given in Eq. (7).

$$Th = \frac{\mu(L_1, L_2, \ldots, L_n)}{max(L_1, L_2, \ldots, L_n)} \tag{7}$$

where $\mu(L_1, L_2, \ldots, L_n)$ is the mean LEE of the iris image, $max(L_1, L_2, \ldots, L_n)$ is the maximum LEE of the iris image, and n is the number of LEEs.

3.6 Iris Code Generation

After computing LEE of all the blocks and the adaptive threshold, the binarized iris code (feature vector) is generated by quantizing these LEE's. One bit code is generated for each LEE. Thus, the length of the feature vector = $nb * ns$, where nb represents the number of blocks and ns denotes the number of subbands used for LEE computation. The normalized image of size 50 × 360 when partitioned into blocks of size 10 × 10 results in 180 blocks. With $nb = 180$ and $ns = 3$, the length of the feature vector is 540 bits. The iris bit code for jth LEE is computed as given in Eq. (8).

$$C_n(j) = \begin{cases} 1, & if \quad \frac{L_j}{max(L_1, L_2, \ldots, L_n)/2} > Th \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

3.7 Identification or Verification

For each iris image, a feature vector (iris code) is generated using the above feature extraction algorithm. During the identification or verification phase, the iris codes are compared using Hamming distance [32] given in Eq. (9) and it measures the fraction of mismatching bits. The function $f_1()$ in Eq. (10) decides whether the individual is genuine or imposter. A value of 1 indicates that the individual is correctly recognized.

$$HD = \frac{1}{N} \sum_{i=1}^{N} A_i \oplus B_i \tag{9}$$

where A and B represent the two iris codes used for matching and N indicates the total length of the iris code.

$$f_1(HD, \text{threshold}) = \begin{cases} 1, & \text{if } HD \leq \text{threshold} \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

4 Experimental Results

4.1 Database Description

The proposed approach has been tested on the SDUMLA-HMT multimodal database [31] introduced by Shadong University. The database contains five biometric traits: face, fingerprint, finger vein, iris, and gait. The database includes real multimodal data from 106 subjects. The iris data is collected with an intelligent iris capture device developed by University of Science and Technology of China under near infrared illumination. The iris database contains a total of 1060 images with 5 images taken per eye. Among the 106 subjects, there are 61 males and 45 females in the age group between 17 and 31. The evaluation of our proposed method was carried out on a PC with Intel (R) Core (TM) Processor (i5-3470, 3.2 GHz), 8 GB of RAM, and the methods were coded in MATLAB.

4.2 Experimental Setup

4.2.1 Block Size

The normalized iris image is partitioned into blocks of varying sizes. By considering a block size of 10×10, the normalized iris image of size 50×360 is divided into 180 blocks. And three subbands are considered per block, thereby generating an iris code of length 540 bits. The block size is selected based on the experiments done by varying the size and it is found that block size of 10×10 is appropriate for getting better accuracy.

4.2.2 Hamming Distance Threshold

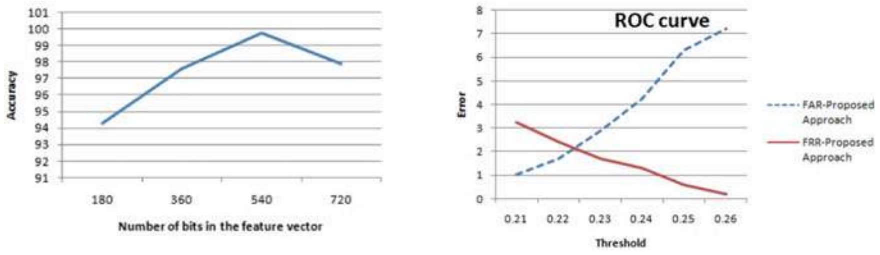
The threshold value is determined after having several experiments where the iris code is compared with the iris codes stored in the database. By increasing the threshold value, rightful owners are falsely classified as non-matches. Hence, an optimum threshold of 0.23 is used in the proposed work.

4.3 Performance Analysis

The performance of the proposed work is analyzed in terms of the accuracy and uniqueness of the iris patterns. False acceptance rate (FAR) and false rejection rate (FRR) are also used to evaluate the performance of the proposed system. The FAR, FRR, and accuracy (true positive rate) are computed as given in Eq. (11). The sample output after detecting pupil boundary and normalization are shown in Fig. 3a, b respectively.

4.3.1 Accuracy

The verification accuracy of the proposed approach by varying feature vector size is shown in Fig. 4a. As it can be seen in this figure, with the increase of the number of detail subbands used (from 1 to 3) for LEE computation, the accuracy was increasing.



(a) Recognition accuracy as a function of feature vector size

(b) ROC Curve

Fig. 4 Performance analysis

Table 1 Comparison of accuracy with existing works

Approach	Accuracy
SIFT features [25]	93.4
Image moment features [24]	98.6
Proposed approach	**99.8**

But the accuracy started decreasing when approximation subband was included for LEE computation. By changing the threshold values, the verification accuracy of the proposed system has been evaluated through the receiver operating characteristics curve (ROC curve) as shown in Fig. 4b. As it can be seen in Fig. 4b, the proposed system has succeeded in achieving a low false acceptance rate and false rejection rate.

$$\begin{aligned}\text{False Acceptance Rate (FAR)} &= \frac{\text{Number of falsely accepted images}}{\text{Total number of images}} * 100\\ \text{False Rejection Rate (FRR)} &= \frac{\text{Number of falsely rejected images}}{\text{Total number of images}} * 100\\ \text{Accuracy} &= \frac{\text{Number of correctly accepted images}}{\text{Total number of images}} * 100 \qquad (11)\end{aligned}$$

Table 1 gives a comparison of the performance of the proposed approach with some of the existing approaches. As it can be seen from the table, using a combination of IWT and LEE gives a better recognition rate.

4.3.2 Uniqueness of Iris Patterns

The degrees of freedom, DOF [8], is also used to evaluate the performance of any iris biometric system. DOF indicates the uniqueness of the iris templates and it is computed by using the mean and standard deviation of the interclass hamming distance distribution, as given in Eq. (12). A larger DOF indicates a better performance at a very small false acceptance rate.

$$N = \frac{p(1-p)}{\sigma^2} \tag{12}$$

where N is the degrees of freedom, p is the observed hamming distance mean and σ is the standard deviation.

With the 1060 iris images from the SDUMLA-HMT multimodal database for experimental analysis, the total number of unique iris pairs whose hamming distance (HD) could be computed was $(((1060 * 1060) - (106 * 25))/2) = 5,60,475$, by avoiding the same iris comparisons and not double-counting iris pairs. The distribution of the interclass hamming distances obtained from 560, 475 unique comparisons between different pairings of iris images in the SDUMLA-HMT database is shown in Fig. 5. The observed mean HD was $p = 0.47$ and standard deviation $\sigma = 0.027$ which results in $N = 341$ degrees of freedom. This indicates that the proposed approach maps the irises into 2^{341} distinct possibilities and it indicates the good performance of the proposed approach.

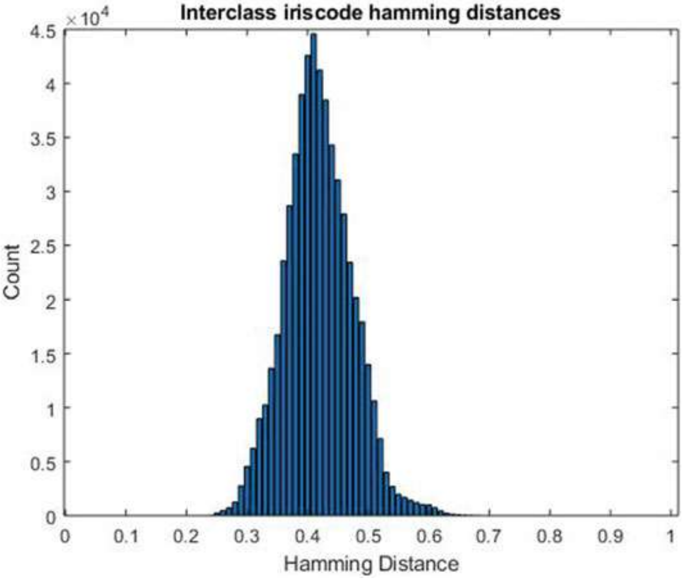


Fig. 5 Interclass distribution of Hamming distances

5 Conclusion and Future Scope

An iris recognition system with improved performance using a combination of integer wavelet transform and log energy entropy is presented. The proposed approach achieved an identification accuracy of 100% with the absence of FAR, whereas verification accuracy is 99.8% with a very low FAR. With the use of block-based IWT applied to the normalized iris image, a good set of local features (textures) are extracted. Among the various entropies, it is found that log energy entropy is more representative of blocks of iris images as it has more significant and better discrimination capability. The future work includes incorporating the iris biometric into the hardware circuit of the professional camera to identify the person who has taken the photograph.

References

1. Adamović, S., Miškovic, V., Maček, N., Milosavljević, M., Šarac, M., Saračević, M., Gnjatović, M.: An efficient novel approach for iris recognition based on stylometric features and machine learning techniques. Future Gener. Comput. Syst. **107**, 144–157 (2020)
2. Bhateja, A.K., Sharma, S., Chaudhury, S., Agrawal, N.: Iris recognition based on sparse representation and k-nearest subspace with genetic algorithm. Pattern Recogn. Lett. **73**, 13–18 (2016)
3. Calderbank, A.R., Daubechies, I., Sweldens, W., Yeo, B.L.: Wavelet transforms that map integers to integers. Appl. Comput. Harmonic Anal. **5**(3), 332–369 (1998)
4. Chen, J.S., Huertas, A., Medioni, G.: Fast convolution with Laplacian-of-Gaussian masks. IEEE Trans. Pattern Anal. Mach. Intell. **4**, 584–590 (1987)
5. Chen, C.H., Chu, C.T.: High performance iris recognition based on 1-D circular feature extraction and PSO-PNN classifier. Expert Syst. Appl. **36**(7), 10351–10356 (2009)
6. Chen, J., Shen, F., Chen, D.Z., Flynn, P.J.: Iris recognition based on human-interpretable features. IEEE Trans. Inf. Forensics Secur. **11**(7), 1476–1485 (2016)
7. Das, A.B., Bhuiyan, M.I.H.: Discrimination and classification of focal and non-focal EEG signals using entropy-based features in the EMD-DWT domain. Biomed. Sig. Process. Control **29**, 11–21 (2016)
8. Daugman, J.G.: High confidence visual recognition of persons by a test of statistical independence. IEEE Trans. Pattern Anal. Mach. Intell. **15**(11), 1148–1161 (1993)
9. Dua, M., Gupta, R., Khari, M., Crespo, R.G.: Biometric iris recognition using radial basis function neural network. Soft Comput. **23**(22), 11801–11815 (2019)
10. Ekici, S., Yildirim, S., Poyraz, M.: Energy and entropy-based feature extraction for locating fault on transmission lines by using neural network and wavelet packet decomposition. Expert Syst. Appl. **34**(4), 2937–2944 (2008)
11. Fernandez, J., Annapoorani, G., Krishnamoorthy, R.: A simple boundary extraction technique for irregular pupil localization. Int. J. Inf. Technol. Comput. Sci. Perspect. **2**(1), 370–376 (2013)
12. Göksu, H.: Ground moving target recognition using log energy entropy of wavelet packets. Electron. Lett. **54**(4), 233–235 (2018)
13. Göksu, H.: BCI oriented EEG analysis using log energy entropy of wavelet packets. Biomed. Sig. Process. Control **44**, 101–109 (2018)
14. Jincy, J.F., Nithyanandam, P., Chavali, R., Appalaghe, A.K.: Image theft identification using biometric feature. Special issue: Advances in Smart Computing and Bioinformatics, Asian J. Pharm. Clin.Res., pp. 385–389 (2017)

15. Fernandez, J.J., Pandian, N.: Fingerprint core point detection using Connected component approach and Orientation map edge tracing approach. Int. J. Biometrics (in press)
16. Johny, A., Fernandez, J.J.: Breast cancer detection in mammogram using fuzzy C-means and random forest classifier. Int. J. Sci. Res. Sci. Eng. Technol. **4**(8), 312–321 (2018)
17. Kapadia, A.M., Pandian, N.: Reversible data hiding methods in integer wavelet transform. Int. J. Inf. Comput. Secur. **12**(1), 70–89 (2020)
18. Karthiga, R., Narasimhan, K.: Automated diagnosis of breast cancer using wavelet based entropy features. In: Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), pp. 274–279 (2018)
19. Kaudki, O., Bhurchandi, K.: A robust iris recognition approach using fuzzy edge processing technique. In: 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), IEEE, pp. 1–6 (2018)
20. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95-International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995)
21. Kumar, M., Pachori, R.B., Acharya, U.R.: Automated diagnosis of atrial fibrillation ECG signals using entropy features extracted from flexible analytic wavelet transform. Biocybern. Biomed. Eng. **38**(3), 564–573 (2018)
22. Minaee, S., Abdolrashidi, A., Wang, Y.: Iris recognition using scattering transform and textural features. In: IEEE Signal Processing and Signal Processing Education Workshop (SP/SPE), pp. 37–42 (2015)
23. Nalla, P.R., Kumar, A.: Toward more accurate iris recognition using cross-spectral matching. IEEE Trans. Image Process. **26**(1), 208–221 (2016)
24. Patil, P., Vasanth, K.: Iris recognition using local and global iris image moment features. In: Innovations in Power and Advanced Computing Technologies (i-PACT), vol. 1, p. 1–5 (2019)
25. Păvăloi, L., Niţă, C. D.: Iris recognition using SIFT descriptors with different distance measures. In: 10th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), pp. 1–4 (2018)
26. Pedersen, S.J.K.: Circular hough transform. Vision, graphics, and interactive systems. Aalborg University 123(6) (2007)
27. Proenca, H.: Iris recognition: on the segmentation of degraded images acquired in the visible wavelength. IEEE Trans. Pattern Anal. Mach. Intell. **32**(8), 1502–1516 (2009)
28. Specht, D.F.: Probabilistic neural networks. Neural Netw. **3**(1), 109–118 (1990)
29. Trokielewicz, M., Czajka, A., Maciejewicz, P.: Post-mortem iris recognition with deep-learning-based image segmentation. Image Vis. Comput. **94**, 103866 (2020)
30. Umer, S., Dhara, B.C., Chanda, B.: Iris recognition using multiscale morphologic features. Pattern Recogn. Lett. **65**, 67–74 (2015)
31. Yin, Y., Liu, L., Sun, X.: SDUMLA-HMT: a multimodal biometric database. In: Chinese Conference on Biometric Recognition, pp. 260–268 (2011)
32. Tisse, C.L., Martin, L., Torres, L., Robert, M.: Person identification technique using human iris recognition. In: Proceedings of Vision Interface, pp. 294–299 (2002)
33. Wildes, R.P.: Iris recognition: an emerging biometric technology. Proc. IEEE **85**(9), 1348–1363 (1997)
34. Zhang, J. S., Chen, C. J.: Local variance projection log energy entropy features for illumination robust face recognition. In: International Symposium on Biometrics and Security Technologies, pp. 1–5 (2008)
35. Zhao, T., Liu, Y., Huo, G., Zhu, X.: A deep learning iris recognition method based on capsule network architecture. IEEE Access **7**, 49691–49701 (2019)

Deep Learning-Based Approach for Skin Burn Detection with Multi-level Classification

Jagannatha Karthik, Gowrishankar S. Nath, and A. Veena

Abstract In most recent years, convolutional neural network (CNN) model is the detail of craftsmanship form fruitful for photograph investigation. In this exploration, we are incorporating CNN models for classification of skin burn based on visual investigation. The aim of this paper is to develop a computerized mechanism in classifying the burn based on severity and compare the accuracies of various CNN algorithms for the same. Rapid development in deep learning enables automated learning of semantics, deep features that are easily learnt which addresses the problems of existing traditional image processing. The proposed method uses deep neural network, recurrent neural network and CNN model. The training is performed using dataset of 104 images classified into degree 1, degree 2 and degree 3 depending on the severity of the burn. Experimental analysis is also provided to compare the accuracies of different methods and identify the best model with better accuracy. The proposed computerized model can aid the medical experts in diagnosing the wound and suggest appropriate treatment depending on the severity of the skin burn. The proposed model could encourage telemedicine practise with the help of modern technology to remotely diagnose the patients especially in rural areas where there could be shortage of physicians.

Keywords Convolution neural network · Deep neural network · Recurrent neural network · Degree of skin burn

J. Karthik (✉) · G. S. Nath · A. Veena
Department of Computer Science and Engineering, Dr. Ambedkar Institute of Technology, Bengaluru, Karnataka 560056, India
e-mail: karthik.j30@gmail.com

G. S. Nath
e-mail: gowrishankarnath@acm.org

A. Veena
e-mail: veenaa1@acm.org

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_3

1 Introduction

The human skin is the outer covering of the body which has up to seven layers of tissues that provide both protection and receive sensory stimuli from external environment. Skin constitutes up to 15% of total weight of the body and regulates the body temperature. The term consume in this paper is referred as burnt area of the skin due to some accidental mishaps that causes annihilation of pores and skin and connecting tissues. It is one of the most common and accidental inconveniences.

After consume injury, the principal medical care is required, as quickly as time permits. The main treatment is done dependent on the profundity and seriousness of the burnt component. Consume zone, force and spot are the crucial components in deciding the seriousness of consumes as shown in Fig. 1. So, it is fundamental to perceive the different types of pores and skin and the arrangement of the skin to choose the seriousness of the injury. The external slenderer piece of the pores and skin is called an epidermis, while the dermis is the thick interior layer.

It is made out of versatile filaments of connective tissue. Generally, shallow dermal consume, profound dermal consume and full-thickness consume are the three styles of consumes, and it is fundamental to make the qualification among those three sorts of consumes [2, 3]. Shallow dermal consume influences the external piece of the pores and skin or epidermis. The influenced component seems red, excruciating and dry. It can likewise have rankled. Profound dermal consumes influence the dermis and a piece of the dermis layer of skin. The consume segment appears to be red, rankled and agonizing as depicted in Fig. 1. Full-thickness consumes obliterate every epidermis and dermis and may likewise go into the subcutaneous tissue. The consume influenced part seems white or charred. In this examination, we comprehend on skin consume pictures.

The objective of these investigations is to develop the automated mechanism for detecting and classifying the consume based on the severity which is dependent on depth of the wound. The consume is frequently experienced in a mishap which is brought about by different triggers comprising sun presentation, warmth exposure,

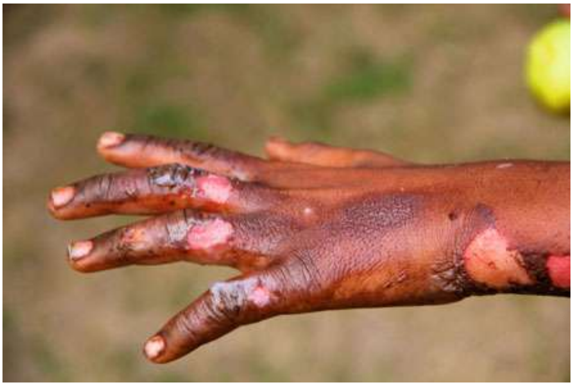


Fig. 1 Consume sample [1]

gadgets, on account of erosion, science. Skin includes epidermis, dermis, fat and muscle. However, to order the degree of skin consumes, we need to investigate and symptomatic test the power of consume wounds. The clinical specialists based on their experience routinely characterize the level of consume into three levels: recognition I, degree II and degree III.

This paper is organized as follows. In Sect. 2, a brief introduction on human skin and classification of skin consume are presented. In Sect. 3, literature survey containing reviews of different papers relevant to current topic is presented. The proposed method is presented in Sect. 4 containing details of specific approach. Results and discussions are presented in Sect. 5, and in the last section, the conclusion is presented.

2 Literature Survey

In the paper of Badea et al. [4], the author has proposed a strategy for consume picture recognition. According to the homes of an entire enveloping patch, they have completed the pixel-wise technique. This technique is computationally expensive and furthermore tricky to commotion. Subsequently, their classifier is not fit for offering enhanced outcomes. It likewise requires supplemental clinical measurements which bolster for order. Causing the computerized to consume characterization, King et al. [5] have incorporated the valuable frequencies in this paper and tune the gadget for computerized consume type. They moreover expect fate porcine consume tests.

As the porcine consume information base is a solid base, consequently it is anything but difficult to guide to in the long run investigate the primary running structure in a clinical body for occasions and act Receiver Operating Characteristic (ROC) perceptions to quantify the assistance that the mechanical assembly presents clinicians in clinical determination making and analysis. Multispectral imaging (MSI) is executed by method of Li et al. [6, 7] to analyse consume injury. It helps consume specialists for arranging consume tissue for the consume debridement medical procedure.

Their calculation broadly extended consume model sort precision. In the paper [7], Rangaraju et al. have characterized the clinical strategy to decide styles of consume, a level of consume and uniting procedure. Because of warmth on pores and skin, unique structure of pores and skin changes, and thickly pressed shape appears on the pores and skin.

Thickening (coagulation) is used to degree the level of consume. Collagen proportion underneath 0.35 is considered as shallow consume, and the proportion somewhere in the range of 0.35 and 0.65 is considered as profound dermal consume, though a proportion more noteworthy than 0.65 is considered as complete thickness consume. Sabeena and Kumar have given a cautious way the utilization of SVM for consume personality and its division [8]. The primary idea in the rear of their compositions is to get pleasantly consume or followed up on process [9]. Their framework utilizes an assortment of strategies for the photograph getting and pre-handling. It points of fact layouts and concentrates the consume from the pictures, anyway the precision is not sufficient.

3 Proposed Method

The objective of this paper is to develop the automated mechanism to detect and classify the skin burn using convolutional neural network (CNN) as shown in Fig. 2. The different stages of the proposed method have been depicted in Fig. 3 shown.

The accuracies of different CNN models are calculated, tabulated and compared on a finite dataset of 104 Web sourced .jpg images. The accuracies of all the models vary each time the models are trained on same dataset due to random shuffling. This model primarily identifies the area of burn in the consume picture and then enhances the quality of the image using the existing library methods. Then, total area burnt is calculated using total area in the consume picture to the segmented area considered as burnt area. Three different variants of CNN model used are deep neural network (DCNN), recurrent CNN (RCNN) and CNN models. Each model is trained using the same dataset with epoch set to 30, and accuracies are calculated and compared to find the best model.

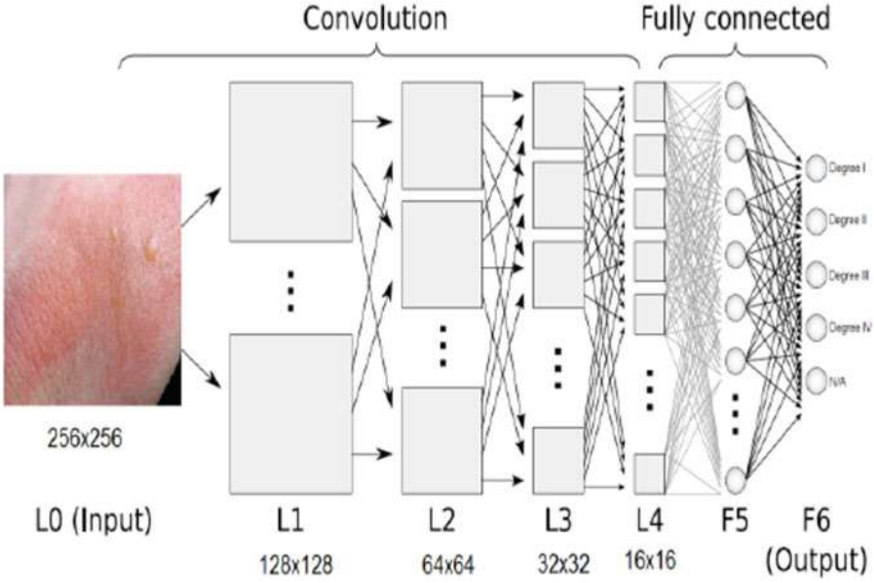


Fig. 2 CNN for burn analysis [4]

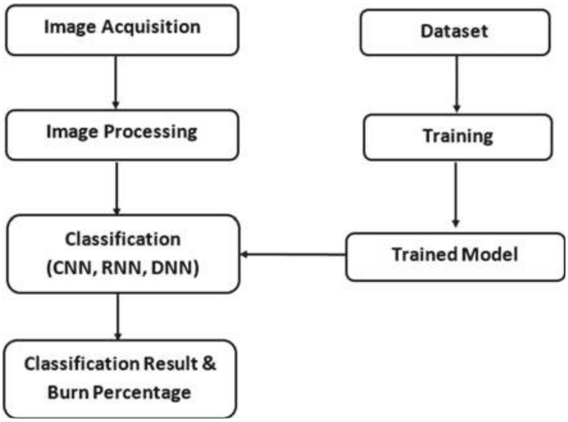


Image Acquisition
Image Processing
Classification (CNN, RNN, DNN)
Classification Result & Burn Percentage
Dataset
Training
Trained Model

Fig. 3 Block diagram

3.1 Convolutional Neural Networks

CNNs principally centre around the possibility that the data might be comprised of pictures. This centres the engineering to be establishment in way to fine match the requirement for dealing with the specific type of data.

CNNs comprise four types of layers once the consume picture is available in the information layer. These are input layer, convolutional layers, pooling layers and fully connected layers. At the point when those layers are stacked, CNN design has been shaped as shown in Fig. 4.

1. The input layer is the input to artificial neural network which will store the pixel values of the consume picture in the form of convolutional matrix. The process involves local binary pattern (LBP) algorithm which is an efficient texture operator which labels each pixel of an image to its neighbour if within its threshold and helps to eliminate the need to store each trivial and similar pixels, thereby enhancing the robustness and achieving computational simplicity.

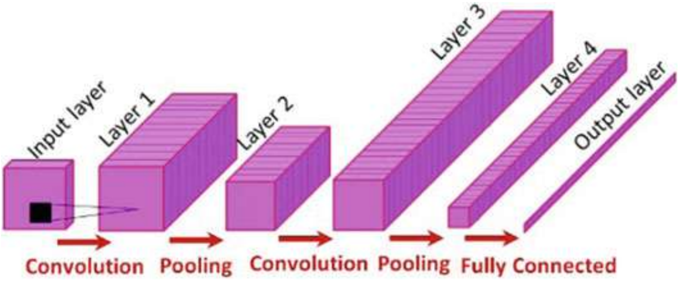


Fig. 4 CNN stages [2]

2. Convolutional neural networks layer will apply a filter to an input consume image to create a feature map that summarizes the presence of automatically detected features in the input by deciding the yield of neurons of which are connected to neighbourhoods the contribution through the count of the scalar product among their loads and the zone identified with the information volume. The redressed direct unit (usually abbreviated to ReLu) expects to apply a detail insightful actuation includes comprising sigmoid to the yield of the enactment created through the previous layer.
3. The pooling layer of CNN functions to reduce the spatial dimensionality of the given information, further diminishing computation in the network that operate on each feature map independently inside that of activation.
4. CNN begins with convolution and pooling layer, and result of this process is fed into fully connected neural network layer that drives the final classification decision.

4 Results and Discussion

The images in Figs. 5, 6 and 7 are being classified as first, second and third degree consume, respectively.

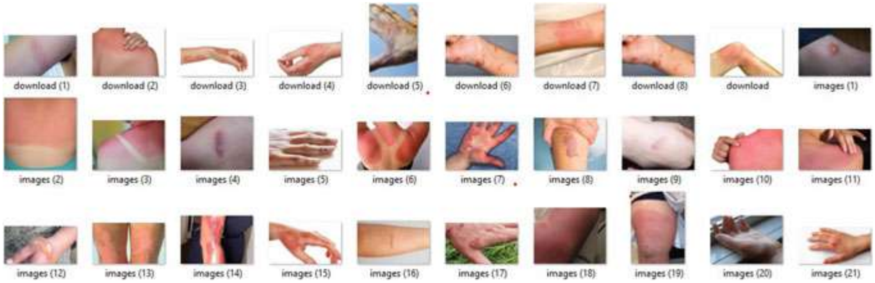


Fig. 5 First degree consume

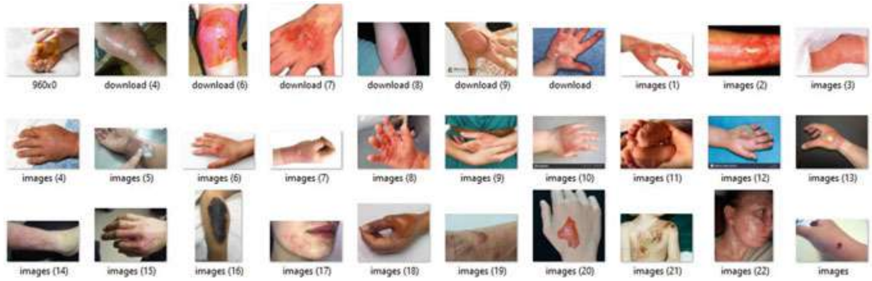


Fig. 6 Second degree consume

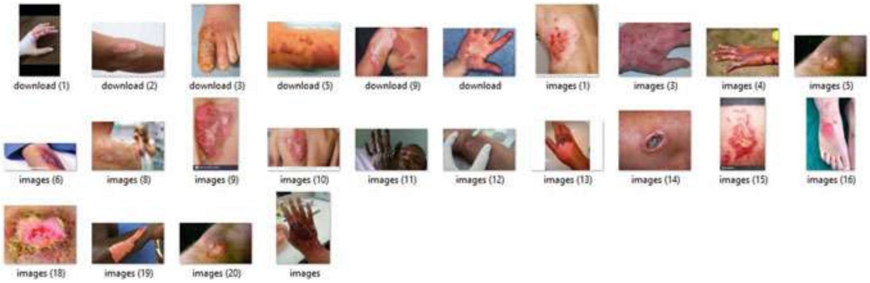
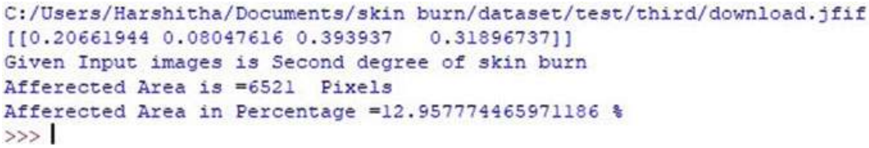


Fig. 7 Third degree consume



```
C:/Users/Harshitha/Documents/skin burn/dataset/test/third/download.jfif
[[0.20661944 0.08047616 0.393937   0.31896737]]
Given Input images is Second degree of skin burn
Afferected Area is =6521  Pixels
Afferected Area in Percentage =12.957774465971186 %
>>> |
```

Fig. 8 Output showing consume degree and percentage

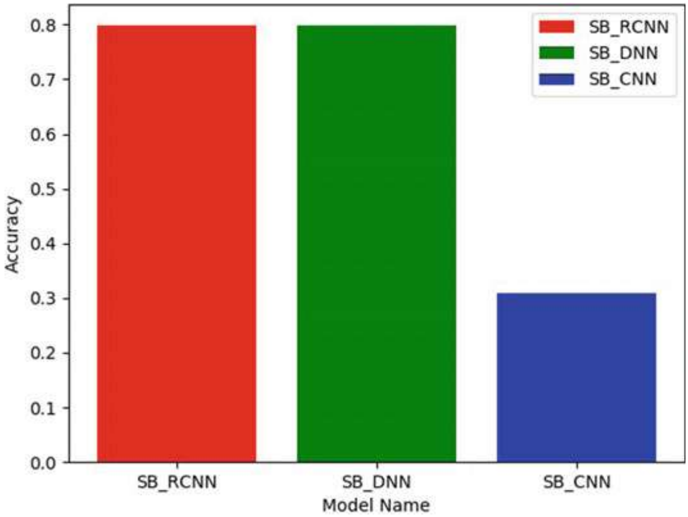


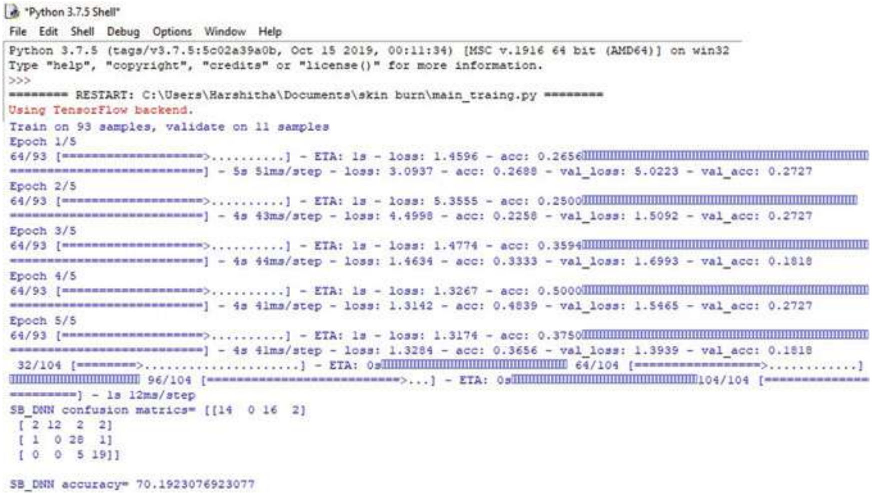
Fig. 9 Bar graph of accuracies of different CNN model

The output screen of the proposed model using CNN is shown in Fig. 8 which clearly showing the degree of the consume and percentage of affected area or the wound area out of the total segmented area.

The accuracies of all the models are plot using the bar graph shown in Fig. 9 and also tabulated in Table 1.

Table 1 Table of accuracy of each method

Method	Accuracy (%)
RCNN	81.4
DCNN	80.02
CNN	30.1



```
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:\Users\Harshitha\Documents\skin burn\main_traing.py ========
Using TensorFlow backend.
Train on 93 samples, validate on 11 samples
Epoch 1/5
64/93 [===================>..........] - ETA: 1s - loss: 1.4596 - acc: 0.2656
==============================] - 5s 51ms/step - loss: 3.0937 - acc: 0.2688 - val_loss: 5.0223 - val_acc: 0.2727
Epoch 2/5
64/93 [===================>..........] - ETA: 1s - loss: 5.3555 - acc: 0.2500
==============================] - 4s 43ms/step - loss: 4.4998 - acc: 0.2258 - val_loss: 1.5092 - val_acc: 0.2727
Epoch 3/5
64/93 [===================>..........] - ETA: 1s - loss: 1.4774 - acc: 0.3594
==============================] - 4s 44ms/step - loss: 1.4634 - acc: 0.3333 - val_loss: 1.6993 - val_acc: 0.1818
Epoch 4/5
64/93 [===================>..........] - ETA: 1s - loss: 1.3267 - acc: 0.5000
==============================] - 4s 41ms/step - loss: 1.3142 - acc: 0.4839 - val_loss: 1.5465 - val_acc: 0.2727
Epoch 5/5
64/93 [===================>..........] - ETA: 1s - loss: 1.3174 - acc: 0.3750
==============================] - 4s 41ms/step - loss: 1.3284 - acc: 0.3656 - val_loss: 1.3939 - val_acc: 0.1818
 32/104 [========>.....................] - ETA: 0s 64/104 [=================>............]
 96/104 [==========================>...] - ETA: 0s104/104 [===============
==========] - 1s 12ms/step
SB_DNN confusion matrics= [[14  0 16  2]
 [ 2 12  2  2]
 [ 1  0 28  1]
 [ 0  0  5 19]]

SB_DNN accuracy= 70.1923076923077
```

Fig. 10 Output showing training the model using DNN

Out of 104 images in the training dataset, 93 is being used to train, and remaining 11 is used for testing the model, and accuracy is calculated for each model as shown in Fig. 10 with epoch set to 5. The acccuracy for RNN, DNN and CNN is 81.4%, 80.02% and 30.1%, respectively.

The graph in Fig. 11 shows the model accuracies and loss of training and validation dataset.

5 Conclusion

In this proposed method of classifying the consume, we have used three variants of convolutional neural network (CNN), i.e. deep neural network (DNN), CNN and recurrent neural network (RNN). RNN model has the better accuracy than compared to DNN and CNN models. The consume analysis by the above three different models of CNN will assist the doctors in taking the proper medical decision for the treatment of the skin consume.

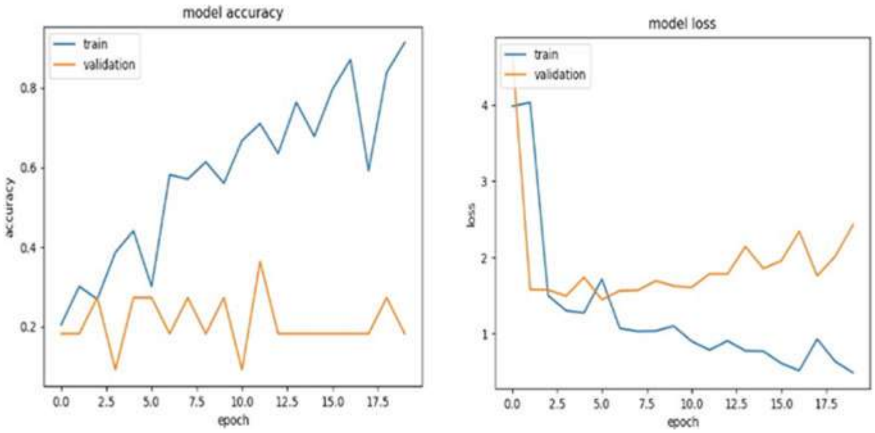


Fig. 11 Model accuracy and loss in training and validation

Apart from classification, accuracies of prediction by the models and percentage of consume in the picture are also calculated. The model with highest accuracy could be identified as the better algorithm.

References

1. Skin burn treatment image at https://dermatology.co.za/patient-area/skin-burn-treatment/
2. Jaskille, A.D., Shupp, J.W., Jordan, M.H., Jeng, J.C.: Critical review of burn depth assessment techniques: Part I. Historical review. J. Burn Care Res. **30**(6), 937–947 (2009)
3. Atiyeh, B.S., Gunn, S.W., Hayek, S.N.: State of the art in burn treatment. World J.Surg. **29**(2), 131–148 (2005)
4. Badea, M.-S., Vertan, C., Florea, C., Florea, L., Badoiu, S.: Automatic burn area identification in color images. In: Proceedings of International Conference Communications (COMM), pp. 65–68 (2016)
5. King, D.R., et al.: Surgical wound debridement sequentially characterized in a porcine burn model with multispectral imaging. Burns **41**(7), 1478–1487 (2015)
6. Li, W. et al.: Burn injury diagnostic imaging device's accuracy improved by outlier detection and removal. In: Proceedings of SPIE, vol. 9472, Art. no. 947206 (2015)
7. Rangaraju, L.P., Kunapuli, G., Every, D., Ayala, O.D., Ganapathy, P., Mahadevan-Jansen, A.: Classification of burn injury using Raman spectroscopy and optical coherence tomography: an ex-vivo study on porcine skin. Burns **45**(3), 659–670 (2019)
8. Sabeena, B., Kumar, P.D.R.: Diagnosis and detection of skin burn analysis segmentation in colour skin images. Int. J. Adv. Res. Comput. Commun. Eng. **6**(2), 369–374 (2017)
9. Haller, H.L., Giretzlehner, M., Dirnberger, J., Owen, R.: Medical documentation of burn injuries. In: Jeschke, M.G., Kamolz, L.P., Sjöberg, F., Wolf, S.E. (Eds.) Handbook of Burns. Springer, Vienna, Austria (2012)
10. Samala Ravi, K., et al.: Deep-learning convolution neural network for computer-aided detection of microcalcifications in digital breast tomosynthesis. International Society for Optics and Photonics, Medical Imaging (2016)

11. Tran, H.S., Le, T.H., Nguyen, T.T.: The degree of skin burns images recognition using convolutional neural network. Indian J. Sci. Technol. **9**(45), 1–6 (2016). https://doi.org/10.17485/ijst/2016/v9i45/106772
12. Kuan, P.N., Chua, S., Safawi, E.B., Wang, H.H., Tiong, W.: A comparative study of the classification of skin burn depth in human. J. Telecommun. Electron. Comput. Eng. (JTEC) **9**(2), 15–23 (2017)
13. Rangayyan, R.M., Acha, B., Serrano, C.: Color image processing with biomedical applications. SPIE Press, Bellingham, USA (2011)
14. Jaskille, A.D., Shupp, J.W., Jordan, M.H., Jeng, J.C.: Critical review of burn depth assessment techniques: Part I Historical review. Burn Care Res. **30**(6), 937–47 (2009)
15. Hardwicke, J., Thomson, R., Bamford, A., Moiemen, N.: A pilot evaluation study of high resolution digital thermal imaging in the assessment of burn depth. Burns **39**(1), 76–81 (2013)
16. Yadav, D.P., Sharma, A., Singh, M. and Goyal, A.: Feature extraction based machine learning for human burn diagnosis from burn images. IEEE J. Transl. Engi. Health Med. (2019)
17. Serrano, C.: Features identification for automatic burn classification. Burns **41**(8), 1883–1890 (2015)
18. Gonzalez, R.C., Woods, R.E.: Digital image processing, 3rd ed. Prentice Hall, Pearson, New Jersey (2008)
19. Burns BIP US database, http://personal.us.es/rboloix/ Burns BIP US database.zip
20. Devlin, J.B., Herrmann, N.P.: BoneColour. Elsevier BV (2015
21. Alidoost, F., Arefi, H.: Application of deep learning for emergency response and disaster management conference Paper (2018)
22. Machine Learning in Python, https://scikit-learn.org

Semantic Retrieval of Microbiome Information Based on Deep Learning

Joshy Alphonse, Anokha N. Binosh, Sneha Raj, Sanjay Pal, and Nidheesh Melethadathil

Abstract Pathogenic microorganisms are always a challenge when they form biofilms on submerged surfaces such as pipes, drains, or sewers, which are difficult to remove using normal chemical or biological treatments. Developing a fundamental understanding of the biodiversity of sewage microbiome or finding out the key species that can be targeted to significantly reduce the pathogenic population within can be critical in advancing and optimizing the technology for maintaining environmental health. Hence to find articles with relevant information about this microbiome and the interactions within is like finding a needle from the haystack. There comes the need for data mining tools, a key part of such a tool would be named entity recognition. To train a NER model, a relevant dataset with the required entities tagged is required and no such were to be found in the biomedical domain. So, in our study, we intended to develop a microbiome dataset with all the relevant concepts tagged for training a NER model which is to be a part of a semantic information retrieval tool. For this, we engineered a dataset specifically focusing on keywords related to the characteristics of the wastewater microbiome that could cluster out the relevant information from the bulk data of PubMed literature. The new engineered data was then used for fine-tuning NER models with different variants of BERT models for analyzing which had the most efficiency with our dataset. We implemented NER models capable of accurately predicting the concepts tagged in the microbiome dataset and designed experiments to validate the efficiency of the different models on our dataset and also other open-source biomedical datasets like JNLPA and BC5CDR. The results show that out of the three BERT variants, BioBERT was the most performant model, and also even with a fairly limited size compared to other biomedical NER datasets, we were able to achieve similar scores. The NER model fine-tuned using the microbiome dataset was able to successfully predict the tagged concepts/named entities in the datasets.

Keywords Deep learning · BioNLP · BERT · NER · Microbiome

J. Alphonse · A. N. Binosh · S. Raj · S. Pal · N. Melethadathil (✉)
School of Biotechnology, Amrita Vishwa Vidyapeetham, Amritapuri Campus, Coimbatore, India
e-mail: nidheesh@am.amrita.edu

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_4

1 Introduction

The biological wastewater treatment techniques are the issue of concern we are pondering, in which we focus on the method where the keystone bacterial species in sewage is knocked out. The keystone bacterial species are what we consider the harmful dominant one within the sewage environment which are hard to remove from the system and the one which builds the base for biofilm, which can harbor even more harmful pathogens. The species can be disposed of by utilization of both concoction or organic anti-infection agents/bacteriocins or by the utilization of explicit bacteriophage focusing on just the keystone species. In any case, it is likewise imperative to secure and advance the development of those species which helps in the remediation of wastewater/sewage. To discover all the conceivable keystone species that add to this pathogenic bacterial settlement is a demanding errand. Furthermore, numerous earlier exploration works have uncovered numerous potential species. This expanding abundance of writing in the natural area makes it hard for specialists to stay up with the latest with continuous exploration. Refining related information from scholastic articles genuinely is exorbitant and moderate, if at all the full substance is given to the analyst. Many web search tools, for example, PubMed search permits clients to inquiry writing dependent on words or expressions present in the archive. But the outcomes will be shown directly with a colossal number of comparable indexed lists and making sense of the true data from each of those is a humongous task. That acquires the requirement for a semantic information retrieval tool.

Semantic information retrieval frameworks intend to broaden the exemplary information retrieval approach. As the capacity to recover significant data is at the core of each part of innovative work in the existence sciences industry and as information being regularly dispersed over different systems and recorded such that makes it hard to bits together with the total picture also, besides the ability to tap into these diverse information sources, the user interfaces should also succor users in filtering out irrelevant information and featuring the key relationships hidden within an aggregated array of information. These requirements in information retrieval shall be confronted with the semantic information retrieval tools built on burgeoning machine learning algorithms that excavate documents for categorizing information, and model user interests to help them define queries, lessen the burden, and can guarantee more progressively predictable behavior.

In the booming field of machine learning, we are constantly trying to make the machines think, decipher, and behave like humans and improve their learning progressively in a self-sufficient fashion by feeding them data [1]. The machines can store and process more data much faster than the human, and they not only just learn, but can predict things by understanding the patterns in them. The ability to learn without comprehensive programming is achieved by computer via machine learning. Deep learning is a new area that is a specialized form of machine learning focused on making efficient futuristic models which will make human tasks easier [2]. Here, the model is trained by utilizing a huge set of labeled data and neural network (NN) architectures that contain many layers so that it can progressively extract more significant level highlights from the crude information, each layer learns to transport its input data into a more abstract and composite level. Deep learning models are nowadays widely used for NER-based systems. NER is a remarkable framework that is capable of exploiting the advantages of semantic information, and NER is a core task of natural language processing (NLP) which consists in detecting and classifying named entities (NE) from the free, unstructured text source. Named entities are typically characterized as real-world genuine ideas that can be alluded through unbending markers.

In this paper, we address this problem by improvising a powerful machine learning algorithm, a BERT-based NER model, which is explicitly programmed for biological domain data that is employed to detect and classify predetermined groups of entities from literature thereby retrieving authentic information from the massive load articles given.

The model we are looking at is a SciBERT model pre-trained on multiple domains NER datasets and then fine-tuned on target datasets. The transfer learning method is used to fill in the limitation created by the small amount of target data, and we use BioNLP techniques like named entity recognition (NER) for developing our model for automated information retrieval (IR) from unstructured data (research articles).

BIONLP: NLP is one of the most important subfields of machine learning for a variety of reasons. NLP bridges the gap between the human brain and machines. Biomedical natural language processing (Bio-NLP) specifically focuses on the extraction of information targeting entities in biological and medical domains which can help to interpret the prodigious amount of literature. Merged with cutting edge models, these techniques facilitated the research in the biological field considerably.

BERT Model: Bidirectional encoder representations from transformers (BERT) were recently promulgated by researchers at Google AI language. It has created a twitch in the machine learning community by proffering futuristic results in a comprehensive selection of NLP tasks, including question answering (SQuAD v1.1), natural language inference (MNLI), and others. The BERT model works on an attention mechanism that learns semantic correlations between words (or sub-words) in a text.

Applying BERT on different domain-specific areas gave rise to new pre-trained contextualized models SciBERT [3] and Bio-BERT. Both models follow the same architecture as that of the BERT model which is a multi-bidirectional transformer and performs text representation by predicting masked tokens and the next sentence.

BERT models are 16-layer transformer neural networks with close to around 20 million parameters which can be adjusted. And training BERT models from scratch would take around 20 weeks and huge computational resources, so using a pre-trained model was more feasible. All pre-trained BERT models are natural language understanding (NLU) models that can be fine-tuned for any NLP tasks ranging from NER to NEL.

SciBERT: Scientific BERT principally a BERT-based model trained on a large amount of scientific literature is a futuristic model which can be fine-tuned to any particular NLP task of our need. It addresses the shortfall of high quality, large-scale labeled scientific data. Bio-BERT [4] is another BERT-based model which is primarily pre-trained with a huge amount of biomedical corpora and can perform biomedical text-mining tasks more precisely than any other BERT models.

Named Entity Recognition (NER): Named entity recognition encompasses an extensive variety of approaches in the field of natural language processing and information retrieval [5]. NER layers can discover entities from unstructured raw data and categorize them into predetermined categories. As the sentence is introduced to the layer, it scrutinizes and highlights the entities from the text. The NER layer is trained on separate sensitive entities depending upon the task.

2 Related Works

Nowadays, it is increasingly congenial to analyze in a NER model and later store the entities associated with them indelibly than to design a search engine algorithm that searches for a query that we look for over a large indefinite quantity of articles or websites online. Many such models are widely in use. This paper [6] has dispensed a framework, which is profitable in many applications in web search. Through this paper, they came up with a concept that detects the named entity within the query and pinpoints the most probable classes of the named entity. The TurkuNLP [7] is a similarly implemented model which uses BioNLP with a concoction of several long short term memory (LSTM) networks. Also, a language model that makes use of an enormous unlabeled corpus for learning a distributed representation was proposed by Colbert and Weston [8] with an idea of developing a unified architecture that can perform various NLP tasks like POS tagging, chunking, named entity recognition, and semantic role labeling, while the AraBERT model [9] works on Arabic semantic NLP and BERT-based structure. BERN [10] is also such a text-mining tool that aids neural network-based high-performance BioBERT that perceives known entities and locates new entities and recognizes the sorts of overlapping entities. It uses probability-based decision rules for the identification of nested or overlapping entities. Textpresso [11] is a far more advanced text-mining tool for scientific literature than a basic keyword search engine. The categories or classes in textpresso are a collection of different biological concepts, the relation between two objects, or processes all these together forms a type of objects and concepts called an ontology. This information retrieval system described in this paperwork is based on this ontology developed.

DySE system [12] actualizes a setting steered methodology in which the catchphrases are prepared with regards to the data in which they are recovered, to comprehend semantic uncertainty, and to give a progressively precise recovery dependent on client interests.

3 Methodology

In this paper, we scrutinize the use of SciBERT, a BERT-based transformer network pre-trained on medical literature as a NER model for identification of precise information related to the topic of microbial and environmental interaction in wastewater management or treatment. As the task (NER) is a sequence labeling, i.e., a sequence of words or text is classified into entity categories, to achieve this objective we primarily need two components: (1) Microbiome Dataset (2) NER Model.

3.1 Microbiome Dataset

The corpus for the creation of the dataset was collected from several research articles that expressed information about our topic of interest from several sources like PubMed, semantic scholar, etc., i.e., the articles cover topics such as wastewater treatment, various bacteria species which survive in the sewage, their optimum pH conditions, the biological oxygen demand (BOD), and chemical oxygen demand (COD) of the wastewater and experiments on those species of bacteria which are or likely to be a keystone species.

To retrieve the paper information, we used bio-python and semantic scholar python package that provides an API interface to databases for easy parsing and storing of the information. The retrieved corpus we then pre-processed using the NLTK sentence and word tokenization for further manual annotation. For the manual annotation, we followed the BIO tagging scheme. For the development of the dataset, we need to identify and segregate the information into different entity classes on doing so we were left with four entity classes. The different tags used in the current situation are bacterial species (BAC), BOD (includes BOD, COD, pH, and temperature conditions), habitat, and others (Biological wastewater treatments, metabolic reactions, etc.). Also, all the other non-entities were tagged as O. Along with these tags, the prefixes ‘B’ and ‘I’ were also given indicating the beginning and intermediate portions of the entity as per the BIO schema. For example, in the case of *Escherichia coli*, *Escherichia* will be tagged as B-BAC, and *coli* will be tagged I BAC, as, during sentence tokenization, each word independent of their meaning is separated.

The whole corpus was later separated into two parts with 75% as training and 25% for testing. The corpus was then manually annotated and verified by experts in the relevant field.

3.2 NER Model

For fine-tuning, we used uncased versions of the BERT models [3, 4, 13] as it mostly performed better than cased in most datasets in other studies. The model architecture and hyperparameters were followed as described in Devlin et al. [13] for NER. The tokenized text was transformed into their representative Bert tokens Ids and fed to the Bert embedding layer which outputs the corresponding Bert vector for the token, which is then fed to a linear classification layer for predicting the probability of that token for each of the individual entity types.

For training, we used Bert Adam [14] as the learning optimizer with a learning rate of 2e-5 with an epoch of 5. The dropout for the BERT vectors was set to 0.01, and cross-entropy loss of the NER outputs was calculated for optimizing the model during training.

3.3 Evaluation Metrics

The performance of the model has been evaluated using experimental metrics such as R (recall), P (precision), and $F1$ ($F1$-score).

$$\text{Precision} = \frac{TP}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$F1\text{Score} = \frac{\text{2PR}}{\text{P} + \text{R}}$$

In the above equation, true positive (TP) indicates the total number of correctly identified named entities, false positive (FP) is the total number of incorrectly identified named entities, false negative (FN) is the number of non-entities that are incorrectly identified as named entities by the model. P represents precision and R represents recall. $F1$ score is the balanced mean of precision and recall.

4 Experimental Result and Discussion

4.1 Baseline

To evaluate the performance of our implemented model, we did a baseline comparison of all three different BERT models with two different open-sourced biomedical NER datasets and compared it with the reported SOTA scores. While the implementation did not yield SOTA scores as seen in Table 2, they were relatively close to them and decreases in F1 scores can be attributed to the use of additional layers like CRF in the NER model in the SciBERT paper which will yield better-formed entities as outputs. While adding a CRF layer to the final logit/linear layer of the model which might have increased the performance, the main aim of the study was to analyze how much of the concepts tagged in the microbiome dataset was the model able to correctly learn and predict, so CRF was not implemented in the final layer.

4.2 Microbiome

From the experimental results shown in Fig. 2, we observe that SciBERT and BioBERT surpass the BERT-base by a significant margin on the microbiome dataset. One can attribute the degradation of the BERT-base performance on the difference of corpus used for pretraining these models. Both SciBERT and BioBERT were trained on a combination of Wikipedia + biomedical literature, whereas BERT-base was trained on Wikipedia + general text. So, in contrast, the biomedical BERT models are more likely to have better BERT embeddings for the named entities tagged in the microbiome dataset.

In order to evaluate the impact of different BERT models on the microbiome dataset, we trained the NER model consisting of the pre-trained BERT layers in a fine-tuning configuration over the training set. Later the fine-tuned model outputs were validated against the test set, and their respective macro $F1$ scores were calculated. The experimental results show that the BioBERT is the top-performing BERT model on the microbiome dataset with an $F1$ score of 75.13 while SciBERT performed slightly worse (-0.3 $F1$ Score). However, on the evaluation of both NER models based on SciBERT and BioBERT, we found out that both of them can be successfully integrated into a text-mining pipeline for the extraction of meaningful information about microbiome from biomedical literature.

5 Conclusion

In this paper, we developed a new dataset for microbiome named entities and implemented a NER model based on the microbiome dataset using the different variants of BERT. Compared to other NER dataset in the biological domain, our dataset is fairly limited but due to the performance of natural language understanding models of language models like BERT, the development of state-of-the-art models for tasks like named entity recognition, relation extraction, and question answering fairly smaller dataset by finetuning is ease due to the superiority of pretraining step involved in the development of these models. We also designed experiments to validate which BERT variants are the most fitting for the microbiome dataset, by comparing the F1 scores of the validation dataset. While the train and test datasets were comparatively limited in scale, we are yet to check whether the increase in tag counts translates to much higher performance in the metrics.

In this paper, for addressing the problem statement of biological wastewater treatment and its information retrieval from biomedical literature, we engineered a new microbiome dataset with all the relevant classes of entities tagged, and NER models with different BERT variants were implemented and trained using the developed dataset. While the limited size of our dataset was used for fine-tuning NER model, we were still able to achieve a respectable F1 score overall due to the superiority of the pretraining step involved in the development of BERT. Compared to the yesteryears models like Bi-LSTM CRF, these scores would not have been possible with this dataset. The experimental results demonstrate that the concepts needed for answering the problem statement of biological wastewater treatment such BOD, microbial names, pH can be tagged using a NER model and also the development of efficient and reliable NER models required large amounts of data is not the case with language models like BERT.

In the future, we will be increasing the amount of tagged data used for fine-tuning and adding a CRF layer for improved efficiency of the model, while an entity relation extraction with a coreference resolution model will also be trained for the extraction of the relation between the different entities mentioned in the literature.

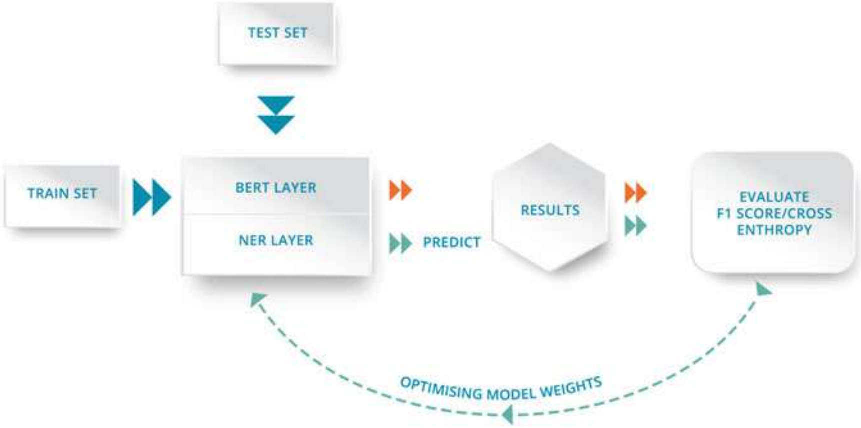


Fig. 1 The architecture of the proposed BioNLP model with named entity recognition

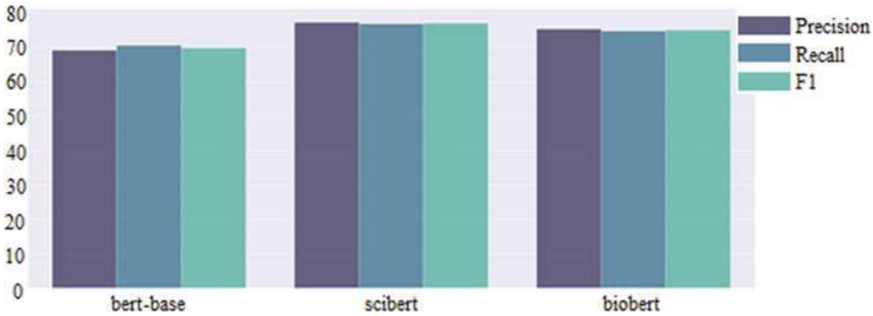


Fig. 2 Graphical representation of evaluation of matrices

Table 1 Distribution of named entities used in the model for training and testing

Named entity tags	Training set: no. of total named entities	Testing set: no. of total named entities
B-BAC	6896	1186
B-BOD	3480	489
B-HABITAT	5197	982
B-OTHERS	844	118

Table 2 Metrics evaluation of BERT-based models on an open- source biomedical dataset

		Reported F1 scores	Our implementation score
JNLPBA	BERT-base	69.57	68.60
	SciBERT-SciVocab	77.28	77.16
	BioBERT-PubMed-PMC	77.59	76.59
BC5CDR	BERT-base	86.72	85.72
	SciBERT-SciVocab	90.01	88.65
	BioBERT-PubMed-PMC	88.85	88.21

Table 3 Metrics evaluation of BERT-based models

	Precision	Recall	$F1$-score
BERT-Base	69.23	70.65	69.93
SciBERT-SciVocab	77.34	76.93	74.83
BioBERT-PubMed-PMC	75.44	74.83	75.13

Acknowledgements This work derives direction and ideas from the Chancellor of Amrita Vishwa Vidyapeetham, Sri Mata Amritanandamayi Devi. The work The authors would like to thank Aswathi Viswanathan, Athul C R, and Archa P for their support in this work.

References

1. Babu, R., Simon, A., Singh Deo, M.: An Overview of Machine Learning and its Applications Medical Image Processing View project Vision based Navigation of Unmanned Air Vehicles View project Venkatesan Selvam Dayananda Sagar Institutions An Overview of Machine Learning and its Applications (2016)
2. Dargan, S., Kumar, M., Ayyagari, M.R., Kumar, G.: A survey of deep learning and its applications: a new paradigm to machine learning. Arch. Comput. Methods Eng. (2019)
3. Beltagy, I., Lo, K., Cohan, A.: SCIBERT: a pre trained language model for scientific text. In: EMNLP-IJCNLP 2019 – 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing Proceeding Conference, pp. 3615–3620 (2020)
4. Lee, J., et al.: BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics **36**, 1234–1240 (2020)

5. Wen, Y., Fan, C., Chen, G., Chen, X., Chen, M.: A survey on named entity recognition. In: Lecture Notes Electrical Engineering, vol. 571 LNEE, pp. 1803–1810 (2020)
6. Alshuwaier, F.A., Almutairi, W.A., Areshey, A.M.: Smart search tools using named entity recognition. In: Proceedings - 2013 International Conference Information Technology Applications ITA 2013, pp. 304–311 (2013)
7. Mehryary, F., Björne, J., Pyysalo, S., Salakoski, T., Ginter, F.: Deep Learning with Minimal Training Data: TurkuNLP Entry in the BioNLP Shared Task 2016, pp. 73–81 (2016)
8. Collobert, R., Weston, J.: A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceedings of the 25th international conference on Machine learning (2008)
9. Antoun, W., Baly, F., Hajj, H.: AraBERT: Transformer-based Model for Arabic Language Understanding
10. Kim, D. et al.: A neural named entity recognition and multi-type normalization tool for biomedical text mining. IEEE Access **7**
11. Müller, H.M., Kenny, E.E., Sternberg, P.W.: Textpresso: an ontology-based information retrieval and extraction system for biological literature. PLoS Biol. **2** (2004)
12. Rinaldi, A.M.: An ontology-driven approach for semantic information retrieval on the web. ACM Trans. Internet Technol. (TOIT) **9**(3), 1–24 (2009)
13. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, vol. 1, pp. 4171–4186 (2019)
14. Kingma, D.P., Ba, J.L.: ADAM: A Method for Stochastic Optimization (2017)

Early Detection of COVID-19 from CT Scans Using Deep Learning Techniques

P. Limna Das, A. Sai Manoj, Sachin Sharma, and P. B. Jayaraj

Abstract The novel coronavirus 2019 (COVID-2019) which began from China, further spread to all over the planet and was announced as a pandemic by WHO. It has blocked our daily lives and world economy to a large extent. In the lack of any particular vaccine for present pandemic COVID-19, it is necessary to recognize the disease at an early stage and quarantine these infected patients to stop the further spread. The popular diagnosis of COVID-19 is being done using polymerase chain reaction (PCR), but there are some cases of false interpretation. Rapid antibody test also has faulty/wrong implications. Till now, we have witnessed the global deficit of testing labs and testing kits for COVID-19. There is an urgent requirement for developing quick and reliable devices that can help doctors in diagnosing COVID-19. Developing a computer-based COVID detection tool will be very useful as it can screen the positive cases from a mass collection. Radiological imaging like computed tomography (CT) scans can be used for the early diagnosis. With the invention of AI algorithms, we can apply learning algorithms for early detection of COVID-19. 2016 on wards, deep learning, a deep neural network-based learning technique is widely applied in biomedical problems. In this article, we suggest a fast and reliable diagnostic tool using deep learning algorithms for identifying this pandemic. We have built two models for this purpose; one with an EfficientNet architecture using focal loss and a GradCam heatmap for testing its reliability in practical use. We also built a model using ResNet by custom vision AI of Microsoft Azure. Data was collected from different sources and the highly scaled EfficientNet architecture outperformed the Resnet architecture of MS Azure for classifying the COVID CT scans by an increase in accuracy of 10%. We are planning to deploy this software in the form of a chatbot. Also, our model continuously learns from data regularly and would attain better accuracy in future.

P. Limna Das · A. S. Manoj · S. Sharma · P. B. Jayaraj (✉)
Department of Computer Science and Engineering, National Institute of Technology Calicut, Kozhikode, India
e-mail: jayarajpb@nitc.ac.in

Institute of Advanced Research, Gandhinagar, India

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_5

Keywords Coronoa virus · CT imaging · Artificial intelligence · Deep learning · Early detection · Neural networks

1 Introduction

A huge explosion of a novel coronavirus, first happened in Wuhan, China which was happened in 2019 December, caused acute respiratory illness in patients. The novel coronavirus was named as SARS-CoV-2, and the diseases were termed as COVID-19 by WHO [15]. This SARS-CoV-2 virus was expanded from Wuhan to other cities of China in 30 days [10, 11]. The initial seven cases in the United States of America had reported in January the month of 2020 and is arrived at a count of 34,79,483 by July 14, 2020. In India, 907,645 patients are infected until today, and 23,727 people are died due to COVID-19. It is reported that the world wide 13,235,760 coronavirus cases are affected by July mid. The main symptoms of COVID-19 are mainly muscle pain, fever, headache, sore throat, cough, and shortness of breath.

The real-time reverse transcriptase-polymerase chain reaction (RT-PCR) is a broadly used laboratory-based testing process for finding COVID-19 [1]. The strict requirement of constructing difficult laboratory environment is delaying the fast diagnosis of the disease in patients. This is actually creating a delay in preventing the spread of the disease. The auxiliary diagnostic tools, if any, will be helpful in this regard to detect the disease early as no accurate methods are available currently. Chest X-rays are not quite suited to detect COVID-19 at initial stages as it is observed that the majority of COVID-19 patients have normal or only mildly abnormal X-ray scans.

For the clinical diagnosis of this disease, computed tomography (CT) scans are considered as a faster and easier treatment technique [9]. In the clinical practice, by joining the patient's clinical symptoms, travel history, and laboratory findings, it is possible to diagnosis the disease quickly. By doing this, we can effectively isolate the infected patients on time and control this pandemic. So, chest CT can be considered as a critical modality for checking the disease [13, 14].

Early detection of COVID-19 has importance in timely quarantining & treating the disease and therefore assuring public health safety. Currently, a bulk number of suspected patients are taking chest CT scans, and undergoing auto COVID detection, removes the overhead of radiologists. Moreover, radiologists visually inspecting the CT scans may cause intra- and inter-observer problems to miss the diagnosis of the disease. With the invention of artificial intelligence (AI) algorithms, many latest learning algorithms have recently gained popularity in the medical field, using which one can build a model of the given training data. But these machine learning algorithms need feature extraction to find the features of the data using which they classify the unknown data.

Deep learning, a newly derived sub-area in AI [12], avoids the need for manual feature extraction that has been successfully employed in medical problems like pneumonia detection on X-ray images, skin cancer classification, and lung segmentation. The rapid rise of this pandemic necessitated us the need of developing expertise in this domain. AI-based automated diagnostic tools are important to be developed. With the limited number of radiologists across the country, this technique will help us to accurately predict the disease and to give more care and assistance to the needy patients. These tools will provide accurate interpretations very fastly. In otherway, these tools can largely balance the insufficient number of available RT-PCR test kits and testing costs. In the ImageNet image classification challenge, deep learning methods surpassed other techniques by training one million images on advanced hardware. In this article, two deep learning models are suggested for the automatic diagnosis of COVID-19. The recommended model will have an end-to-end architecture without using any feature extraction methods, and it needs CT images to give the diagnosis.

In the following sections, the report discusses the literature survey carried out for the project (Sect. 2), the methodology (Sect. 3) describing its design and implementation, followed by the results and conclusions in Sects. 4 and 5.

2 Literature Survey

A lung CT scan can be used for diagnosing pneumonia and other lung diseases [16]. As COVID-19 infects lungs, we identified that a lung CT scan can successfully be used to diagnose in the preliminary stage accurately than the current reverse transcription-polymerase chain reaction(RT-PCR). Different works are reported in this area. Some of them are described below.

Zheng et al. [1] introduced a new method to recognize COVID-19 by applying a 3D deep convolutional neural network on CT images. These authors solved the shortage of data by using augmentation on available CT volumes to generate more number of training data. The use of a spatially global pooling layer and the temporally global pooling layer solved the issue related to weak supervised COVID-19 detection. Some small infected regions of COVID-19 might be missed by professional radiologists by their visual examination using the eye. But this can be managed by taking the benefits of deep learning and by a pre-trained UNet [8] for providing the lung masks to direct the learning of DeCoVNet. The limitations of their study are as follows: The UNet model for lung segmentation could not handle temporal information, it was trained using defective ground-truth masks, the data used in their research came from only a single hospital, and cross-center authentications were not performed.

Tulin et al. [2] suggested a new design for automatic COVID-19 detection utilizing raw lung X-ray images. Their method allowed us to create the exact diagnostics for binary classification (COVID vs. No-Findings) and multi-class classification (COVID vs. Pneumonia vs. No-Findings). Their design set off a classification accuracy of 98.08% for multi-class and 87.02% for binary classes. The DarkNet model was used in their investigation as a classifier. They implemented 17 convolutional layers then introduced different filtering on each layer. A limitation of their study was the inadequate number of COVID-19 X-ray images for the study. Lin Li et al.

[3] offered a 3D deep learning framework named COVID-19 detection neural network (COVNet) to detect COVID-19 using lung CT scans. This machine learning system using a convolutional network model could distinguish COVID-19 and CAP from chest CT images. The main limitation of their work was, (1) firstly, COVID-19 might have similar imaging features as pneumonia caused by different types of viruses (2). The main drawback of all deep learning methods is the absence of interpretability and transparency. This means to determine the output it is not capable of determining what imaging features are being used. Xu et al. [4] suggested a model in which differentiation between COVID-19 and Influenza-A viral pneumonia was done by applying deep learning technology. Feature extraction was done using classical ResNet. The location attention mechanism combined and compared with the network model without having it. In their multi-center case study, they had introduced a novel method that could screen COVID-19 fully automatically using deep learning technologies. Models with location attention can be used to increase the accuracy rate, for classifying COVID-19 with an overall accuracy rate of 86.7% and it can help doctors for screening more reliably.

Amyar et al. [5] proposed image reconstruction method using an encoder and two decoders for classification (COVID vs. non-COVID) using multilayer perception. A 2D U-NET, an encoder-decoder used for reconstruction and segmentation process. They conducted three experiments to estimate their model. They adjusted the hyperparameters and add/remove a task to find the best model using only the training data set. Then compared their model with the state-of-the-art method U-NET to analyze the performance on the segmentation process. Then compared their model with the convolutional neural network trained to perform classification only. This multi-task learning method could improve the segmentation results without having many segmentations ground truths.

Singh et al. [6] suggested a novel deep learning model described using multi-objective differential evolution (MODE) and convolutional neural networks (CNN) for classification of human beings based upon whether they are affected from COVID-19 or not. The MODE algorithm used to upgrade the hyperparameters of CNN. The 20-fold cross-validation is employed for preventing the over-fitting. The actual experimental results reveal that this model surpasses the competitive models, i.e., ANFIS, ANN, and CNN models in terms of accuracy. The model is useful for real-time COVID-19 disease classification from lung CT images. Wang et al. [7] suggested a deep learning system for diagnostic and prediction of COVID-19 using computed tomography which doesn't require any radiologist for annotation. In most of the earlier studies, the region of interest (ROI) was only used for segmenting the lung fields. But in this work, the authors had used a 3-dimensional bounding box of the whole lung. In this study, the authors tried to identify the severity of lung effect and classified severe and mild COVID-19. To improve diagnosing, authors suggested converting CT images of different slice thickness into unified slice thickness, by using a generative adversarial network. Some of the other deep learning-based works can be seen here [9, 12] for more details.

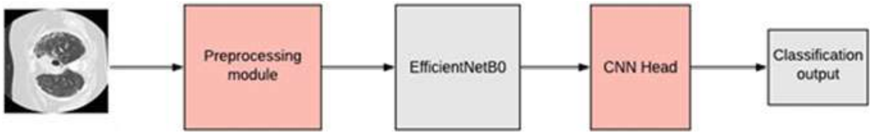


Fig. 1 Work flow diagram of our proposed method

2.1 Motivation

Despite increasing the costly PCR testing lab setup, developing a tool for auto-detecting COVID disease from CT images will be helpful for the society. An open-source framework developed for this purpose will have high social value of helping the poor patient community and by reducing the over burden of doctors.

3 Methodology

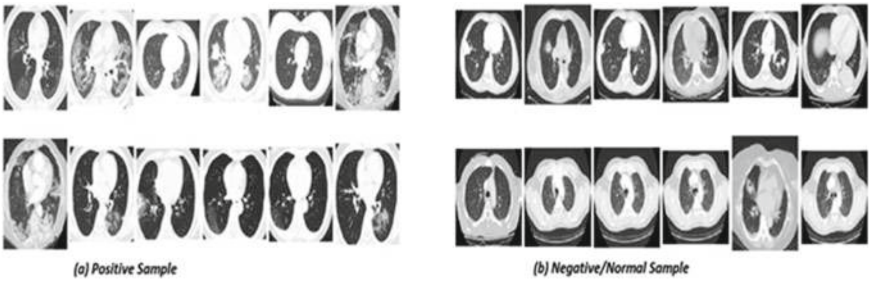
This section describes the workflow of our design method and the details of our proposed model architectures. The main challenge to train a model here was less available data. Since, COVID breakout had happened just few months before, there was less data available for training. To reduce this impact, we relied on pre-processing techniques such as segmenting out the region of interest using BCD U-Net. We have mainly tried two models; one using EfficientNet and the other one using Resnet in an Amazon Azure environment.

3.1 Work Flow of the Design

Our main pipeline is briefly indicated in the diagram shown in Fig. 1. The collected CT images were pre-processed and then passed to the neural network for the final prediction. The details of each section is described below.

Data Collection: The data we have considered is from different sources and have bias in it. We have considered 684 images of 512 × 512 × 3 dimensions. These CT scans are collected from different patients and from different hospitals. The sample data is shown in Fig. 2.

Pre-processing Module: The pre-processing involves resizing and extracting the region of interest (RoI) from each image. As some of the images in the data set contain annotations in the corners, it is important to extract the RoI from the given images. So, we have applied standard BCD U-net architecture, a deep learning model to perform the segmentation task and segment lungs portion from the CT scans. The


(a) Positive Sample

(b) Negative/Normal Sample

Fig. 2 Sample data: COVID positive and negative images

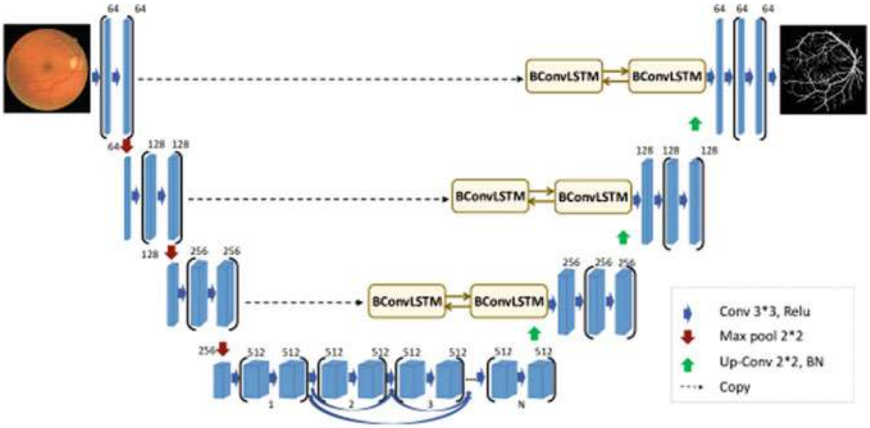


Fig. 3 U-Net architecture [?]

details of BCD U-net are given below. This model was trained on the open-sourced data available on Kaggle.

BCD U-net: is a special variant of U-net. This network utilizes the strengths of both bi-directional ConvLSTM state and densely connected convolutions. The network architecture of BCD U-net is shown in Fig. 3. As every U-net does, BCD U-net is also of an encoder-decoder architecture. The encoding path consists of two 3×3 convolution filters followed by a 2×2 max pooling function and ReLU activation. The number of feature maps is doubled at each step. Ultimately, the encoder path extracts a high-dimensional image representation with high semantic information. Unlike a traditional U-Net, feature maps learned at each stage are concatenated and then forwarded to the next convolution to facilitate better learning procedure. Each step in the decoding path starts with an up-sampling function over the output of the previous layer. Unlike in U-net, a bi-directional ConvLSTM is employed to process corresponding feature maps in a more complex way.

Before passing our data set to the classifier for training, we have used this pre-trained segmentation model to segment out region of interest, i.e., lungs portion and

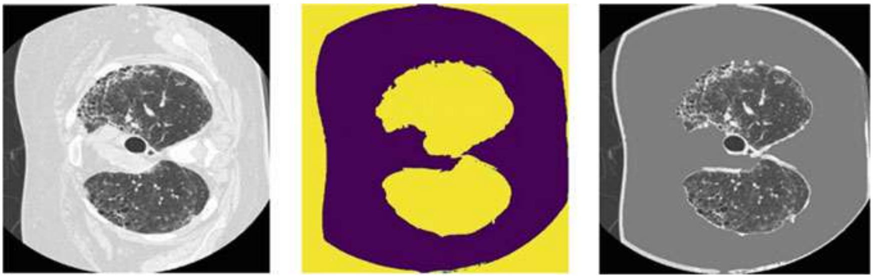


Fig. 4 **a** Image from data set. **b** Segmentation mask. **c** Segmented output

obtain a segmentation mask. Details of the segmented data set is shown in Fig. 4. Then, this segmentation mask was multiplied with the image to make the pixels of non-lung portion zero. This segmented image is taken and normalized such that every pixel lies between −1 and +1.

3.2 Model Architecture of Proposed Method1

Our model of proposed method1 uses pre-trained EfficientNetB0 (i.e., we have initialized the model with Imagenet weights) architecture as the backbone with few convolutional layers, MaxPooling, and dense layers on top of it. Model architecture is shown in Fig. 5. To understand the working of our proposed model, the architecture of EfficientNet is required to be explained.

Efficient Net: are a family of networks introduced by Google AI. These are highly scaled architecture yielding much better performance than usual ResNets. A com-

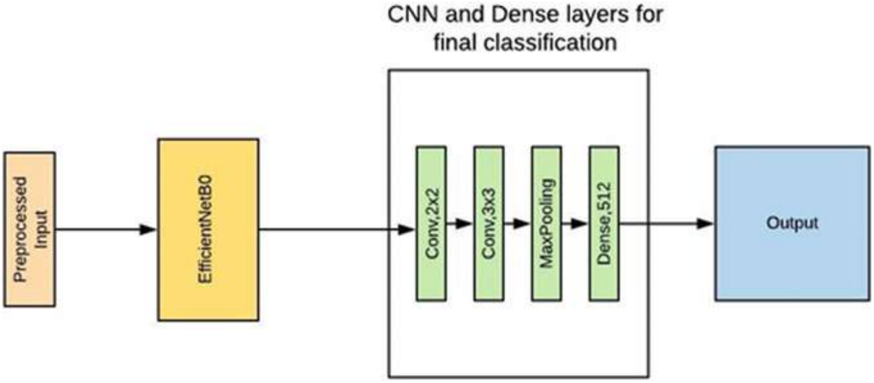


Fig. 5 Model architecture of method1

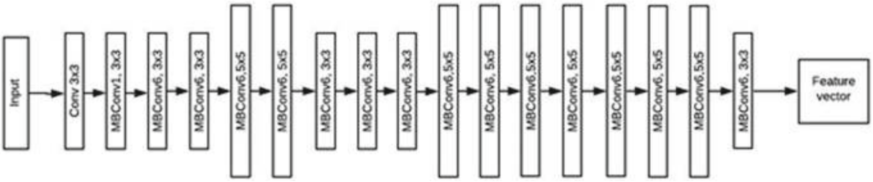


Fig. 6 Network arcthitecture of EfficientNetB0 [17]

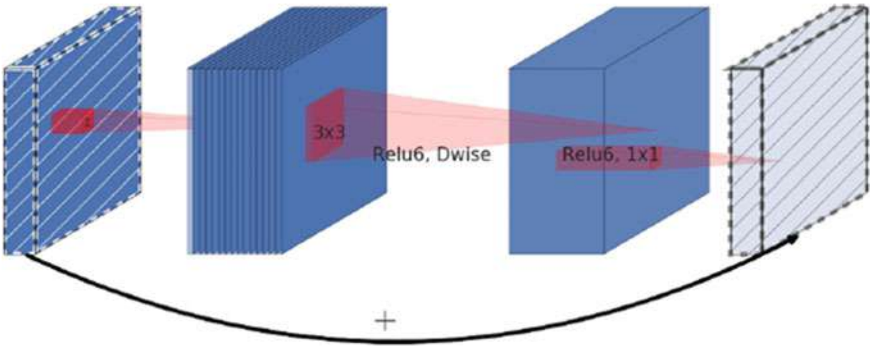


Fig. 7 Inverted residue blocks

pound scaling method is adapted to scale the width (no of filters), depth (no of layers), and resolution. A grid search is performed to find the relationship between different scaling dimensions of the baseline network under a fixed resource constraint. The detailed architecture of EfficientNet is shown in Fig. 6. EfficientNets use a special type of blocks called MBConv blocks which are nothing but residual inverted blocks used in MobilenetV2.

Inverted Residual block: These blocks are designed to deal with two problems: 1. Feature maps can be encoded in a low-dimensional subspace. 2. Nonlinear activation results in loss of information. These blocks take a low-dimensional tensor with c channels and apply point-wise (1 × 1) convolutions to expand or project the tensor on to higher-dimensional space which suits to nonlinear activation. This is followed by applying ReLU activation. This is shown in Fig. 7. A depth-wise convolution is performed using 3 × 3 kernels which are again followed by a ReLU nonlinearity. Now finally, this tensor is projected back to the low dimensionality using another point-wise (1 × 1) convolution. And now, a skip connection is given from the input to the output of the block.

Depth-wise Convolution: In depth-wise convolutions, the input image is convoluted without changing the depth(third dimension). So, for an input tensor having ‘c’ channels, depth-wise convolution is performed using c convolutional filters by obtaining scalar products of each channel with each filter.

Since the data set is too small, we have used EfficientNetB0 architecture which has around four million parameters. Our model as a whole has nine million parameters. We have used dropouts to prevent overfitting.

Loss Function used: Since there is a lot of class imbalance in the data set, we adopted focal loss to train our network. The focal loss mathematically forces the model to focus on the class which has fewer data points (which is COVID in our case).

$$L = -P_t * ((1 - P)^{\gamma}) * log(P) - (1 - P_t) * P^{\gamma} * log(1 - P_t)$$

where P_t is true probability
γ is hyperparameter
P is predicted probabaility.

3.3 Model Architecture of Proposed Method2:

Model of our proposed method2 uses ResNet which is explained below.

ResNet is designed to address the problem of vanishing gradient which is the most common problem faced by researchers while trying to train deeper networks. This is because when the networks are too deep, gradients form which loss is calculated can easily shrink to zero after several applications of the chain rule. This results in stagnant weights, thereby stopping the process of learning. ResNet solves this problem by introducing a skip connection which enables the flow of gradient information directly through skip connection to initial layers from later layer which thereby preventing the problem of vanishing gradients and facilitates a better learning process. The network architecture of ResNet is shown in Fig. 8.

For a baseline performance, we have trained a standard ResNet with binary cross-entropy loss and just normalizing the image such that all its pixel values lie between 0 and 1.

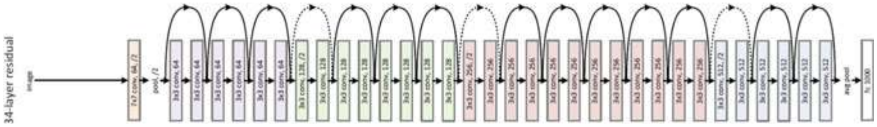


Fig. 8 Resnet architecture [1]

Table 1 Performance metrics of efficient net

Matric	Value
Accuracy	0.806
F1 Score	0.823
Recall	0.848
Precision	0.800
ROC AUC score	0.806

4 Results

This section contains the details of data sets and platform used in the experimentation followed by the detailed result analysis for the two proposed methods designed by us. We have used standard metrics like accuracy, recall, and F1 scores to compare our two models. The highly scaled efficientnet architecture along with segmentation model(for identifying region of interest) outperformed the basic ResNet classifier. The model is trained on around 1200 images and validated on 100 images. We tabulated the standard metrics of our best performing model in Table 1.

Dataset and Platform Used: The data we have considered is from different sources and have a bias in it. The model was implemented using TensorFlow framework and was trained on NVIDIA Tesla K80 (on Google Colab) of 12GB graphics RAM. The model was trained for 30 epochs which took around 40 min.

4.1 Results: Method1

After the training phase, we have validated the model on 100 images, and we have observed 80.6% accuracy. After 25 epochs, there is no significant decrease in training loss is observed and training accuracy went up to 98%. Model accuracy and loss are plotted and shown in Figs. 9 and 10, respectively. And our model classified all true positives correctly. We have observed a very less rate of false positives which shows the robustness of our classifier. We have observed a high recall of 0.848 which strongly suggests false negatives are very less. And having high recall is a good thing especially in models where cost associated with false negatives is high. The Performance metrics are shown in Table 1.

We used a technique called ‘GradCam’ for plotting and obtaining a visual heatmap which is useful to test the reliability of the model in the real world. Gradcam is a technique that enables researchers to debug their models visually. It uses gradient information flowing from every neuron into the last layer to compute each neuron’s interest. This eventually gives us a heatmap activating some of the regions in which

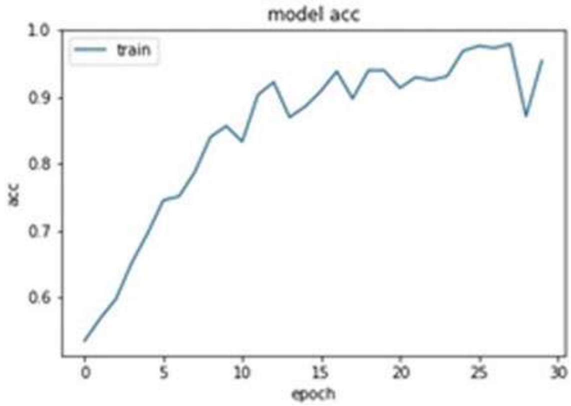


Fig. 9 Model accuracy

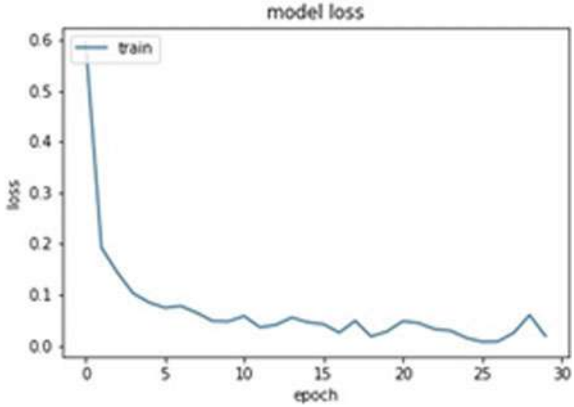


Fig. 10 Model loss

most of the neurons are interested in. The heatmaps of positive COVID CT images are shown in Fig. 11. The activated regions while predicting the label for an image are shown invariants of red and yellow depending on how strongly they agree with the final prediction.

4.2 Results: Method2

ResNet model was trained in Microsoft Azure with binary cross-entropy loss which took around one hour to complete the training procedure. After the training procedure, we have validated the model on 300 images, and an accuracy of 70% was observed. When compared to the ResNet model, EfficientNet model outperformed in terms of accuracy because of its highly scaled architecture. EfficientNet's lightweight architecture helped in preventing over-fitting up to some extent. As part of the work, we have developed a tool whose details are given below.

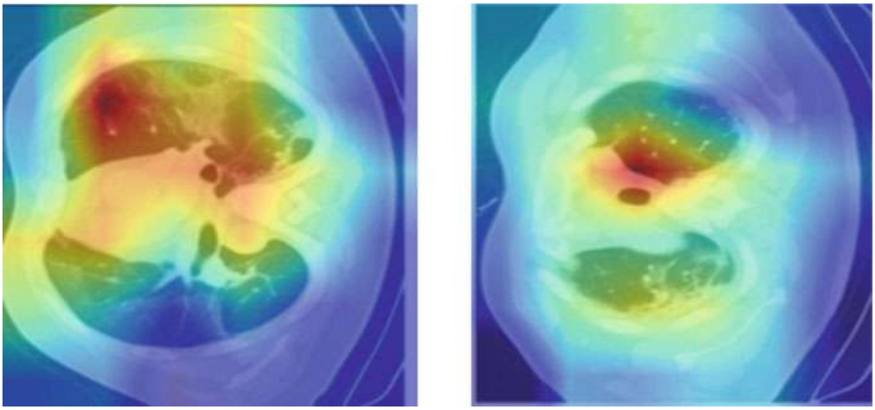


Fig. 11 Heatmaps of the positive COVID CT images

Details of the Tool Developed: This tool uses TensorFlow frame work, and the program was written in Python language. The source code is kept open and is available in the following link https://github.com/mano3-1/Covid-detection-with-CT-scans.

5 Conclusion and Future Work

5.1 Conclusion

COVID-19 had made a total disruption to human being and our economy till today. CT scans are a commonly used imaging modality to diagnose this disease. But the intra- and inter-observer variation of the doctors analyzing these CT images causes this decision making hard. So, it would be nice if a computational algorithm can perform this using recent AI techniques. We have proposed and developed two deep learning-based methods for the fast detection of COVID-19 from CT scans. From the results, it is evident that the EfficientNet-based method is better performing than the Resnet in terms of accuracy. Pre-processing like segmentation using BCD U-net made the training process a lot easier and forced the model to look for features in region of interest. Focal loss succeeded in forcing the model to learn both positive and negative data points equally despite of low positive data points in the data set. With all these techniques, we achieved an accuracy of 80.6% in EfficientNet architecture. Making an open-source initiative like this, we hope many hospitals can use this tool directly or customize it to suit their need.

5.2 Future Work

Firstly, we are planning to increase performance using self-supervised techniques. Since the data is of CT scan images, there is no significant effect of using pre-trained Imagenet weights. Techniques like denoising autoencoder can drastically improve the performance of the model. Secondly, we would like to use the Bayesian optimization technique to fine tune the parameters for better performance. We are also working on improving gradcam heatmaps.

Acknowledgements We would like to express our sincere thanks to the National Institute of Technology for its continuous support for doing this project. We also thank NVIDIA and CDAC Pune for giving us a chance to take part in their SAMHAR Hackathon 2020.

References

1. Zheng, C., Dengy, X., Fu, Q., Zhou, Q., et al.: Deep Learning-based Detection for COVID-19 from Chest CT using Weak Label medRxiv (2020)
2. Ozturka, T., Talob, M., Azra Yildirimc, E., Baloglud, U.B., et al.: Automated detection of COVID-19 cases using deep neural networks with X-ray images. Comput. Biol. Med. p. 103792 (2020)
3. Li, L., Qin, L., Xu, Z., Yin, Y. et al.: Artificial intelligence distinguishes covid-19 from community acquired pneumonia on chest CT. Radiology, p. 200905 (2020)
4. Xu, X., Jiang, X., Ma, C., Du, P., Li, X., Lv, S., et al.: Deep Learning system to screen coronavirus disease 2019 pneumonia. Appl. Intell. **22**, 1 (2020)
5. Amyar, A., Modzelewski R., Ruan, S.: Multi-Task Deep Learning Based Ct Imaging Analysis For Covid-19:Classification And SegmentationmedRxiv (2020)
6. Singh, D., Kumar, V., Kaur, M.: Classification of COVID-19 patients from chest CT images using multi-objective differential evolution-based convolutional neural networks. Eur. J. Clin. Microbiol. Infect. Diseases, pp. 1–11 (2020)
7. Wang, S., Zha, Y., Li, W., Wun Q., et al.: A fully automatic deep learning system for COVID-19 diagnostic and prognostic analysis. Eur. Res. J. (2020)
8. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention, pp. 234–241. Springer (2015)
9. Wang, S., Kang, B., Ma, J., Zeng, X., Xiao, M., Guo, J., Cai, M., Yang, Y., Li, Y., Meng, X., et al.: A deep learning algorithm using ct images to screen for corona virus disease (covid-19). medRxiv (2020)
10. Zhu, N., Zhang, D., Wang, W., et al.: A novel coronavirus from patients with pneumonia in China. New England J. Med. (2020)
11. Li, Q., Guan, X., Wu, P., et al.: Early transmission dynamics in Wuhan, China, of novel coronavirus-infected pneumonia. New England J. Med. (2020)
12. Hesamian, M.H., Jia, W., He, X., Kennedy, P.: Deep learning techniques for medical image segmentation: achievements and challenges. J. Digit. Imag. **32**(4), 582–596 (2019)
13. Song, F., Shi, N., Shan, F., et al.: Emerging coronavirus 2019-nCoV pneumonia. Radiology **295**(1), 210–217 (2020)
14. Xie, X., Zhong, Z., Zhao, W., Zheng, C., Wang, F., Liu, J.: Chest CT for typical 2019-nCoV pneumonia: relationship to negative RTPCR testing. Radiology, p. 200343 (2020)
15. WHO, World health organization (who) website. https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200213-sitrep-24-covid-19.pdf?sfvrsn=9a7406a4_4. Accessed 15 June 2020

16. Ardila, D., Kiraly, A.P., Bharadwaj, S., Choi, B., Reicher, J.J., Peng, L., Tse, D., Etemadi, M., Ye, W., Corrado, G., Naidich, D.P.: End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. Nat. Med. **25**(6), 954–61 (2019)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
18. Tan, M., Le, Q.V.: Efficientnet: Rethinking Model Scaling for Convolutional Neural Networks. arXiv preprint arXiv:1905.11946 (2019)

Towards Protein Tertiary Structure Prediction Using LSTM/BLSTM

Jisna Antony, Akhil Penikalapati, J. Vinod Kumar Reddy, P. N. Pournami, and P. B. Jayaraj

Abstract Determining the native structure of a protein, given its primary sequence is one of the most demanding tasks in computational biology. Traditional protein structure prediction methods are laborious and involve vast conformation search space. Contrarily, deep learning is a rapidly evolving field with outstanding performance at problems where there are complicated relationships between input features and desired outputs. Various deep neural network architectures such as recurrent neural networks, convolution neural networks, deep feed-forward neural networks are becoming popular for solving problems in protein science. This work mainly concentrates on prediction of three-dimensional structure of proteins from the given primary sequences using deep learning techniques. Long short-term memory (LSTM) and bidirectional LSTM (BLSTM) neural network architectures are used for predicting protein tertiary structures from primary sequences. The result shows that single-layer BLSTM networks fed with primary sequence and position-specific scoring matrix data gives better accuracy compared to LSTM and two-layer BLSTM models. This study may get benefited to the computational biologists working in the area of protein structure prediction.

Keywords Proteins · Primary structure · Tertiary structure · Deep learning · LSTM · BLSTM

J. Antony (✉) · A. Penikalapati · J. V. K. Reddy · P. N. Pournami · P. B. Jayaraj
Department of Computer Science and Engineering, National Institute of Technology Calicut, Kozhikode 673601, Kerala, India
e-mail: jisna_p170107cs@nitc.ac.in

A. Penikalapati
e-mail: akhilchowdary562@gmail.com

J. V. K. Reddy
e-mail: vinod.jammala@gmail.com

P. N. Pournami
e-mail: pournamipn@nitc.ac.in

P. B. Jayaraj
e-mail: jayarajpb@nitc.ac.in

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_6

1 Introduction

Proteins are biological polymers composed of amino acids, connected together by peptide bonds to form a polypeptide chain. Proteins are helpful in carrying out fundamental biological processes necessary for life. They help in creating and maintaining the shapes of cells and tissues, act as molecular factories, as a signal and receiver for cellular communications, as transporters and motors, and much more.

A protein's biological function is based on the arrangement of atoms of amino acids in its three-dimensional structure. Having a three-dimensional structure provides a greater level of understanding of how a protein functions and is crucial for computational drug design. The structure of a protein determines its function as well as the extent of its dysfunction in diseases.

Proteins have complex shapes involving various folds, loops, and curves. Chemical bonding between portions of the polypeptide chain helps in holding the protein together and giving it its shape. There are four types of protein structures namely primary, secondary, tertiary and quaternary structures.

Determining the structure of a protein given its sequence is still one of the greatest challenges in computational biology, since there are complex dependencies between protein sequence and its three-dimensional structure. Traditional methods for protein structure prediction involve complex stages and are time consuming. With the invention of GPUs providing high computation power, high performance computers are being largely used for solving compute intensive tasks.

In recent years, deep learning [1] techniques outperform traditional methods in solving problems involving complex input–output relationships. The increase in the amount of data and the computation power has led to greater use of deep learning solutions to many hard problem domains. Deep learning techniques have proved its capability in extracting the features from input data and utilizing it for solving problems at diverse fields, viz. computer vision, natural language processing and healthcare. Computational biologists can apply these techniques for solving biological problems. Since protein sequences show a greater similarity to natural languages, those techniques can be utilized in protein prediction problems. Majority of the deep learning techniques involve neural networks such as feed-forward neural networks, convolutional neural network (CNN) and recurrent neural network (RNN). Sequential models that utilize the input from previous outputs are mostly dealt with recurrent neural networks. These artificial intelligence (AI)-based deep learning methods have brought significant advancements in many areas including predicting protein folds based on the knowledge of their amino acid sequence.

LSTM networks are a type of recurrent neural networks that mainly focus on learning order dependencies in sequence prediction problems by extending the memory of RNN. They have a short-term memory to process the previous information to be used in the current cell. As the requirement here is predicting coordinates at every time step that corresponds to one amino acid, many-to-many architecture will be a great choice. In order to make use of future information, not limited to previous information while prediction, a many-to-many BLSTM [2] architecture with one forward layer and a backward layer will suffice our demands.

The paper focuses on long short-term memory(LSTM) and BLSTM models, which are recurrent neural networks, for predicting the protein tertiary structure. The model developed in this paper predicts the tertiary structure of a protein from its amino acid sequence using deep learning techniques.

The rest of the paper is organized as follows: Sect. 2 discusses the literature survey carried out for the work, Sect. 3 focuses on the methodologies adopted and Sect. 4 is about results and discussion followed by the conclusion in Sect. 5.

2 Related Works

This section describes some of the related works in predicting protein tertiary structure using deep learning techniques.

Bidirectional encoder representations from transformers (BERT) [3], a recently introduced transformer architecture, captures the folding structure of proteins even though they are not trained to do so. BERT is a masked language model, which is developed mainly for natural languages. The model has been successfully applied on protein sequences since the protein sequence can also be considered as a language with a vocabulary of 20 amino acids. Similar to the relationships between words in a sentence, there exists contacts between amino acids in a protein sequence (contact maps). The motivation behind the BERT analysis is that: a model capable of understanding the relationship among the words in a language can predict contact maps for a protein sequence. This paper mainly concentrates on: what BERT learns about the 3D structure of a protein, while training only with primary sequences.

By making use of the attention mechanism in BERT, one amino acid is capable of gathering information from all other amino acids in order to predict the contact maps and binding sites of a protein. Binding sites are those regions in proteins where a binding occurs with other molecules. It is also noted that a deeper layer on BERT learns more complex biophysical properties. BERT proves to be successful in computing the proportion of attention an attention head spends on each amino acid. BERT also compares the similarity matrices of different amino acids with standard substitution matrices. Apart from these, BERT identifies local substructures (secondary structures) from the given primary sequence.

AlphaFold [4], a deep neural network-based architecture, was trained to predict the distances between pairs of amino acids as well as their bond angles. Using the distance information and by using gradient descent optimization, AlphaFold searches for native protein structures. A second neural network compares the accuracy of the predicted values with actual ones.

The model mentioned in paper [5] predicts the tertiary structure of a protein from its primary sequence and position-specific scoring matrix (PSSM) values. PSSM gives the position-specific mutation scores of an amino acid. The model predicts the output without using co-evolutionary data and structural templates. The input sequence fed to the network is a vector of vectors of length 41. A recurrent geometric network (RGN) [5] that involves three steps, viz. computation, geometry, and assessment, has been developed and implemented in the paper. In the first stage, torsional angles are predicted. These angles along with a starting coordinate serves as the input to the geometric unit, which generates corresponding coordinates from these torsional angles. And, the third stage does the assessment of the predicted structure using the distance-based root-mean-square deviation (dRMSD) metric.

The model in paper [6] uses deep convolutional neural networks and learns complex features from a given input and output pair. More the number of deep CNN, more the learned relationships. The model also places two restricted Boltzmann machine (RBM) layers on the top of CNN to extract global features that improves structure prediction process. Their study shows that more complex features can be learned by deeper networks.

In recent years, critical assessment of structure prediction (CASP), a biennial event to determine the state of the art in protein structure prediction, witnessed a drastic improvement in the accuracy of predicted structures by utilizing residue–residue contact relationships. MULTICOM [7], which is one of the top ranked servers in CASP13 competition employs deep learning techniques for contact-distance prediction. They also employed similar techniques for ab-inito modelling as well as for quality assessment. MULTICOM [7] combines the co-evolutionary information of proteins with deep convolutional networks and residue networks and was able to enhance the contact prediction results. Success of the contact-map based methods depends on distance restraints. High-quality 3D models can be obtained if enough distance constraints are available. Noisy or sparse contact distances poses difficulties in 3D model prediction.

2.1 LSTM and BLSTM Networks

LSTM networks are recurrent neural network architectures suited for capturing long-term temporal dependencies in long length sequences. LSTM networks [8] are made using LSTM cells. Each LSTM cell has the capability to decide on how much of data to be permitted from the previous state, to what amount the current cell needs to be updated and how much of the data to be made visible to the next state. These decisions are made by forgot gate, input state, and output state in an LSTM cell. An LSTM cell is shown in Fig. 1.

Since LSTM networks process the data using previous context only, BLSTM networks are introduced to include previous as well as future context information, by processing the data from both directions with separate hidden layers. Hence, the output at current time step considers its previous and future context information.

A lot of papers on protein structure prediction using deep learning techniques have been published in recent years. Though deep learning is an emerging field in solving various complex problems, they are not fully successful yet in predicting an accurate protein structure. The proposed work finds the feasibility of neural network models like LSTM and BLSTM for predicting protein structures using its primary sequence.

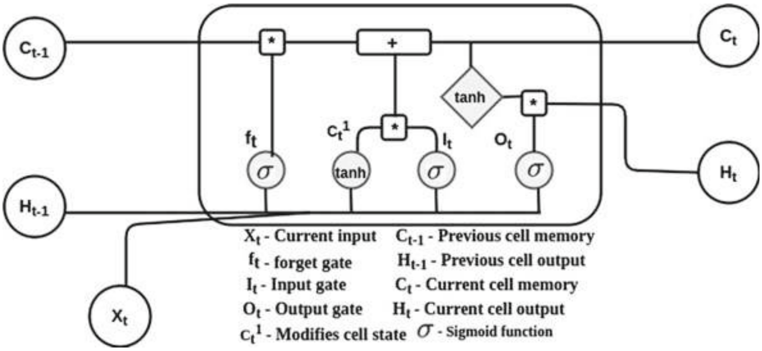


Fig. 1 An LSTM cell

3 Methodology

This section discusses the workflow and various models employed for this work. The workflow diagram is shown in Fig. 2.

3.1 Data Collection

ProteinNet [9, 10], a standard dataset for machine learning problems that involves protein structure prediction, is used as the dataset for the proposed work. The dataset contains primary sequences, secondary, and tertiary structures and position-specific scoring matrices along with the information content. This dataset includes the data splits for training, validation and testing. ProteinNet contains datasets from CASP 7 to CASP 12. CASP 7 dataset is taken for training the model. Around 10,650 samples, including 10,333 for training, 224 for validation, and 93 for testing have been taken for first-level training. The output format comprises 3D coordinates of backbone atoms, three for each of the amino acid residues.

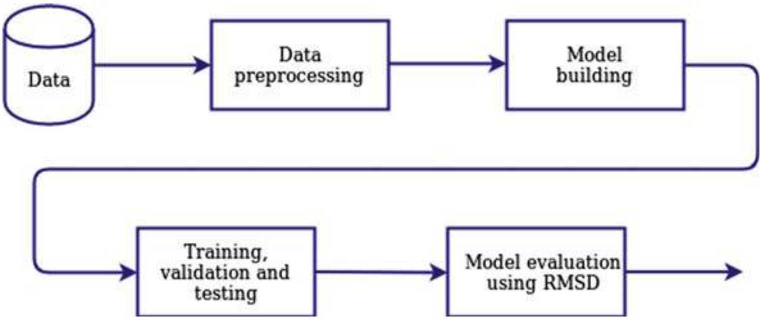


Fig. 2 Workflow of the proposed system

3.2 Data Pre-processing

For the sake of simplicity, only the backbone atoms of each amino acid in the sequence corresponding to any one chain of the protein is taken. Then the primary sequences and their respective 3D coordinates are brought together in one file, which forms the data sample. Abiding by the input specifications of LSTM architecture, one-hot encoding of sequence as well as vectoring of both the dependent and independent variables into 3D matrix forms has been done. All the pre-processing has been done through Python coding.

ProteinNet dataset contains the primary sequence, PSSM values along with the information content, secondary structures, tertiary structures and its mask. Among these primary sequences, PSSM values along with information content and tertiary structure coordinates are of main interest for this work. So, a parser has been created to generate individual files for each protein consisting of its primary sequence, PSSM values, information content, and the three coordinates of backbone atoms, for each amino acid in the sequence. And now, the data is in the required format that can be read into the input matrices easily.

3.3 Proposed Models

A detailed analysis of protein data has suggested us to use LSTM models. For improving the results, training was also extended with BLSTM and a two-layered BLSTM model. More details on these models are discussed below and Fig. 3 shows various neural network architectures employed for this experimentation.

3.3.1 Method 1: Using LSTM Network

An LSTM model with one LSTM layer and a dense layer (activation layer) has been implemented initially. The number of time steps (LSTM units) is set to 500, as the maximum length of sequence is capped to 500 amino acids. At each time step, the input is a 21 dimensional one-hot vector representation of amino acid and the 3D coordinates of that amino acid forms the corresponding output. The α-carbon atom is considered as a representative of each amino acid. The model was trained with approximately 10,000 samples and tested with around 2000 samples. The activation function used in the LSTM layer is the 'tanh' function and the activation function used the in dense layer is a 'linear' function. Root-mean-square deviation (RMSD) is taken as the evaluation metric.

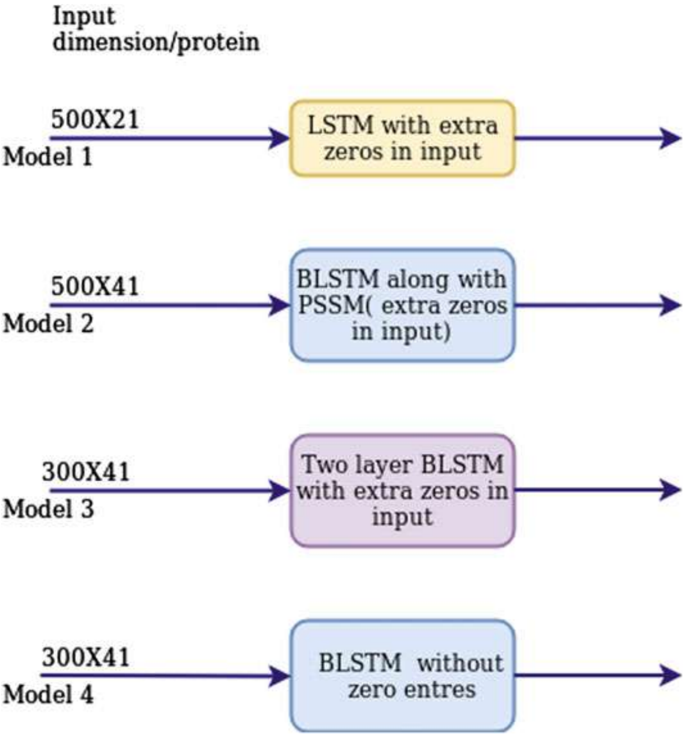
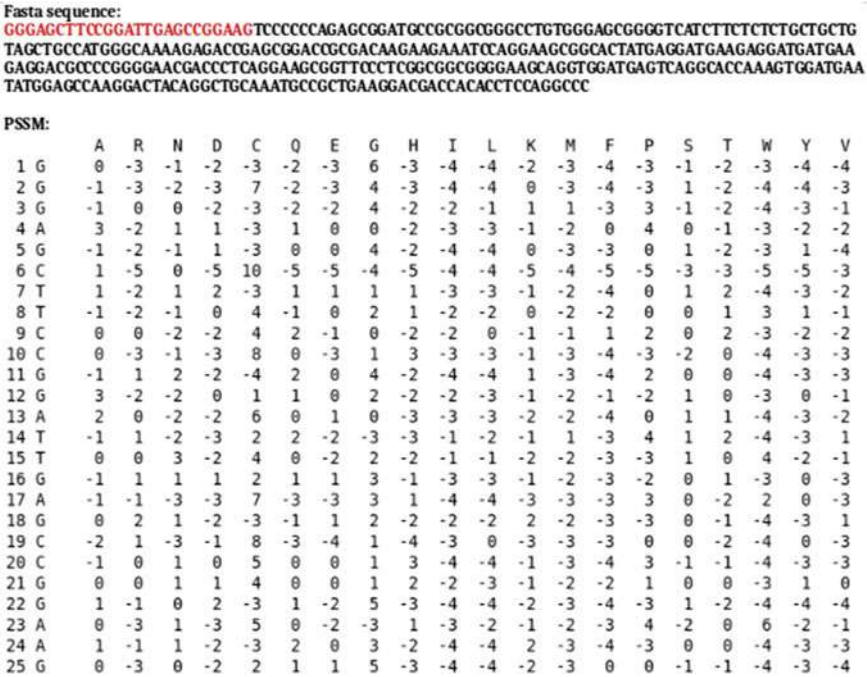

Input dimension/protein
500X21
Model 1
LSTM with extra zeros in input
500X41
Model 2
BLSTM along with PSSM(extra zeros in input)
300X41
Model 3
Two layer BLSTM with extra zeros in input
300X41
Model 4
BLSTM without zero entres

Fig. 3 Proposed models

3.3.2 Method 2: Using BLSTM Network Along with PSSM Data

In the first model based on LSTM, only the amino acid sequence is passed as input for each LSTM unit. For effective training, it has been decided to use PSSM scores as input, along with the one-hot vector of amino acids. A PSSM is a type of scoring matrix which is passed along with amino acid. Each entry in this matrix contains a score which indicates the log-likelihood of substitution of an amino acid for a particular position in the sequence. As there are 20 well known amino acids, there will be 20 values for each entry in the matrix. PSSM values are generally not normalized ones, with a positive value indicating a favoured substitution and a negative value for a less frequent substitution. These values are usually normalized before feeding to a model. For each amino acid, three backbone atoms are taken for training. So, at each time step, there are nine output coordinates, three for each amino acid in backbone structure. A sample PSSM matrix is shown in Fig. 4.

A BLSTM model as shown in Fig. 5 has been implemented and trained with approximately 10,000 input samples and 200 validation samples, collected from CASP 7 dataset. The input shape is (500, 41), since the maximum length of the sequence under consideration is 500 and 41 is its dimension. For one-hot encoding



Fasta sequence:

GGGAGCTTCCGGATTGAGCCGGAAGTCCCCCCAGAGCGGATGCCGCGGCGGGCCTGTGGGAGCGGGGTCATCTTCTCTCTGCTGCTG
TAGCTGCCATGGGCAAAAGAGACCGAGCGGACCGCGACAAGAAGAAATCCAGGAAGCGGCACTATGAGGATGAAGAGGATGATGAA
GAGGACGCCCCGGGGAACGACCCTCAGGAAGCGGTTCCCTCGGCGGCGGGGAAGCAGGTGGATGAGTCAGGCACCAAAGTGGATGAA
TATGGAGCCAAGGACTACAGGCTGCAAATGCCGCTGAAGGACGACCACACCTCCAGGCCC

PSSM:

		A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
1	G	0	-3	-1	-2	-3	-2	-3	6	-3	-4	-4	-2	-3	-4	-3	-1	-2	-3	-4	-4
2	G	-1	-3	-2	-3	7	-2	-3	4	-3	-4	-4	0	-3	-4	-3	1	-2	-4	-4	-3
3	G	-1	0	0	-2	-3	-2	-2	4	-2	-2	-1	1	1	-3	3	-1	-2	-4	-3	-1
4	A	3	-2	1	1	-3	1	0	0	-2	-3	-3	-1	-2	0	4	0	-1	-3	-2	-2
5	G	-1	-2	-1	1	-3	0	0	4	-2	-4	-4	0	-3	-3	0	1	-2	-3	1	-4
6	C	1	-5	0	-5	10	-5	-5	-4	-5	-4	-4	-5	-4	-5	-5	-3	-3	-5	-5	-3
7	T	1	-2	1	2	-3	1	1	1	1	-3	-3	-1	-2	-4	0	1	2	-4	-3	-2
8	T	-1	-2	-1	0	4	-1	0	2	1	-2	-2	0	-2	-2	0	0	1	3	1	-1
9	C	0	0	-2	-2	4	2	-1	0	-2	-2	0	-1	-1	1	2	0	2	-3	-2	-2
10	C	0	-3	-1	-3	8	0	-3	1	3	-3	-3	-1	-3	-4	-3	-2	0	-4	-3	-3
11	G	-1	1	2	-2	-4	2	0	4	-2	-4	-4	1	-3	-4	2	0	0	-4	-3	-3
12	G	3	-2	-2	0	1	1	0	2	-2	-2	-3	-1	-2	-1	-2	1	0	-3	0	-1
13	A	2	0	-2	-2	6	0	1	0	-3	-3	-3	-2	-2	-4	0	1	1	-4	-3	-2
14	T	-1	1	-2	-3	2	2	-2	-3	-3	-1	-2	-1	1	-3	4	1	2	-4	-3	1
15	T	0	0	3	-2	4	0	-2	2	-2	-1	-1	-2	-2	-3	-3	1	0	4	-2	-1
16	G	-1	1	1	1	2	1	1	3	-1	-3	-3	-1	-2	-3	-2	0	1	-3	0	-3
17	A	-1	-1	-3	-3	7	-3	-3	3	1	-4	-4	-3	-3	-3	3	0	-2	2	0	-3
18	G	0	2	1	-2	-3	-1	1	2	-2	-2	-2	2	-2	-3	-3	0	-1	-4	-3	1
19	C	-2	1	-3	-1	8	-3	-4	1	-4	-3	0	-3	-3	-3	0	0	-2	-4	0	-3
20	C	-1	0	1	0	5	0	0	1	3	-4	-4	-1	-3	-4	3	-1	-1	-4	-3	-3
21	G	0	0	1	1	4	0	0	1	2	-2	-3	-1	-2	-2	1	0	0	-3	1	0
22	G	1	-1	0	2	-3	1	-2	5	-3	-4	-4	-2	-3	-4	-3	1	-2	-4	-4	-4
23	A	0	-3	1	-3	5	0	-2	-3	1	-3	-2	-1	-2	-3	4	-2	0	6	-2	-1
24	A	1	-1	1	-2	-3	2	0	3	-2	-4	-4	2	-3	-4	-3	0	0	-4	-3	-3
25	G	0	-3	0	-2	2	1	1	5	-3	-4	-4	-2	-3	0	0	-1	-1	-4	-3	-4

Fig. 4 Position-specific scoring matrix (PSSM) for a segment (highlighted) of a fasta sequence

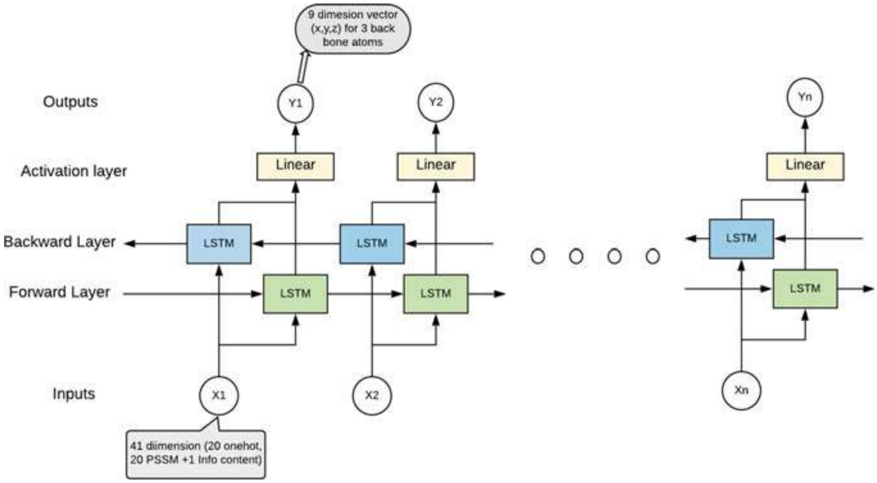


Fig. 5 Bidirectional LSTM architecture

of amino acids, a dimension of 20 is used, another 20 for PSSM score, and the information content takes one dimension. The activation function used is 'tanh'. Both the forward and backward LSTM layers of BLSTM are merged by *concat* method, which means outputs of forward and backward layers are concatenated.

3.3.3 Method 3: Using Two-layer BLSTM

In the third method, a hybrid model is built with two BLSTM layers and a dense layer. The input shape is reduced to (300, 41), i.e considering protein sequences with length less than or equal to 300 amino acids and with the dimension same as that of previous training input. The reason is that most of the protein sequences under consideration have length less than 300 and only a few proteins contain more sequence length and can be considered as outliers. Thus, reducing to length 300 will save the samples from padding more zeros. The model is trained with 8263 samples and validated with around 200 validation samples. The activation function used in the BLSTM layer is 'tanh' function. Both the forward and backward LSTM layers of BLSTM are merged by *concat* method.

3.3.4 Method 4: Using BLSTM by Replacing Zero Values in Data:

Here also, the maximum protein length is fixed to 300. For proteins whose length is less than 300, they are padded with zeros in order to make it a 300 length sequence. This creates a lot of zeros in training data and hence the weights learned by model may not be accurate in predicting new structures. In order to deal with this problem, the zero values have been replaced with nonzero neighbourhood values. With the belief that the drawback of the above models is in dealing with zeros, these zero values are replaced with nonzero neighbourhood values.

Table 1 shows the summary of all the models.

Table 1 Data description of proposed models

Model	Training samples	Validation samples	Test samples	input shape
LSTM	10,000	0	1757	(500, 21)
BLSTM	9815	213	91	(500, 41)
Two-layer BLSTM	8263	173	75	(300, 41)

3.4 Implementation Details

The model was implemented in Python and was trained and tested on NVIDIA Tesla K80 GPU (on Google Colab). The size of batches was set to 32. For better convergence of the model, Adam optimizer is used. RMSD is used as the final evaluation metric.

4 Results and Discussion

This section presents the results obtained and the rationale behind the selection of various models.

4.1 Method 1: LSTM

The RMSD of training data has been reduced from 25.26 to 22.14. For testing, 2000 data samples were chosen. The RMSD of the test data is 22.31. The input for this model is only an amino acid sequence, which does not include enough position-specific information for the model to learn. Also, the lack of long-term dependencies is another issue, and hence, a second type of model is chosen.

4.2 Method 2: BLSTM Along with PSSMs

The RMSD for this model has been reduced from 25.3 to 23.13 in 10 epochs for training data. For testing the model, about 100 samples were selected and the RMSD obtained is 17.13. The loss convergence graph can be seen in Fig. 5. The gap between the training and the validation losses persists even after experimenting with various over-fitting techniques. Model is able to predict coordinates for three backbone atoms in each amino acid. Even though the X coordinates of predicted protein are much similar to the actual coordinates, Y and Z coordinates are not learned properly. So a third modification is given to this model.

4.3 Method 3: Two-layer BLSTM

The RMSD for this model has been reduced from 27.76 to 25.19 in 15 epochs. The RMSD for the test data is 19.84. The loss convergence graph can be seen in Fig. 6. Here also, the predicted X coordinates are close to actual X coordinates, but predicted Y and Z coordinates are distant from actual coordinates. Also, the coordinates of the predicted amino acids are closer in space (Fig. 7).

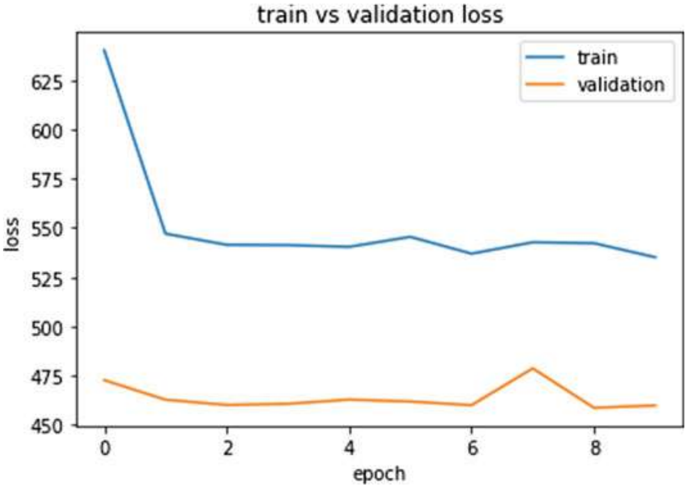


Fig. 6 Loss convergence over epochs for BLSTM

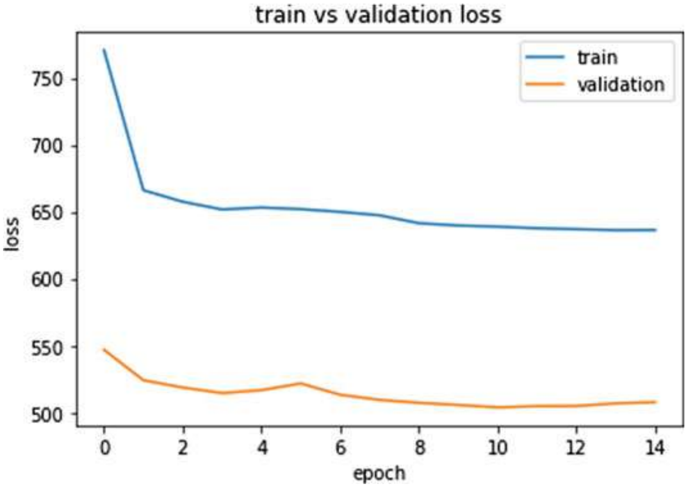


Fig. 7 Loss convergence over epochs for two-layer BLSTM

4.4 Method 4: Two-layer BLSTM

The 3D coordinate values in the predicted result fall to a particular range, and hence, this method does not give useful results. The loss obtained for this method is much higher than expected. Here, the validation curve lies above the training curve and hence is not shown in the plots.

RMSD is used for evaluating the predicted models. RMSD between the predicted coordinates and the actual coordinates is calculated. The RMSD for the Method 4

Table 2 Performance evaluation of the proposed models

Model	RMSD
LSTM	22.31
BLSTM	17.13
Two-layer BLSTM	19.84

is very large compared to the other methods and hence is not shown in the table. Here, the replaced nonzero values in the input sequence sounds dominating to other values, and thereby leading to a situation where the weights are over learned.

Table 2 shows root-mean-squared error of all the models except the last method. The lower the values of RMSD, better the results obtained. From the table, it can be seen that the BLSTM model gives better results compared to LSTM networks. The better results for BLSTM networks can be justified with their ability to learn long-term dependencies from both the directions. It can use the activation from future and past time steps in order to predict values at the current time step. The two-layer BLSTM shows high root-mean-square error compared to BLSTM since it is trained with fewer samples only.

Since PSSMs would not give much evolutionary information, the use of multiple sequence alignment (MSA) data along with the input sequence may improve the results. The proposed work comes under an end-to-end learning model where a single deep neural network represents the entire system. The RGN [5] method mentioned in the literature is an end-to-end differential model and is trained with CASP 7–12 datasets which comprises of a total size of 48GB of data. The proposed model is only trained with CASP 7 dataset (3.2 GB). Hence, it is not possible to perform a fair comparison due to this difference in the dataset. This study is a preliminary work to analyse the suitability of deep neural networks for protein structure prediction problems and to identify which among those models perform better.

5 Conclusion and Future Work

Artificial intelligence has got wide scope in many applications. The deep learning solutions are outperforming the state-of-the-art methods in various fields of research. From the present work, it is evident that recurrent neural networks like LSTM and BLSTM were able to predict the three-dimensional coordinates of protein from its primary sequences. The best root-mean-square error obtained by experimenting with various neural network models is 17.13. The RMSD value can even be reduced by adding more features to the model. So, the work concludes that deep learning models are good in dealing with problems where there are complex relationships between input features and desired outputs, and they have the ability to take advantage of increasing computational power and the amount of data.

As the above experimented models are quite reluctant in dealing with zeros, creating a dynamic time step model would give better results. Instead of directly predicting coordinates for amino acids, a model that predicts properties like torsion angles or contact distances, that is quite powerful to construct the 3D coordinates will be fine. Language models like transformer architecture which uses attention mechanisms will be good enough in capturing the relationship among the amino acids and hence predicting its three-dimensional structures. Since a protein sequence can be considered as a language constructed using the vocabulary of amino acids, and the contacts between amino acid residues show similarities to the relationships between words in a language, natural language processing models have a greater scope of applicability in this domain.

Acknowledgements The authors would like to thank Central Computing Centre, National Institute of Technology Calicut (NITC) for providing GPU servers for the execution of this work.

Funding This research did not receive any grants form any of the funding agencies.

Competing Interests The authors declare that they have no competing interests.

References

1. Paliwal, K., Lyons, J., Heffernan, R.: A short review of deep learning neural networks in protein structure prediction problems. In: Advanced Techniques in Biology and Medicine, vol. 24, pp. 1–2 (2015)
2. Ray, A., Rajeswar, S., Chaudhury, S.: Text recognition using deep BLSTM networks. 2015 Eighth International Conference on Advances in Pattern Recognition (ICAPR), Kolkata, pp. 1–6 (2015). https://doi.org/10.1109/ICAPR.2015.7050699
3. Vig, J., Madani, A., Varshney, L.R., Xiong, C., Socher, R., Rajani, N.F.: BERTology Meets Biology: Interpreting Attention in Protein Language Models. arXiv:2006.15222v2 (2020)
4. Senior, A.W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., et al.: Improved protein structure prediction using potentials from deep learning. Nature **577**(7792), 706–710 (2020)
5. Mohammed AlQuraishi, I.: End-to-end differentiable learning of protein structure. Cell Syst. (2019). https://doi.org/10.1016/j.cels.2019.03.006
6. Bai, L., Yang, L.: A Unified Deep Learning Model for Protein Structure Prediction. https://doi.org/10.1109/CYBConf.2017.7985752 (2017)
7. Hou, J., Wu, T., Cao, R., Cheng, J.: Protein tertiary structure modeling driven by deep learning and contact distance prediction in CASP13 (2019). https://doi.org/10.1002/prot.25697
8. https://medium.com/deep-math-machine-learning-ai/ . Accessed on 9 Oct 2019
9. AlQuraishi M., ProteinNet: a standardized data set for machine learning of protein structure. BMC Bioinformatics **20**, 311 (2019). https://doi.org/10.1186/s12859-019-2932-0
10. https://github.com/aqlaboratory/proteinnet . Accessed on 9 Dec 2019
11. http://www.rcsb.org/ . Accessed on 9 Oct 2019
12. Fiser A.: Template-based protein structure modeling. In: Computational Biology, pp. 73–94, Humana Press, Totowa, NJ (2010)
13. Rohl, C.A., Strauss, C.E., Misura, K.M., Baker, D.: Protein structure prediction using Rosetta. In: Methods in Enzymology, vol. 383, pp. 66–93, Academic Press (2004)
14. Antony, J., Sreenivas, V., Jayaraj, P.B.: Towards building a coordinate clustered library for template-based modeling of protein structures In: IEEE Recent Advances in Intelligent Computational Systems (RAICS). Thiruvananthapuram, India, pp. 219–223 (2018). https://doi.org/10.1109/RAICS.2018.8635068

An Android-Based Smart Home Automation System in Native Language

Nayan Thara Prakash, Mathew Santhosh, M. P. Sneha Raj, G. Gokul, and Gemini George

Abstract Speech recognition is the ability of a machine to analyze and respond to the oration of a person. The proposed system mainly concentrates on the people who are visually disabled, paralyzed and handicapped, so that such users can monitor home appliances from anywhere inside a home. The project develops a home automation system which is connected with an Android smartphone by an Arduino device. In this advanced world, people want to switch from the conventional switches to centralized control system. Especially, when it comes to the elderly or handicapped people as mentioned earlier, they may feel trouble in managing all switches which are located at different parts of their residence. So, the system assists them in handling everything within their smartphones. Here the other strong point of the work is that when it comes to the people can use their native language to interact with the device, and hence, can cater to a larger unprivileged sections of the society.

1 Introduction

Wireless home automation system is a consolidated, convenient home automation system that can be entirely controlled on a speech request. The system aids one to supervise his or her household gadgets from a centralized management unit. It is a setup that can be unified as a unique mobile device to manage all the home device.

Automation simply means the computerization or taking automatic control on certain electrical and electronic systems in a building. These include lighting, temperature control, etc. There was a significant advancement in the field of consumer electronics in the past decade. For the people who might otherwise require caretakers, the home automation system can provide increased quality of life.

People are living an ever-growing high quality of their lives today, and this leads to the consumption of more and more facilities and home appliances include into

N. T. Prakash (✉) · M. Santhosh · M. P. Sneha Raj · G. Gokul · G. George
Department of Computer Science and Engineering, St. Joseph's College of Engineering and Technology, Palai, Kerala, India
e-mail: nayantharaprakash2020@cs.sjcetpalai.ac.in

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_7

their buildings. Generally, conventional wall switches are located in different corners of a house, and this becomes very difficult for the elderly or physically handicapped people to operate them.

It is of comfort and helpful for such people to control every appliances within a touch or a command. This work aims to find a solution toward achieving this. Sometimes, people may leave their home without switching off their appliances due to their busy lifestyles. This carelessness not only results in the wastage of electricity but also brings potential danger to their home. Earlier, people were using simple security systems like CCTV monitoring. The condition and safety of the house are unknown when the user is away from the house for certain days. Now, when everything got automated, even house doors were unlocked by the owner's voice command. A minor change in the voice pattern of the commander may result in an alert message from the alarm system.

As per recent surveys, the smart home market is contemplated to be valued at USD 1514 billion by the year 2024. The growing installations of smart speakers, home healthcare products, smart furniture and many more have directed the growth of the smart home market.

2 Related Works

Asadullah [1] gives an overview about the current and emerging home automation systems. The work presents different types of wireless communication systems that are also mentioned. Ransing [2] explained about a system which focusses mainly on elderly care. Gnanavel [3] also presents a similar work. Shinde [4] proposed a system which works with Bluetooth and GSM, but the system faces problems like limited communication facility and also it has slow processing speed. Mtshali [5, 6] developed smart home appliance control systems for physically disabled people. Stefanov [7] developed a system in which both the elderly people and physically disabled ones can interact.

Elshafee [8] designed and implemented a Wi-Fi-based home automation system. The system consists of two elements: The first part acts as a server, which manages and controls the system, and the second one is hardware interface module. Tulshan [9] made a survey on virtual assistant: Google Assistant, Siri, Cortana, Alexa [10]. Amazon Alexa uses the artificial Intelligence technology developed by the Amazon. The assistant provides information like weather, traffic, sports also other real-time information like news, etc. The system is slow in updating new version of the software. Siri [11] is a smart assistance which does a faster, easier way to do things in an apple device. Siri supports user commands like performing phone actions checking basic information, navigating areas, etc. But Siri does not work in languages other than English. Google Assistant [12] is a virtual personal assistant developed by Google. It recognizes languages other than English and now works in both Android and iPhones. It consumes high power, and also it slows down the processing speed of users phone.

Microsoft developed Cortana to perform some tasks like setting remainders and giving answers to the user's questions. But hackers used the Cortana commands to break into a Windows10 PC. Behera [8] presented a system which does automation using SMS. But the system can be operated only in short distance. Livingston [11] also developed a digital assistant which helps the physically disabled people by voice command. Kodali [13] introduced a low-cost home automation system, and in the system a smartphone is used to control and monitor the home appliances using Node-RED, Mongoose OS, Android app and Google Assistant.

3 Implementation Details

3.1 Node MCU ESP32

The reason to choose the node MCU (Fig. 1) module is because of its low cost and low power consumption feature. Madhu [14] introduced the implementation of cost-effective smart home controller with Android application using node MCU and Internet of things (IOT) [15]. The main advantage of this module is that it is integrated with Wi-Fi and dual-mode Bluetooth. Programming of node MCU is done with Arduino IDE. The node MCU board has two on-board buttons known as BOOT and ENABLE. BOOT button boots the program, whereas the ENABLE button reboots the program. Bhandari [16] also designed a system with node MCU and IoT [17].

3.2 Relay Module

Relay is an electromechanical switch. The relay has a spring-like structure inside it, or more precisely we can say that there is a coil which acts as a magnet temporarily.

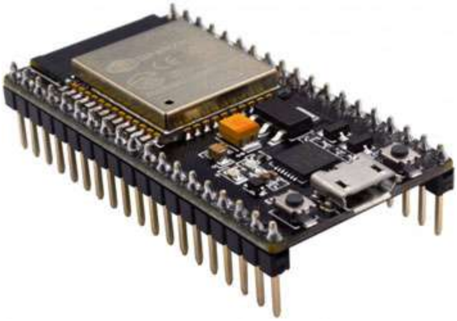


Fig. 1 Node MCU ESP32 module

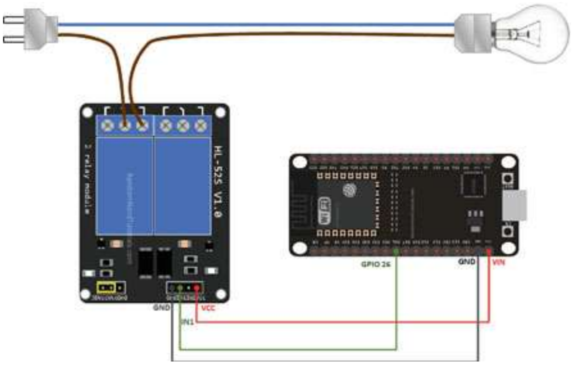


Fig. 2 Connection of the relay module with node MCU

Relay can act as a CLOSED as well as an OPEN switch. When coming to the relay operations, there are two circuits within a relay: One is the control circuit and the other one is the switched circuit.

The switched circuit is open when the relay acts as an OPEN switch; therefore, when a voltage is applied to the control circuit, the electromagnetic coil in the relay activates and pulls the switch closed. This enables the current to flow through the switched circuit.

The opposite will happen when the switch is CLOSED, when a voltage is applied to the control circuit, the electromagnetic coil in the relay activates and pulls the switch to open; therefore, the current flow will be forcefully stopped. The connections of relay module with the node MCU are shown in Fig. 2.

3.3 Wire Pieces

The wire pieces are used to connect the LED bulb with the node MCU board so that the wire gives a connection from the 27th, 26th and the two ground ports in a node MCU board.

3.4 LED Bulb

The positive terminal of the bulb is connected with 26th/27th ports of the node MCU board, and the bulb turns “ON” on the command “bulb on cheyyuka”. The negative terminal of the bulb is connected with the ground ports of the board, and the bulb turns “OFF” on the Arduino IDE command “bulb anaykkuka”. It is important to note that the native language used in our work is Malayalam, which is a regional language used in the state of Kerala, India.

3.5 Arduino IDE

The electronics programming division is done by the Arduino IDE. David [18] and Chandramohan [19] mentioned the design of a home automation system using Arduino. Arduino actually supports the Atmel chip. For running the program in this platform, additional supports have to be given, here the ESP32 module is used; therefore, the link of ESP32 module has to be copied in the preferences. Later the corresponding software of ESP32 will be installed to the device. So that we can use specific libraries of ESP32 like "wifi.h" inside the program. The ESP32 module has two core processors, and we are using the Wi-Fi antenna. The processor can be used as a host or a server. Malav [20] have done a research on home automation using Arduino. Here, in the proposed system, the speech is recognized by the Google speech to text, afterward a GET request is sent to the http header of the ESP32 module according to the speech command, afterward appropriate action is performed with respect to the command. The user's ssid and password are written in the program so that the system must be connected with the same user's Wi-Fi. After uploading the program, there will be an IP address on the output which is the IP of the node MCU. The IP is then copied to the program which is written in the Android studio.

3.6 Android Studio

Android studio is Android's official integrated development environment. Android studio is the platform which is used to develop the mobile application. The typical block diagram of the system is shown in Fig. 3.

Agarwal [21] presented a review on home automation using IoT. Also, Kodali [22] also proposed a project with IoT which implemented a security sytem, and Nagendra Reddy [23] also discussed on IoT-based home automation.

We are using four commands: They are "light on aakkuka, light anaykkuka, Fan on aakkuka, Fan off aakkuka". Ramlee [24] stated about a home automation system using Android. Javale [25] developed an automation system using Android accessory development kit. The speech is converted into the native language, and the converted text will be displayed on the screen. Any commands other than the programmed ones will give a toast like "wrong command". Gurek [26] proposed an Android-based system which has three hardware components, a local device to transfer signals to home appliances, a Web server to store customer records and support services to the other components and a mobile smart device running Android application.

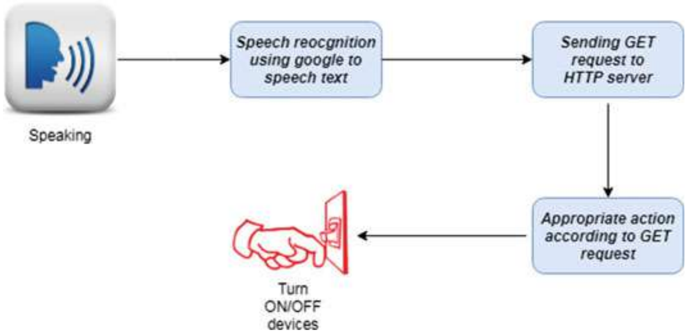

Speaking
Speech reocgnition using google to speech text
Sending GET request to HTTP server
Appropriate action according to GET request
Turn ON/OFF devices

Fig. 3 Block diagram

4 Applications

- Assists elderly and handicapped people—Our system is mainly designed for these elderly people who cannot move from their bed without a foreign help of other people. In the case of these elderly people, they can use our application through their smartphone simply by lying on the bed. This application is also supportive for the people who are physically disabled. Thus, the proposed system can reduce the user's physical strength that would be taken to move from their position to the switch board.
- Supervises all the home devices from one place—The devices which have been connected can be supervised and be controlled from any place of the home. All the devices can be controlled by the smartphone, we can switch ON/OFF the device from anywhere, and also we can know whether the device is switched ON/OFF.
- Adaptability toward new device—Our system has a capability to add more devices whenever we want to . And also deletion of any device is also updated in the system.
- Upgraded performance of appliances—Since we can know the status of the devices in our phone, if the device is ON for no use, then we can switch OFF by our phone, sometimes we may be lazy to go there and switch OFF that device, we can avoid this kind of unwanted roaming inside the house, thus we can save energy as well by properly turning OFF these devices.
- Works with native language—This is the main feature of our system that would make our system different from other home automation devices that the device can work in our native language, that is, the Malayalam language, which is a regional language used in the state of Kerala, India. This would help the illiterate and under privileged people who are not comfortable to interact with the English language. The proposed system has included the recognition of different slangs in the Malayalam language that would be useful to turn ON and turn OFF the light and the fan.

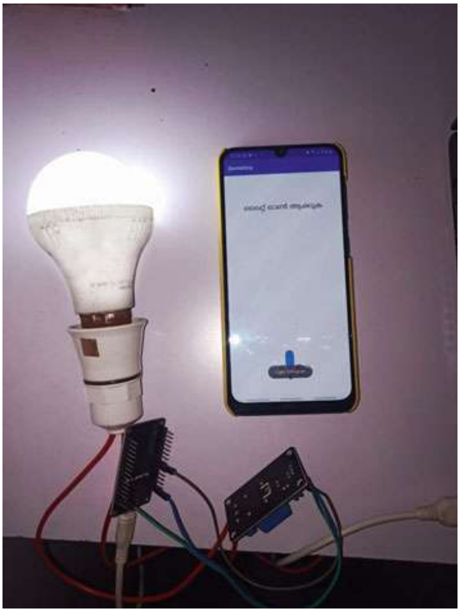


Fig. 4 Working model

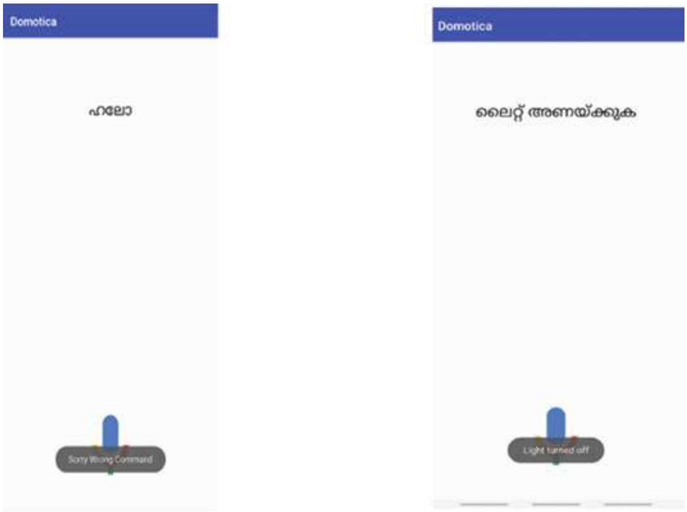
5 Results and Discussions

The speech will be recognized when the user starts to speak after turning ON the record button of the application (Fig. 4).

As mentioned earlier, the commands we have implemented according to our native language are “light on aakkuka, light anaykkuka, Fan on aakkuka, Fan off aakkuka”. When one of these commands are recognized, the toast will be shown corresponding to the command. If a wrong command is given other than the four commands, the toast will be shown as “Sorry Wrong Command”. The bulb turns ON for the command “light on cheyyuka” (in the native language). The views are shown in Fig. 5.

6 Conclusion

The major intention of this work is to support the handicapped or elderly people by providing them with a control system for managing different devices in home environment. The paper provides the implementation details of the proposed home automation system that supports Android smartphone by recognizing native language. Android studio and Arduino platform are used for the comprehensive software design of the system. The modules implemented are of low cost and thus make the system affordable to common people.



ഹലോ

Sorry Wrong Command

ലൈറ്റ് അണയ്ക്കുക

Light turned off

Fig. 5 Application interface view of wrong and correct commands

At present, the automation is limited with controlling of light and fan. However, the system can be enhanced by including other appliances such as refrigerator, air conditioners, washing machines, so that almost all the devices that have to be managed in a home scenario can be controlled at one's fingertips.

References

1. Asadullah, M., Raza, A.: An overview of home automation systems. In: 2016 2nd International Conference on Robotics and Artificial Intelligence (ICRAI), Rawalpindi, pp. 27–31 (2016). https://doi.org/10.1109/ICRAI.2016.7791223
2. Ransing, R.S., Rajput, M.: Smart home for elderly care, based ON wireless sensor network. In: 2015 International Conference on Nascent Technologies in the Engineering Field (ICNTE), Navi Mumbai, pp. 1–5 (2015). https://doi.org/10.1109/ICNTE.2015.7029932
3. Gnanavel, R., Anjana, P., Nappinnai, K.S., Sahari, N.P.: Smart home system using a Wireless Sensor Network for elderly care. In: 2016 Second International Conference on Science Technology Engineering and Management (ICONSTEM), Chennai, pp. 51-55 (2016). https://doi.org/10.1109/ICONSTEM.2016.7560922
4. Shinde, A., Kanade, S., Jugale, N., Gurav, A., Vatti, R.A., Patwardhan, M.M.: Smart Home automation system using IR, bluetooth, GSM and android. In: 2017 Fourth International Conference on Image Information Processing (ICIIP), Shimla, pp. 1–6 (2017). https://doi.org/10.1109/ICIIP.2017.8313770
5. Raju, K.L., Chandrani, V., Begum, S.S., Devi, M.P.: Home automation and security system with node MCU using internet of things. In: 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), Vellore, India, pp. 1-5 (2019). https://doi.org/10.1109/ViTECoN.2019.8899540

6. Mtshali, P., Khubisa, F.: A smart home appliance control system for physically disabled people. In: 2019 Conference on Information Communications Technology and Society (ICTAS), Durban, South Africa, pp. 1–5 (2019). https://doi.org/10.1109/ICTAS.2019.8703637
7. Stefanov, D.H., Bien, Z., Bang, W.-C.: The smart house for older persons and persons with physical disabilities: structure, technology arrangements, and perspectives. IEEE Trans. Neural Syst. Rehab. Eng. **12**(2), 228–250 (2004). https://doi.org/10.1109/TNSRE.2004.828423
8. Behera, S., Saha, A.K., Kumar, D., Polai, J.: Home automation control system using SMS. Int. Res. J. Eng. Technol. (IRJET) **4**(3) (2017). e-ISSN: 2395 -0056
9. Tulshan, A.S., Dhage, S.N.: Survey on virtual assistant: google assistant, siri, cortana, alexa. In: 4th International Symposium SIRS 2018, Bangalore, India, 19–22 Sept 2018. Revised Selected Papers (2019). https://doi.org/10.1007/978-981-13-5758-9-17
10. Këpuska, V., Bohouta,G.: Next-generation of virtual personal assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home). In: 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, pp. 99–103 (2018). https://doi.org/10.1109/CCWC.2018.8301638
11. John Livingston, J., Amala Princeton, X., Jesan Gold Immanuel, T.S., Chandra Athithan, S., Abishek A.: Digital assistance for physically disabled people based on voice control system. Int. J. Emerg. Technol. Innov. Res. issue 4, pp. 514–520 (2020)
12. Lalitha, V.K., Mahalakshmi, B., Madhusudan, S., Srinivasaperumal, M., Srikanth, S., Kumar, S.R.: Smart control of home amenities using google assistant and clap switch circuit. In: 2019 5th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, pp. 350-352 (2019). https://doi.org/10.1109/ICACCS.2019.8728535
13. Kishore Kodali, R., Rajanarayanan, S.C., Boppana, L., Sharma, S., Kumar, A.: Low cost smart home automation system using smart phone. In: 2019 IEEE R10 Humanitarian Technology Conference (R10-HTC)(47129), Depok, West Java, Indonesia, pp. 120–125 (2019). https://doi.org/10.1109/R10-HTC47129.2019.9042467
14. Madhu, G.M., Vyjayanthi, C.: Implementation of cost-effective smart home controller with android application using node mcu and internet of things (IOT). In: 2018 2nd International Conference on Power, Energy and Environment: Towards Smart Technology (ICEPE), Shillong, India, pp. 1–5 (2018). https://doi.org/10.1109/EPETSG.2018.8659128
15. Saikrishna, M., Vijaykiran, G.: IOT Based Home Electrical Appliances Control Using Node MCU **6** (4), 0783-0788 (2014). ISSN 2319-8885
16. Bhandari, R., Patel Darshan, M., Dhimar Richi, K., Patel, K.: A smart home with NodeMCU using IOT. Int. J. Comput. Appl. (0975 – 8887) 176(17) (2020)
17. Miraj Shekh, M., Asha, S.R., Hariprakash, H.: IoT based home automation using NODE MCU. Int. J. Eng. Sci. Comput. **8**(5)
18. David, N., Chima, A., Ugochukwu, A., Obinna, E.: Design of a home automation system using arduino. Int. J. Sci. Eng. Res. **6**(6), 795–801 (2015)
19. Chandramohan, J., Nagarajan, R., Satheeshkumar, K., Ajithkumar, N., Gopinath,P.A., Ranjithkumar, S.: Intelligent smart home automation and security system using arduino and wi-fi. Int. J. Eng. Comput. Sci. **6**(3) (2017). ISSN:2319-7242
20. Vaibhav, M.: Research Paper On Home Automation Using Arduino (2019)
21. Agarwal, K., Agarwal, A., Misra, G.: Review and performance analysis on wireless smart home and home automation using IoT. In: 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, pp. 629–633 (2019). https://doi.org/10.1109/I-SMAC47947.2019.9032629
22. Kodali, R., Jain, V., Boppana L.: IoT based smart security and home automation system, pp. 1286–1289 (2016). https://doi.org/10.1109/CCAA.2016.7813916
23. Nagendra Reddy, P.S., Kumar Reddy, K.T. Kumar Reddy, P.A., Kodanda Ramaiah, G.N., Kishor, S.N.: An IoT based home automation using android application. In: 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), Paralakhemundi, pp. 285–290 (2016). https://doi.org/10.1109/SCOPES.2016.7955836

24. Ramlee, R.A, Othman, M.A., Leong, M.H., Ismail, M.M., Ranjit, S.S.S.: Smart home system using android application. In: 2013 International Conference of Information and Communication Technology (ICoICT), Bandung, pp. 277–280 (2013). https://doi.org/10.1109/ICoICT.2013.6574587
25. Javale, D., Mohsin, M., Nandanwar, S., Shingate, M.: Home automation and security system using Android ADK. Int. J. Electron. Commun. Comput. Technol. (IJECCT) **3**(2), 382–385 (2013)
26. Gurek, A., Gur, C., Gurakin, C., Akdeniz, M., Metin, S.K., Korkmaz, I.: An android based home automation system. In: 2013 High Capacity Optical Networks and Emerging/Enabling Technologies, Magosa, pp. 121–125 (2013). https://doi.org/10.1109/HONET.2013.6729769
27. Yue, C.Z., Ping, S.: Voice activated smart home design and implementation. In: 2017 2nd International Conference on Frontiers of Sensors Technologies (ICFST), Shenzhen, pp. 489-492 (2017). https://doi.org/10.1109/ICFST.2017.8210563
28. Basanta, H., Huang, Y., Lee, T.: Assistive design for elderly living ambient using voice and gesture recognition system. In: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, pp. 840–845 (2017). https://doi.org/10.1109/SMC.2017.8122714
29. Elshafee, A., Hamed, A.K.: Design and Implementation of a Wi-Fi Based Home Automation System, p. 2177. Engineering And Technology, World Academy Of Science (2012)
30. Celebre, A.M.D., Dubouzet, A.Z.D., Medina, I.B.A., Surposa, A.N.M., Gustilo, R.C.: Home automation using raspberry Pi through Siri enabled mobile devices. In: 2015 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), Cebu City, pp. 1–6 (2015). https://doi.org/10.1109/HNICEM.2015.7393270
31. Tharaniya Soundhari, M., Brilly Sangeetha, S.: Intelligent interface based speech recognition for home automation using android application. In: 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, pp. 1–11 (2015). https://doi.org/10.1109/ICIIECS.2015.7192988

Live Acoustic Monitoring of Forests to Detect Illegal Logging and Animal Activity

J. C. Karthikeyan, S. Sreehari, Jithin Reji Koshy, and K. V. Kavitha

Abstract Illegal cutting of trees and poaching in the forest has become a serious issue regarding environmental conservation. Trespassing in the forest has an adverse effect on the habitat of animals. There is no effective solution for real-time detection and warning of such activity. Image-based monitoring solutions are too costly and cannot cover a wide range of areas. A novel approach of audio-based monitoring systems using deep neural learning can be proposed as a solution to this problem. A model has to be trained using various audio samples of cutting of trees, gunshots, etc., along with the outliers. There are numerous tree felling techniques and hunting techniques. In the case of methods that are known to the model, the model detects that event and hence warns the authorities. The audio samples in the dataset in the time domain are converted to the frequency domain using fast Fourier transform (FFT). This distributes the signal across corresponding frequencies. For better visualization of features, it is then converted into a Mel scale, and the spectrum of this spectrum is computed using cosine transformation to obtain the Mel-frequency cepstral coefficients. Relevant features are then extracted using these coefficients and classify them using the proposed deep neural learning method. There is a significant difference between the energy concentration distributions of the sound that has to be detected with that of the outliers. This enables to classify the audio samples with a greater signal-to-noise ratio. The resulting model is then used for live monitoring of forests against illegal activities. The current situation of the wildlife demands an accurate database of animal activity in a particular area. This helps both the wildlife tourism and various studies. For addressing this issue, the proposed model is also

J. C. Karthikeyan (✉) · S. Sreehari · J. R. Koshy · K. V. Kavitha
Sree Chitra Thirunal College of Engineering, Trivandrum, Kerala, India
e-mail: jckarthikeyan2016@gmail.com

S. Sreehari
e-mail: sreehariperiya@gmail.com

J. R. Koshy
e-mail: jithinrkoshy@gmail.com

K. V. Kavitha
e-mail: kavitha@sctce.ac.in

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_8

trained to detect the presence of animals, and it will accomplish it without disturbing the wildlife activity.

Keywords Deep learning · Fast Fourier transform · Mel-frequency cepstral coefficients

1 Introduction

Illegal logging is the harvesting of wood that is in violation of national regulations. This includes harvesting timber from protected areas, felling protected species, or exceeding logging quotas. Illegal logging often takes place in countries with poor governance and law enforcement, especially in tropical rainforests. The impacts of illegal logging are varied, ranging from unchecked deforestation to the deprivation of sustainable livelihood opportunities for local communities. It accounts for over 70% of some countries' timber exports. At present, the action that prevents logging relies on patrol and reporting by witnesses. This way is very limited in time and space. In addition, the people who cut forest unlawfully take high skills in cutting and transportation, which increases the difficulty of patrol. Using satellite remote sensing is another kind of detection method, but satellite remote sensing can only find large and obvious areas of illegal felling. It is difficult to find accurate illegal felling at first time, and hardware construction and operating costs of satellite remote sensing are expensive to afford. There is no effective solution for real-time detection of such behavior.

Wildlife tourism, in its simplest sense, is interacting with wild animals in their natural habitat. This type of tourism has grown dramatically in recent years. Wildlife watching tourism can have adverse effects on wildlife by causing changes in their behavior, changing their physiology, or damaging their habitat. Tourists are often particularly keen to watch mother–offspring groups, and therefore, great care is needed to limit and control any tourism around them. Also, every year, a large number of wildlife tourists become prey to wild animals. An approach based on live acoustic monitoring is proposed to counter both these problems. Acoustic sensing at timely and correctly detection of the events is the main objective of the system.

2 Objectives

The core objectives of live acoustic monitoring are as follows:

- Report suspicious activities without human supervision.
- Time and cost efficient.
- Real-time monitoring.
- Eliminate any possibility of human negligence.
- Increase effectiveness and reliability of the existing surveillance system.

3 Related Works

One of the early methods of acoustic detection is based on the energy threshold of the incoming audio signals [1]. The idea is based on the statistical method of similarity threshold information extracted from the FFT spectrum of the voice signal. Since it is an easy method, the computational complexity is low and the detection speed is quick but has less accuracy. In this method, the detection is done using the analysis of the sound energy distribution. From observation, it was evident that in the sound of logging by chainsaw, 75% of the energy is distributed in the region between 2400 and 2900 Hz, whereas in the sound of wind, energy is concentrated in the region around 1000 Hz. This is the information that is used to identify logging activity. This method is not an intelligent surveillance system and cannot classify many different sounds. It uses a simpler approach and does not take noise into consideration. It has an accuracy of only 70%.

Over the years, many researches have been done on live acoustic classification, and this resulted in an efficient method of converting audio data into image representation using spectrograms, and these spectrograms are then used to classify the sounds [2]. This method has much greater accuracy than the energy threshold method. This method uses two sets of features to describe the audio signals: mean values of Mel-frequency cepstral coefficients and energy of discrete multilevel wavelet decomposition. Mel-frequency cepstral coefficients (MFCCs) are short-term spectral-based features. Their success for speech recognition has been due to their ability to represent the speech amplitude spectrum in a compact form. The wavelet transform is a tool that cuts data or functions or operators into different frequency components and then studies each component with a resolution matched to its scale. These features obtained are then fused together to form a single set of features. It is then used to train the neural networks against different classes of sounds. It considers four different classes of sounds, associated with possible dangerous situations: chainsaw class (C), gunshot class (G), human voice class (H), and tractor class (T). The method proposed in [2] uses two ANN—one to identify whether the event is an alarming one or not. If alarming, then the second ANN is used to identify which type of alarming event it is.

The problem of animal sound classification can be done using CNN together with deep learning [3]. It is a slight variation of the MFCC method. The system proposed is based on convolutional neural network architecture. The advantage of the method proposed here is the usage of CNN in the neural network. A two-layer convolution network can achieve higher accuracy when compared to a multi-layer perceptron (MLP), a single-layer CNN, and a Gaussian mixture model–Mel-frequency cepstral coefficient (GMM-MFCC) system. The deep learning model proposed here consists of three convolution layers with a maxpooling layer and three dense layers (fully connected layer). Three convolution layers and two dense layers use rectified linear units (ReLUs) as the activation function. The final dense layer uses softmax as the activation function.

Another method of live acoustic monitoring is done using RNN [4]. It is similar to the CNN method. The variation here is that recurrent neural network is used here instead of the convolutional neural network. In audio processing, recurrent neural network (RNN) achieved human-level performances on speech recognition. There are many variations of RNN. One of the most successful variants is the long short-term memory (LSTM) RNN. LSTM cell is an RNN which uses additional parameters (i.e., the cell state) as a recurrent input.

4 Methodology

4.1 Dataset

We collected our dataset from AudioSet. This dataset is designed by the Sound Understanding group in the Machine Perception Research organization at Google. It consists of an expanding ontology of 632 audio event classes and a collection of 2,084,320 human-labeled 10 s sound clips drawn from YouTube videos. The ontology is specified as a hierarchical graph of event categories, covering a wide range of human and animal sounds, musical instruments and genres, and common everyday environmental sounds. The dataset and machine extracted features are available for download. The Machine Perception team at Google is dedicated to teaching machines to accurately perceive audio by building state-of-the-art machine learning models, generating large-scale datasets of audio events, and defining the hierarchical relationships that exist between sounds.

4.2 Data Preprocessing and Augmentation

The audio samples downloaded from AudioSet were not standardized and hence needed preprocessing. Audacity was used to preprocess the data. The audio files were converted from stereo to mono-channel. These were then sampled every three seconds. But for some classes that did not give us enough samples to build a comprehensive dataset, data augmentation needs to be done. We augmented the data by speed tuning and by adding white noise. This helped us in generating a more comprehensive dataset. The following is the list of different audio classes that were collected.

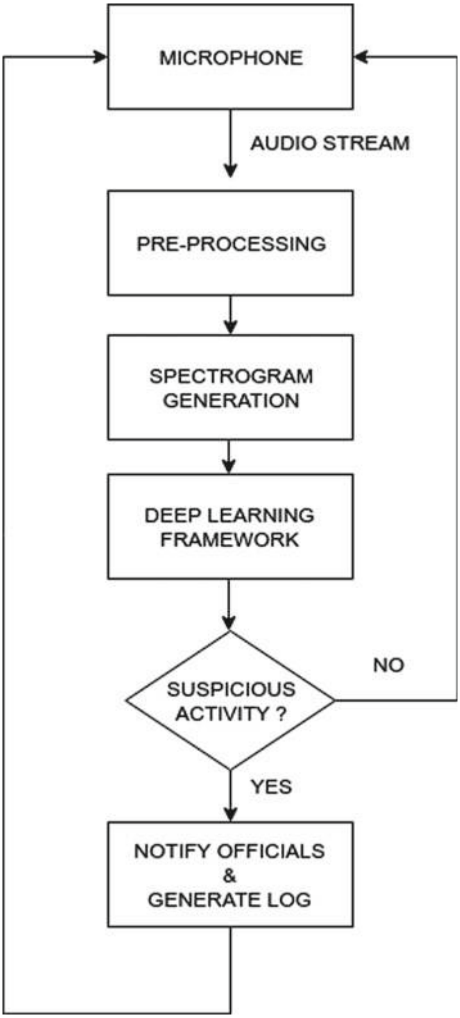
- 374 sound samples of chainsaw each of three-second duration and then did data augmentation to generate another 301 samples, thus, in total, 675 samples.
- 304 samples of elephant and then further did data augmentation to generate another 131 samples, thus, in total 435 samples.
- 150 samples of shotgun and did data augmentation to generate a total of 553 samples.
- 703 samples of wolf howls.

- A class of random white noise to capture the ambient atmosphere of forest.
- Apart from these two outlier classes to avoid ambiguous classifications.

4.3 System Architecture

The architecture has three broad divisions which are the data acquisition state, deep learning state, and alert/notification state as depicted in Fig. 1. Audio data streamed by the microphone is locally processed to generate spectrograms. The audio signal in the time domain is transformed into the frequency domain. This distributes the

Fig. 1 System architecture



signal across corresponding frequencies. The frequencies are converted to Mel scale. These spectrograms are stored in pandas data frame. These are then standardized and transferred to the deep learning block for detecting whether there is any suspicious activity or animal activity. If any such activity is detected, then it is notified to the corresponding officials through email, and also, a log of such activities is kept in the database along with associated timestamp. This log of events is made available for the public through a Web site. This makes this system transparent and much more useful.

4.4 Feature Extraction

The right feature will determine the accuracy of the model. The Mel-frequency cepstral coefficients are considered as a signature for each sound. The audio signal in the time domain is transformed into the frequency domain. This distributes the signal across corresponding frequencies. The frequencies are converted to Mel scale. Mel scale is a scale that relates the perceived frequency of a tone to the actual measured frequency. Then, the spectrum of this spectrum is computed using cosine transformation to obtain the Mel-frequency cepstrum. The coefficients against time in this spectrum are referred to as the Mel-frequency cepstral coefficients. The cepstrum is a gradient of spectral bands and hence constitutes a feature unique to the signal. The whole process of feature extraction has been implemented using the Python package 'Librosa.' The extracted features are stored in the pandas data frame. These are then standardized for the features to be on a similar scale. To reduce the dimension without compromising the features, the dataset has undergone a dimensionality reduction technique—principal component analysis. As a result, the samples are reshaped into a (128, 128, 1) array (Fig. 2).

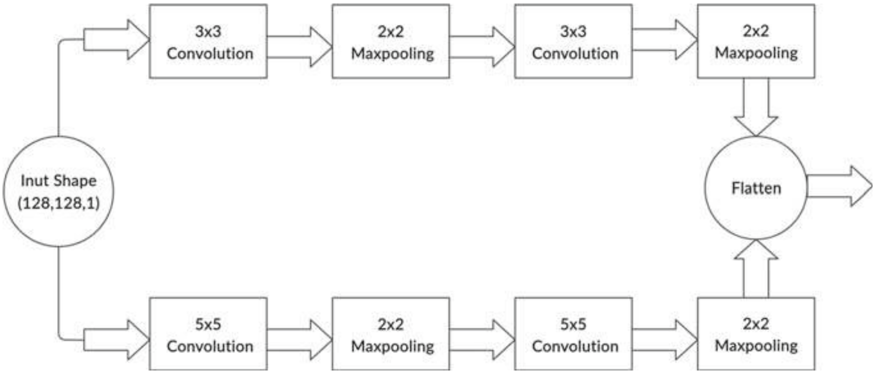


Fig. 2 CNN architecture (1)

4.5 Training

A variant of the convolution neural network (CNN) is used. It is observed that a three-second audio sample is enough to detect the event. The feature for each sample is a (128, 128) matrix. It should be carefully trained as two different sounds may have many similarities. Also, in three-second audio, the event can occur in any frame of the audio. A CNN convolutional layer with an m*m filter will detect a feature different from some other n*n filter. So to leave no confusion for the model, a parallel model similar to the inception model is employed. It is expected to capture the event occurring anywhere within the total 3 s, and the parallel model combining two sequential models would be able to differentiate audio with small differences. It consists of two parallel layers, each of which is a sequential layer taking the same input. The first layer consists of a 3 × 3 convolutional layer followed by a 2 × 2 maxpool layer. This same sequence has been repeated one more time to form the whole sequential model. The batch size is 128 and the input shape is (128, 128, and 1). The other sequential model of the parallel layer is the same as the former layer except for the convolutional layers. It has got a 5 × 5 filter for the convolutional layers instead of 3 × 3. The activation function used in all the layers is ‘tanh.’ A stride of (2, 2) is used.

These two sequential models are combined and flattened to form a feature set of 12,800 features. This forms the input layer. It is followed by a dense layer of 1024 nodes and again a dense layer 512 nodes and then the output layer. The output layer has got seven nodes as of now depending upon the classes selected. A dropout of value 0.25 is used as shown in Fig. 3. The sigmoid activation function is used in the hidden layer, and the softmax activation function is used in the output layer.

The CNN model so formed will take two inputs and produce output in any one output node. The optimizer used for training is ADAM. It enables the model to converge faster by reducing the oscillation at the global minimum of the loss. The loss is computed using categorical cross-entropy. It is given by,

$$\sum_{c=1}^{M} y_{o,c} \log\left(p_{o,c}\right) \tag{1}$$

- M—number of classes.
- log—the natural log.
- y—indicator (0 or 1) if class label c is the correct classification for observation ‘o.’
- p—predicted probability on observation ‘o’ of class ‘c.’

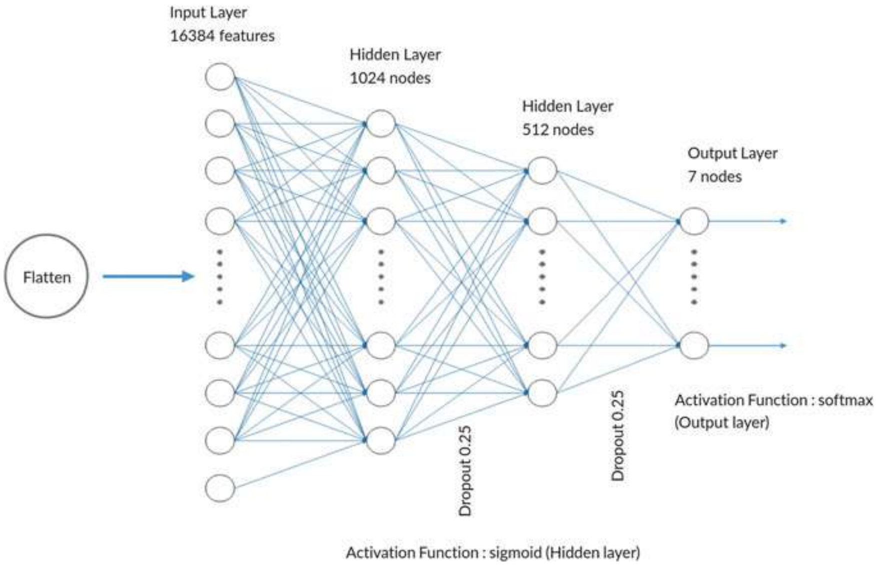


Fig. 3 CNN architecture (2)

4.6 Deployment

The whole process of processing audio and detecting the events is happening at the Apache server. The microphone captures the audio signal continuously. The collected signal is sent to the server. The deep learning model takes the signal and predicts the event. The result will be stored in the database. From there, logs of animal presence will be provided in the associated Web site, and the illegal activities are notified by email to responsible authorities.

4.7 Technology Stack

The technology stack for the proposed system is illustrated in Fig. 4. The deep learning model is implemented in Keras with Tensorflow as a backend. TensorFlow is a symbolic math library for dataflow and differentiable programming across a range of tasks. Keras is a high-level API for building and training deep learning models. Then, Pandas, Librosa, Pyaudio, and Numpy are used to retrieve and process the data. The Web application is developed using HTML and CSS as frontend. PHP is the server scripting language. The application is hosted on Apache server. MySQL is the database used.

We have two databases for this project. The first one has one table, and it is to this table the different microphones from different locations send their audio samples

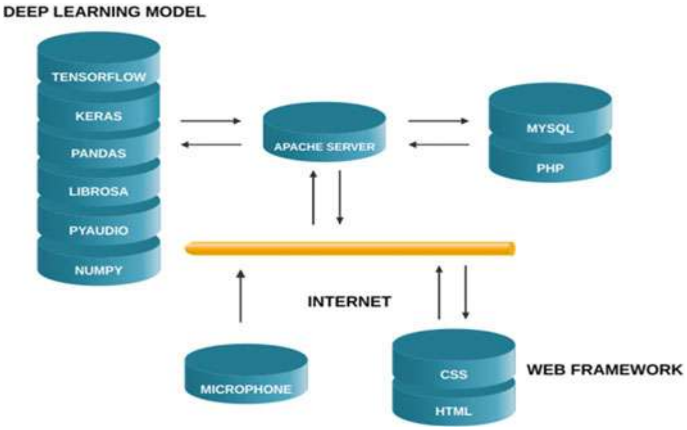


Fig. 4 Technology stack

once they are disturbed by the noise of any event. The schema of this table contains date, time, location, and the audio sample stored in MEDIUMBLOB format. The other database has two tables in it—one for saving the log of illegal activities and the other one for saving the log of animal activities. Both of these tables have the same schema. It contains date, time, location, and the event. There are no dependencies between any of the specified tables.

5 Result and Discussions

The model has been trained and tested with a dataset of 4333 samples. It is split into a training set of 3466 samples and a testing set of 867 samples. It took only 10 epochs to converge. It has got a training accuracy of 100% and a testing accuracy of 99.77%. The model accuracy plot is shown in Fig. 5. To evaluate the performance of the model, it is compared against a KNN approach suggested in [5]. For that purpose, a new dataset of the same class but with a relatively high amount of white noise is prepared. This high noise will make it difficult for the model to predict. The proposed model and KNN model are tested against the new dataset. The KNN approach has got an accuracy of 54.75%, whereas our proposed model has achieved an accuracy of 75.25%. The ROC curve for this comparison for the class chainsaw is shown in Fig. 6. To evaluate the performance of the model against its corresponding sequential model, the same dataset with high noise is tested against the sequential models of our model. The sequential model with a 3*3 filter has got an accuracy of 63.93%, and the sequential model with 5*5 filters has got an accuracy of 65.09%. Hence, our parallel model, which is the combination of the previous two sequential models, has got better accuracy (Table 1).

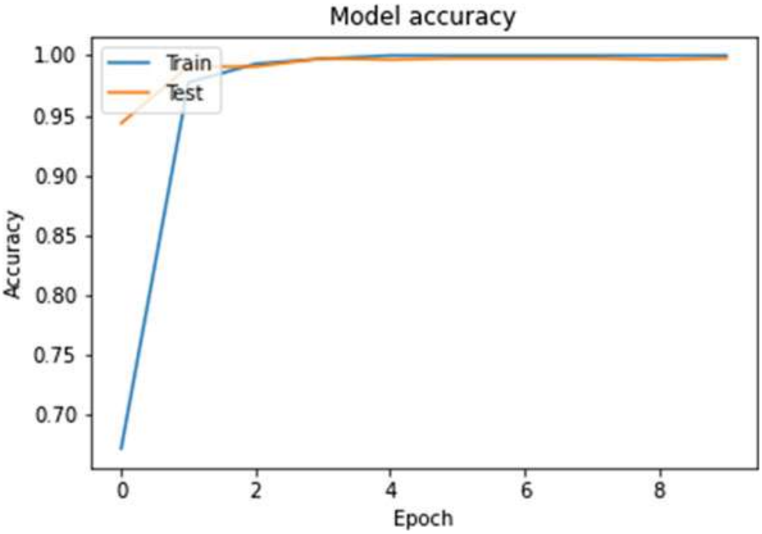


Fig. 5 Model accuracy versus epoch plot

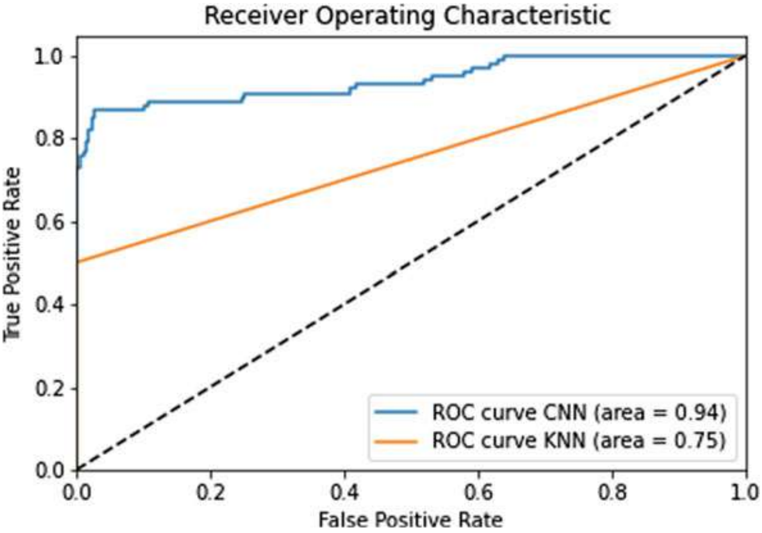


Fig. 6 ROC curve for class chainsaw

6 Conclusion

The increasing exploitation of the forest is demanding an automated real-time detection system. Considering the wide range of areas and cost limitations, our model is providing a better solution to the problem. It will be nearly impossible for the forest

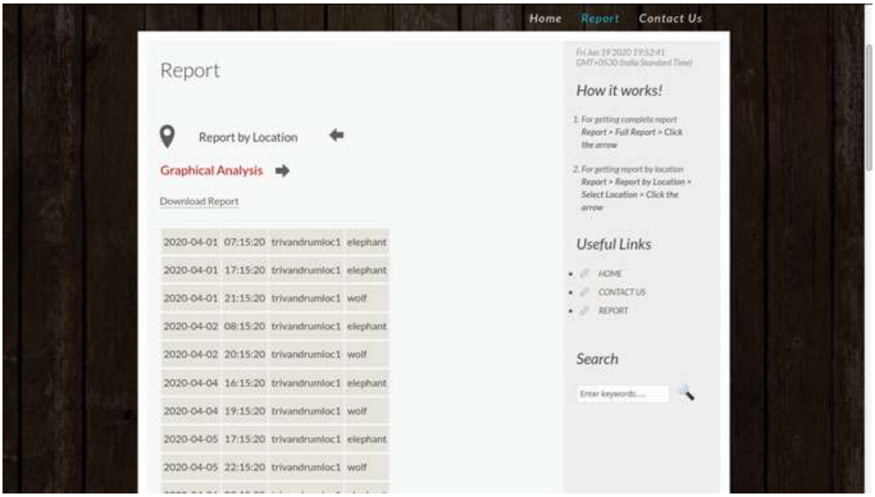
Table 1 Comparison using prepared noisy dataset

Model	Accuracy (%)
Proposed model	75.25
Sequential model of 3*3 filter of the proposed model	63.93
Sequential model of 5*5 filter of the proposed model	65.09
KNN model	54.75

officials to patrol the whole forest for ensuring protection against illegal activities. They may take even a day for checking a part of the forest. Our proposed system plans to deploy a nexus of embedded system consisting of a microphone and a connectivity to a central database. This system is connected to the server via a network. Its cost is way much lesser as compared to a video surveillance system. This allows effective surveillance over a wide range of areas. Apart from this, the system is providing a detailed analysis of the presence of animals in different geographical areas. It is thus providing a platform for data analysis and environment protection without a need for physical presence. This helps to preserve the tranquility of nature. The parallel architecture of the deep learning model is differentiating one class of sound from another quite effectively. Though the architecture is complex, the system is not that much heavier as compared to a normal model. It is capable of running on a normal CPU environment with a standard specification. The system is designed in such a way that there is less amount of local caching. So it will not encounter a performance bottleneck in the long run. The Web application will provide detailed information about the animal events which can be sorted based on date and time. It is providing various insights that can be beneficial for research purposes.

Appendix

See Figs. 7 and 8.



Report

Report by Location

Graphical Analysis

Download Report

2020-04-01	07:15:20	trivandrumloc1	elephant
2020-04-01	17:15:20	trivandrumloc1	elephant
2020-04-01	21:15:20	trivandrumloc1	wolf
2020-04-02	08:15:20	trivandrumloc1	elephant
2020-04-02	20:15:20	trivandrumloc1	wolf
2020-04-04	16:15:20	trivandrumloc1	elephant
2020-04-04	19:15:20	trivandrumloc1	wolf
2020-04-05	17:15:20	trivandrumloc1	elephant
2020-04-05	22:15:20	trivandrumloc1	wolf

Fri Jun 19 2020 19:52:41 GMT+0530 (India Standard Time)

How it works!

1. For getting complete report Report > Full Report > Click the arrow

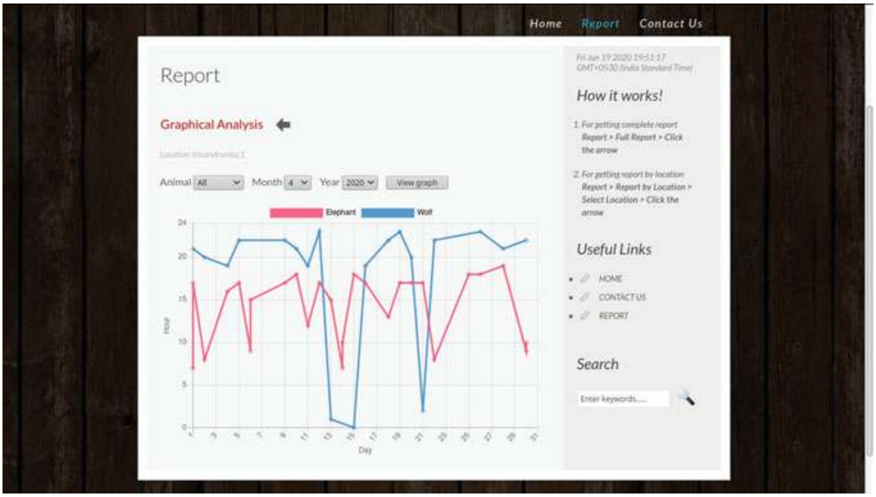
2. For getting report by location Report > Report by Location > Select Location > Click the arrow

Useful Links

- HOME
- CONTACT US
- REPORT

Search

Fig. 7 Report provided by web site



Report

Graphical Analysis

Location trivandrumloc1

Animal: All · Month: 4 · Year: 2020

Fri Jun 19 2020 19:51:17 GMT+0530 (India Standard Time)

How it works!

1. For getting complete report Report > Full Report > Click the arrow

2. For getting report by location Report > Report by Location > Select Location > Click the arrow

Useful Links

- HOME
- CONTACT US
- REPORT

Search

Fig. 8 Analytics provided by web site

References

1. Tang, Y., Han, P., Wang, Z.: Based on intelligent voice, recognition of illegal forest felling detecting methods. In: Proceedings of IEEE CCIS, 2012
2. Oltean, G., Grama, L., Ivanciu, L., Rusu, C., Neri, A.: Alarming events detection based on audio signals recognition. IEEE Trans. Acoust. (2015)
3. Boray Tek, E.Ş.F.: Animal sound classification using a convolutional neural network. In: 3rd International Conference on Computer Science and Engineering (2018)
4. Colangelo, F., Battisti, F., Carli, M., Neri, A.: Enhancing audio surveillance with hierarchical recurrent neural networks. In: IEEE AVSS, August 2017, Lecce, Italy (2017)
5. Chaudhary, M., Prakash, V., Kumari, N.: Identification of vehicle movement detection in forest area using MFCC and KNN. In: Proceedings of the IEEE SMART 2018 (2018)

CATS: Cluster-Aided Two-Step Approach for Anomaly Detection in Smart Manufacturing

Dattaprasad Shetve, Raja VaraPrasad, Ramona Trestian, Huan X. Nguyen, and Hrishikesh Venkataraman

Abstract In the age of smart manufacturing, there are typically multitude of sensors that are connected to each assembly line. The amount of data generated could be used to create a digital twin model of the complete process; wherein virtual replicas of the device and the process can be created before and during the process. An important aspect is automatic anomaly detection in the manufacturing process. Anomaly/outlier detection identifies data-points, events and/or observations that deviate from the dataset's normal behaviour. A major problem in predicting anomaly from datasets is the limited accuracy that can be achieved. Several state-of-the-art techniques provide very high accuracy (>95%). However, these result in a considerable increase in the required time, thereby limiting its use to non-real-time applications. This paper proposes a cluster-aided two-step (CATS) approach for anomaly detection wherein two unsupervised detection techniques are employed in serial. The technique used for the first step is density-based spatial clustering for applications with noise (DBSCAN), while the second technique is local outlier factor (LOF). The output of the first-step technique is fed to the second technique, thereby utilizing the knowledge generated in the first step. An extensive simulation analysis indicates that the proposed CATS algorithm results in >95% accuracy for the outlier population is above 15% with a prediction time of lesser than 85 s.

Keywords Anomaly · Clusters · Two-step · Smart manufacturing

D. Shetve (✉) · R. VaraPrasad · H. Venkataraman
Indian Institute of Information Technology, Sri City, India
e-mail: dattaprasad.s@iiits.in

R. VaraPrasad
e-mail: yrv.prasad@iiits.in

H. Venkataraman
e-mail: hvraman@iiits.in

R. Trestian · H. X. Nguyen
School of Computing, Middlesex University, London, UK
e-mail: r.trestian@mdx.ac.uk

H. X. Nguyen
e-mail: h.nguyen@mdx.ac.uk

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_9

1 Introduction

Smart manufacturing is a broad category of facility that employs sensors, communication and computer-integrated manufacturing to have a high level of adaptability, rapid design changes, digitization in manufacturing process and a more flexible assembly line. Specifically, Industry 4.0 refers to the use of a large-scale machine to machine communication (M2M) and Internet of things (IoT) deployments to provide increased automation, improved communication and self-monitoring, as well as smart machines that can analyse and diagnose issues without the need for human intervention (refer Fig. 1) [1]. In an Industry 4.0 world, it is essential to have zero downtime by predicting potential anomalies. An anomaly is defined as the occurrence of an event that is different from the normal expectation [2, 3]. It is the deviation of a metric/parameter in a system from its normal, pre-defined behaviour. This behaviour is identified by prediction, general observation or identifying patterns. A system shows an abnormal behaviour either when any faults/errors occur or whenever the system, as a whole, sets a new trend in its outcome.

In this regard, machine learning and other techniques have been recently applied to detect anomaly in the manufacturing process. The state-of-the-art techniques mostly deal with feeding the data-points into an automated learning system, wherein the characteristics can be identified from the data-points. However, a major problem of the use of anomaly detection is the amount of accuracy that could be obtained. Typically, as the anomaly increases, the prediction accuracy goes down significantly. On the other hand, if there has to be a high accuracy (say >90%) irrespective of the outliers, the time taken for anomaly prediction increases exponentially, going beyond a few seconds or several minutes. This is a significant problem as a smart manufacturing environment cannot lose accuracy and also cannot wait for several minutes to detect an anomaly. A smart manufacturing ecosystem should detect the

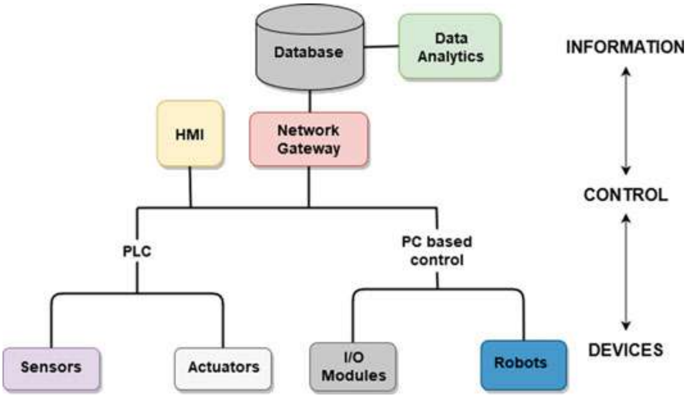


Fig. 1 Block diagram for an IoT-based M2M communication set-up in a smart manufacturing system

anomaly within a minute or so; so that the assembly line in the shop-floor is not disturbed. It is this particular challenge that has been addressed in this paper. This paper proposes and analyses a clustering-based two-step approach (CATS) for machine learning-based mechanisms to increase accuracy and also reduce the time required for anomaly detection. CATS not only results in a very high accuracy of greater than 97.5% for an outlier of 15% but also detects the anomaly in less than 1.5 min.

The paper is organized as follows: Sect. 2 describes the related work on two-step detection. Section 3 describes the CATS approach adopted in this work and explains the technique in detail. The simulation model and experimental results are provided in Sect. 4, while Sect. 5 concludes the paper.

2 Related Work

There has been considerable work done on using a two-step approach for anomaly detection. The authors in [4] proposed a statistical approach wherein they built a probabilistic model for normal instances in step I. An instance was declared as an anomaly if the value of its discriminant function outweighs a pre-defined threshold. The anomalies were re-clustered in step II using a similarity distance metric. A similar study [5] utilized an auto-correlation function to detect dynamical changes in a temporal dataset to isolate outlier candidates in the first stage and evaluated the extent of outlierness by calculating its average, variance and kurtosis in the second stage. Notably, a study carried out in [6] achieved higher accuracy and lower time complexity by eliminating the normal instances by working on a smaller dataset of anomaly candidates. In this scenario, a filtering process was employed wherein a deterministic space partition (DSP) technique constituted step I, while a density-based refinement process that assigns attributes to candidates constituted step II.

An important study [7] used the RA-Clust technique to define a clustering centre in step I and Kernel Density Estimation for Synthetic Hypothesis Testing that detects the outliers using defined rules of hypothesis. Gupta et al. [8] worked on a series of snapshots of a temporal dataset and detected evolving communities. They evaluated the probability distribution of every community of interest to check its continuous behaviour (step I) and calculated its match score for classification as a normal or anomalous instance (step II). Another study [9] was conducted to detect anomalous readings of an electrocardiogram (ECG). The authors isolated length patterns by segmentation with the help of quadratic regression model (step I) and later calculated the anomaly scores of patterns using dynamic time warping technique which detected dissimilar patterns (step II). However, most of these techniques focused either on accuracy or on real-time detection, but not on both aspects. In the next section, the proposed CATS approach for anomaly detection is presented and explained in detail.

3 CATS—Clustering Aided Two-Step Approach

The CATS algorithm is designed, keeping in mind two novel aspects: Firstly, the anomaly detection is carried out in both the steps. Secondly, the second step uses the results of the first step and works with only the remaining data-points, so as to improve the overall performance.

3.1 Techniques Used for CATS

The aim of CATS is to identify anomalous occurrences in the data at hand. In the case of a bidimensional dataset, anomalies occur as single points of faults or errors called 'Point Outliers'. Such faults are easily identified by the clustering-based techniques with the aid of well-defined clusters. CATS works on the principle that the first step has to be a clustering-based approach, wherein all the broad range outliers are first detected. There are different techniques, including K-means clustering, K-nearest neighbour (KNN), etc., that could be used. However, a simple performance evaluation of outlier detection revealed that DBSCAN far outperformed KNN in accurately detecting the outliers. DBSCAN utilizes the concept of relative density to generate clusters of normal points. In contrast, KNN implements a distance-based outlier detection mechanism. The distance metric, being vulnerable to an environment of variable density, proves to be inefficient. The density evaluation in DBSCAN at every instance in the dataset ensures high accuracy [2, 10]. The same concept is exploited by LOF; wherein the ratio of the relative density of an instance to the averaged densities of the neighbourhood points is computed to generate a suspicious index [2, 11]. Such extensive utilization of neighbourhood knowledge makes LOF a suitable choice for precise outlier detection. Studies in [12, 15] indicate that DBSCAN and LOF outperform existing techniques like K-means, KNN in terms of accuracy. The ability to function without prior knowledge of the number of clusters, to generate clusters that are mutually exclusive and to bear a concept of noise while clustering makes DBSCAN a better choice for outlier detection study. The main research questions we tried to address in this paper are (I) To what accuracy is CATS able to detect the outliers in a given dataset, and (II) To what limit of outlier population is CATS able to maintain the accuracy. The techniques utilized in this study are described as follows:

3.1.1 Clustering with DBSCAN

The procedure is carried out as follows:

1. In DBSCAN, a bidimensional dataset is initially fed as input. Each of the data-points is labelled for accuracy evaluation.

2. CATS is keyed in with (i) The dataset, (ii) The initial value of parameters of DBSCAN technique namely, ϵ (epsilon) and $Minpts$ and (iii) Expected Accuracy.
3. DBSCAN evaluates its parameter values for every data-point based on the pre-defined neighbourhood connectivity rules and groups those instances together that obey the rules.
4. These groups are declared as mutually exclusive 'Clusters', while the excluded points are flagged as 'Outliers'.

3.1.2 Refined Search with LOF

LOF is a density-based outlier detection method that is used in the second step of the two-step approach. The main criteria for classification in LOF are the relative density of a point in its neighbourhood. The density of a data-point is decided by the parameters k and $threshold$. Based on these, a suspicious index is generated by LOF. Further, the outliers are classified as points with the value of the index greater than 1.

3.2 Mechanism of CATS

The detailed procedure and mechanism of CATS is explained in the CATS algorithm. CATS begins with an input of a bi-variate dataset. The labelled part is extracted out, and the actual data part is fed to DBSCAN as it is an unsupervised technique. It follows its pre-defined neighbourhood rules and classifies the data-points into clusters and outliers, through a multiple iteration process to update the parameters. It compares the outliers in the output with the labelled ones. This comparison becomes a critical basis for CATS to (i) Evaluate accuracy of steps I and II and (ii) Decide whether to continue with iterations or terminate the operation.

In very rare instances, it is observed that all of the labelled outliers are accurately detected in the first step. Hence, the knowledge of clusters from the first step is utilized to assist the second step. The detected outliers are removed, and the remaining dataset is fed to step II for an extensive refined search. To facilitate this experiment and test the accuracy of LOF exhaustively, an intermediate step is introduced called 'Data Mixing Stage'. For N clusters generated by step I, N smaller combinations of datasets are created. Every small dataset consists of one of the N clusters and outliers which remained undetected at the end of step I.

The cluster-outlier combinations are fed to LOF, to detect residual outliers. The LOF is executed for each of these mixed datasets, generating detection accuracy results for every iteration. Step II is completed after N iterations. The average of all results decides the overall accuracy of step II. Figure 2 describes the entire CATS detection procedure. The overall accuracy is decided from the outliers detected in both steps.

Algorithm 1: CATS

```
Input: Labelled Two-dimensional Dataset
Output: Detection of Outliers with expected accuracy
Initialize values for epsilon, $\epsilon$ and $Minpts$
Feed the data to DBSCAN                      ▷Start of Step I
while Accuracy, Ac <95% do
    Run DBSCAN
    Compare generated and labelled outliers
    Evaluate $Ac$
    if $Ac > 95\%$ then
        Jump out of loop
    Update $\epsilon$ and $Minpts$
Save the list of undetected outliers         ▷End of Step I
for $i^{\text{th}}$ cluster from 1 to N do
    Create Cluster + Outlier data            ▷Data Mixing Stage
for $i^{\text{th}}$ cluster from 1 to N do
    Feed Cluster $i$ + outlier dataset       ▷Start of Step II
    Initialize $k$ and $threshold$, $Th$
    while $Ac < 95\%$ do
        Run LOF
        Compare generated and labelled outliers
        Evaluate $Ac$
        if $Ac > 95\%$ then
            Jump out of loop
        Update $k$ and $Th$
    Save results
Evaluate average $Ac$ of Step II             ▷End of Step II
Incorporate both results and evaluate overall accuracy of CATS
```

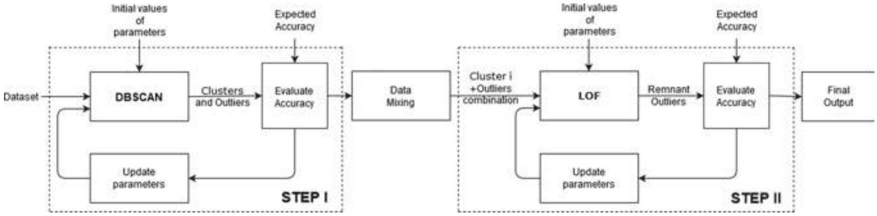


Fig. 2 Overview of the CATS process

4 Model and Experimental Results

4.1 Experimental Model

In order to analyse the performance of CATS, the data-points are initially generated using synthetic data. The outlier population in the dataset is systematically varied from 10 to 25% in steps of 1%. Importantly, a bidimensional dataset of nearly 3000 data-points with 20 clusters was chosen for the study. The detected outliers are utilized for calculation of Type-I errors (false outliers) ($T1$), Type-II errors (missed outliers) ($T2$) and accuracy (A). The detected outliers are the common instances (C) between generated outliers (Q) and labelled outliers (L). Further, the values are calculated as follows:

$$T1 = Q - C \tag{1}$$

$$T2 = L - C \tag{2}$$

$$A(\%) = \frac{\text{length}(L) - \text{length}(T1) - \text{length}(T2)}{\text{length}(L)} * 100\% \tag{3}$$

The time taken for the execution of individual steps and subsequently the entire algorithm was noted down, to give us a better insight on the algorithm speed. The test platform used in this study is a 2.7 GHz Intel Core i5-7200 laptop with 8 GB RAM. The entire simulation set-up was carried out using MATLAB.

4.2 Experimental Results

The result generated by the first step of the proposed CATS algorithm, DBSCAN, is shown in Fig. 3. Every individual cluster is identified separately, seen as coloured patches, and outliers are identified with red crosses. These instances are extracted out, before feeding it to step II. In step II, the LOF technique attempts to detect the residual outliers from step I (refer Fig. 4).

Table 1 depicts outliers and the accuracy of detection, observed from the simulations. For every step of the exercise, the outlier population in the dataset is incremented by 1% and fed back, as shown in Fig. 2. The third column shows the result obtained after executing step I for all outlier conditions. The result of step II is seen in column 4. The results are averaged for N clusters.

It is observed that the overall detected outliers are at par with the increasing outlier population. However, the number of outliers detected by DBSCAN in step I appears to be consistent throughout the simulations. The gap of increasing outlier population and constant step I outliers is bridged by LOF in step II. As seen from column 4, LOF plays a critical role in balancing the load of outlier detection. The most notable observation in this study is that the accuracy of CATS is (>95%) for even when

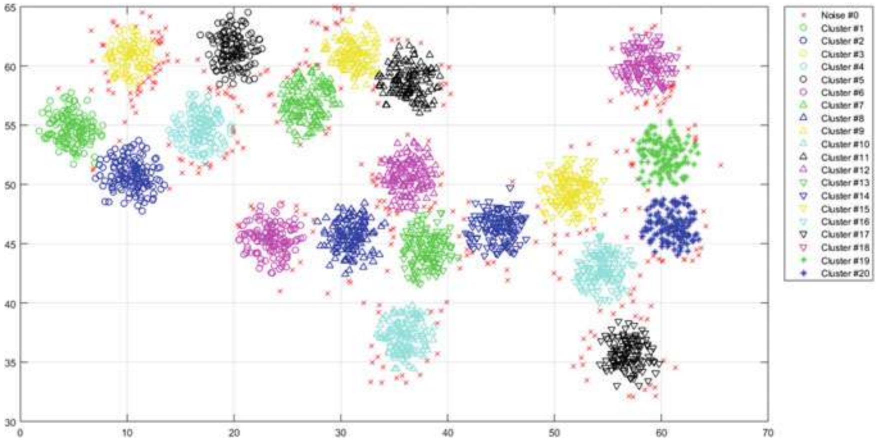


Fig. 3 Clustering result of DBSCAN after step I of CATS

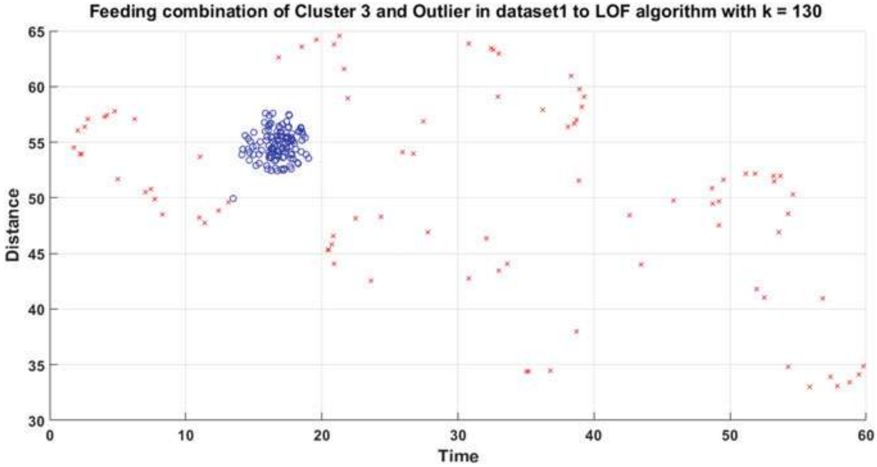


Fig. 4 Undetected outliers from step I are identified and marked as red crosses by LOF in step II

the outliers are around 16% of overall population (refer column 6 of Table 1). For outliers beyond 16%, the decline in total accuracy is witnessed. This phenomenon is better described in Figs. 5 and 6. The outliers are spotted precisely by LOF till 16% population. Figure 5 illustrates this robust operation, where cluster no. 12 is detected accurately for 15% outliers. As the population surges to 25%, LOF is not able to keep up with the increased outlier density around the cluster and the overall 2-D space of the input data, thus resulting in poor detection (refer Fig. 6). Apart from its limited outlier handling capacity, it should be noted that LOF shows a superior performance till its threshold. To further demonstrate the significant relevance of CATS' performance, it is compared with an similar, existing approach. The authors in

Table 1 Experimental results of accuracy of CATS

Outlier population (%)	No. of outliers	Detected outliers in step I	Detected outliers in step II	Total detected outliers	Total accuracy (%)
10	301	208	87	295	98.01
11	336	220	109	329	97.92
12	366	231	128	359	98.09
13	393	246	139	385	97.96
14	423	251	166	417	98.58
15	450	269	172	441	98.00
16	480	282	186	468	97.50
17	519	291	201	492	94.80
18	545	330	162	492	90.28
19	577	244	239	483	83.71
20	600	221	279	500	83.33
21	632	286	243	529	83.70
22	667	291	261	552	82.76
23	692	291	284	575	83.09
24	724	289	316	605	83.56
25	756	288	336	624	82.54

[13] proposed an algorithm for anomaly behaviour detection system with DBSCAN and LOF employed in serial. The algorithm was tested on an unmarked, official vehicle data. The algorithm achieved an accuracy of 99.12%. This proves that our proposed CATS algorithm, achieving a maximum accuracy of 98.58%, is equally competent with the existing approaches. This shows that CATS is a very reliable mechanism for outlier detection.

Significantly, Table 2 depicts the relevant computation times of CATS. These include the execution time of steps I and II, the overall time taken by CATS, along with the time needed to execute a single iteration of DBSCAN and LOF. The latter was incorporated to provide an insight about the correlation between the individual and overall execution times. The step I and step II times comprise (I) multiple iterations of the said techniques that are prone to be variable pertaining to different outlier scenarios and (II) other essential lines of code. The column ‘Total time’ signifies the sum of computation times of steps I and II.

It can be observed from Table 2 that at 10% outliers, the time taken per iteration for DBSCAN is 137.3 ms and for LOF is 1019.3 ms. This time increases for DBSCAN to 159.2 ms and drastically decreases for LOF to 95.1 ms for an outlier of 25%. The sudden decrease in execution time after 16% outlier is due to the limited outlier handling capacity of LOF. Hence, 16% of outlier population marks the limit of outlier tolerance of CATS.

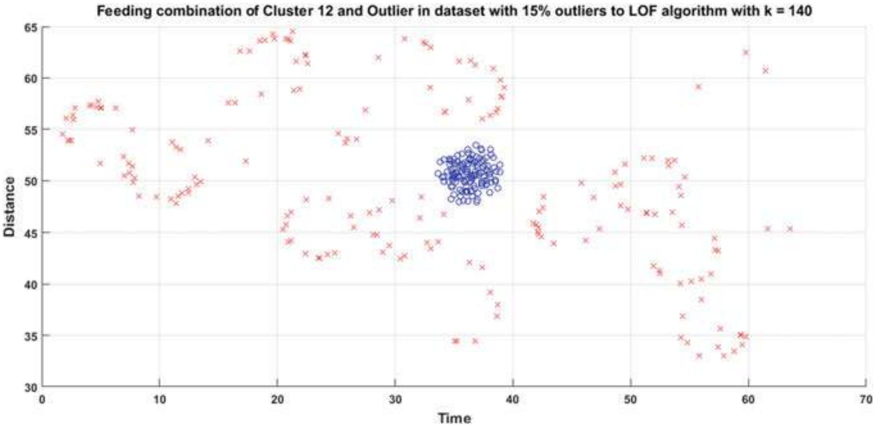


Fig. 5 Output of LOF when the outlier population is 15%

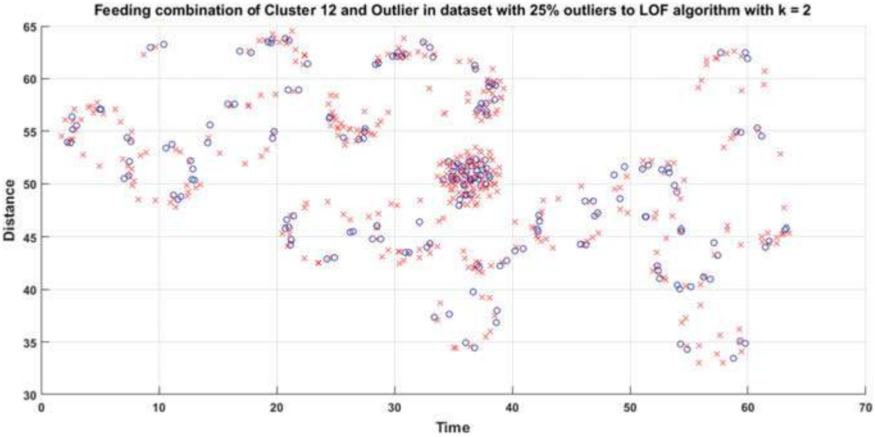


Fig. 6 Output of LOF when the outlier population is 25%

Further, it can be noted from Table 2 that using CATS, the total outliers can be detected in a time range of 60–88 s; for a dataset consisting of nearly 3000 points. For a fair comparison with the existing approaches, the computation time of CATS is compared with the state-of-the-art anomaly detection techniques in the literature [14, 15]. The authors in [14] attempted to assess the performance of DBSCAN technique as a part of their research study. It involved estimating the execution time of DBSCAN against a few bidimensional synthetic datasets with number of records varying from 50,000 to 1,000,000. Table 3 shows the computation time of DBSCAN for these datasets. For the sake of comparison with this study, we have devised a parameter T which is the ratio of 'Execution time' to the 'Number of data-points' (refer column 4 of Table 3). The value of T is observed to vary between 0.07 and

Table 2 Experimental results of Execution Time of CATS

Outlier Population (%)	DBSCAN time per iteration (ms)	Step I time (in s)	LOF time per iteration (ms)	Step II time (in s)	Total time (in s)
10	137.3	54.03	1019.9	26.3	80.33
11	135.9	53.49	1132.3	28.38	81.87
12	134.4	53.38	1125.7	28.91	82.29
13	162.8	54	1153.7	29.8	83.8
14	140.8	55.36	1297.1	33.06	88.42
15	148.5	52.57	1389.1	32.26	84.83
16	160.7	56.45	1455.0	31.6	88.05
17	139.6	59.89	478.0	21.69	81.58
18	144.2	54.26	338.6	14.97	69.23
19	151.6	53.72	295.2	8.49	62.21
20	133.5	55.01	217.5	8.23	63.24
21	143.4	58.15	195.2	7.77	65.92
22	158.6	58.6	92.6	3.02	61.62
23	167.2	61.57	95.2	3.45	65.02
24	166.1	61.24	94.3	3.11	64.35
25	159.2	57.64	95.1	3.15	60.79

Table 3 Execution times of DBSCAN as evaluated in [14]

Datasets	Points	Execution time (ms)	T (ms)
50k	50,000	3930	0.0786
100k	100,000	10,151	0.10151
200k	200,000	46,985	0.234925
500k	500,000	413,684	0.8273
1M	1,000,000	1,385,965	1.385965

1.38 ms per data-point. Another study [15] assessed the DBSCAN technique with a similar clustered data. The execution time for a two-dimensional dataset having nearly 12,000 data-points with 20 clusters was 21 s. The value for T in this scenario is 1.75 ms per data-point. For CATS, the value of T for DBSCAN, with an average step I time of 56.21 s for 3000 data-points, is 18.73 ms per data-point. Considering the differences in the hardware platforms used and the programming approaches adopted (language, code length, program flow, etc.), it could be asserted that CATS shows an adequate performance for a novel approach.

5 Conclusion

In a smart manufacturing environment, prediction and detection of anomalies in real time is a critical problem to increase productivity. Any anomaly is typically to be detected within a time span of couple of seconds rather than taking several minutes. A common approach is to collect huge data-points and apply prediction mechanisms. The accuracy of existing outlier detection algorithms is prone to decrease with an increase in outlier population, limiting the real-time functionality in a smart factory environment. In this regard, this paper proposes clustering-aided two-step (CATS) approach for outlier detection from the received data-points. The first step involves use of DB-SCAN, while the second step uses LOF. CATS is able to detect outliers with accuracy as high as 98.58%. Two unsupervised techniques are exploited in two steps meant for segmentation of the data and refined search of outliers. The novelty in CATS lies in detection of outliers in both steps and implementation of second step only on the remaining, undetected outliers from first step. Importantly, the results indicate CATS' ability to detect outliers with an extremely high accuracy, say, 97.5% even with an outlier of upto 16%. Significantly, even when the outlier is increased to an abnormally high value of 25%, the accuracy achieved is still greater than 80%. Further, significantly, the time taken for CATS varies between 60 and 88 s, irrespective of the outliers. This is a significant result. It shows that the proposed technique—CATS—is an efficient mechanism that can be used for applications where there is a stringent need to accurately detect the anomalies within few seconds. Hence, CATS could be potentially deployed for real-time anomaly detection in a smart manufacturing environment. The next step in this work is to further improve the technique of CATS; so that the detection accuracy is always greater than 95% even when the outliers increase beyond 18 or 20%.

Acknowledgements The authors acknowledge UK-India Education and Research Initiative (UKIERI) and Department of Science and Technology (DST), Govt. of India, for their kind support.

References

1. Lopez, F., Saez, M., Shao, Y., Balta, E., Moyne, J., Mao, Z., Barton, K., Tilbury, D.: Categorization of anomalies in smart manufacturing systems to support the selection of detection mechanisms. IEEE Robot. Autom. Lett. **2**, 1885–1892 (2017)
2. Zhang, J.: Advancements of outlier detection: a survey. ICST Trans. Scalable Inf. Syst. **13**(1), 1–26 (2013)
3. Kotu, V., Deshpande, B.: Data Science Concepts and Practice, 2nd edn, pp. 447–466. Morgan Kaufmann publications, Cambridge (2019)
4. Aissa, N., Guerroumi, M.: Semi-supervised statistical approach for network anomaly detection. Procedia Comput. Sci. **83**, 1090–1095 (2016)
5. Hu, M., Ji, Z., Guo, Y., Yan, K., Hu, H.: A two-stage temporal anomaly detection algorithm based on danger theory: applications and techniques in cyber security and intelligence. In: International Conference on Applications and Techniques in Cyber Security and Intelligence ATCI, pp. 485–494 (2018)

6. Yu X., Tang L.A., Han J.: Filtering and refinement: a two-stage approach for efficient and effective anomaly detection. In: 9th IEEE International Conference on Data Mining, pp. 617–626 (2009)
7. Huang, G., Zhang, Z., Yang, W.: Outlier detection method based on improved two-step clustering algorithm and synthetic hypothesis testing. In: IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC) 2019, pp. 915–919 (2019)
8. Gupta, M., Gao, J., Sun, Y., Han, J.: Community trend outlier detection using soft temporal pattern mining. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 23 Sep 2012, pp. 692–708. Springer, Berlin, Heidelberg (2012)
9. Leng, M., Chen, X., Li, L.: Variable length methods for detecting anomaly patterns in time series. In: 2008 International Symposium on Computational Intelligence and Design, Vol. 2, pp. 52–55. IEEE (2008)
10. Nowak-Brzezińska, A., Xieski, T.: Outlier mining using the DBSCAN algorithm. J. Appl. Comput. Sci. **25**(2), 53–68 (2017)
11. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander J.: LOF: identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD international conference on Management of data, pp. 93–104 (2000)
12. Ertöz, L., Steinbach, M., Kumar, V.: Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In: Proceedings of the 2003 SIAM International Conference on Data Mining, pp. 47–58. Society for Industrial and Applied Mathematics (2003)
13. Yang, M., Ergu, D.: Anomaly detection of vehicle data based on LOF algorithm. Front. Sig. Process. **4**(1), 43–49 (2020)
14. Pirrone, R., Cannella, V., Monteleone, S., Giordano, G.: Linear density-based clustering with a discrete density model (2018). arXiv preprint arXiv:1807.08158
15. Ansari, Z., Azeem, M., Babu, A., Ahmed, W., Babu, A.V.: Quantitative evaluation of performance and validity indices for clustering the web navigational sessions. World Comput. Sci. Inf. Technol. J. (WCSIT) **1**(25), 217–226 (2011)

Prediction of Energy Consumption Using Statistical and Machine Learning Methods and Analyzing the Significance of Climate and Holidays in the Demand Prediction

Naveen Tata, Srivasthasva Srinivas Machiraju, V. Akshay, Divyasree Mohan Menon, N. B. Sai Shibu, and D. Arjun

Abstract With the increase in the development of smart metering in energy systems, a large amount of data is being generated. The data consists of energy generated, energy consumed and energy stored with respect to time. This data can be used to improve the efficiency, reliability and stability of the power system by using machine learning algorithms. Energy requirement of each consumer can be predicted with the available data. Renewable energy generation can also be predicted. In this paper, different statistical and machine learning models are used to analyze the energy usage in smart communities. To validate the prediction models, smart meter data from our campus is used. The results show that the long short-term memory (LSTM) model is more suitable for energy demand prediction. The LSTM model is then used to predict the energy demand in students' hostels during conditions such as climate and holidays.

Keywords Smart grid · Micro grid · Machine learning · Demand prediction · Energy usage analysis

1 Introduction

Power system consists of three main stages, generation, transmission and distribution. In generation, electricity is generated using generators. Thermal, hydro and nuclear generators are commonly used generators. Usually, these generating stations are far from the residential areas. The generated power is stepped up to higher voltages

N. Tata · S. S. Machiraju · D. Arjun
Department of Computer Science and Engineering (CSE), Center for Wireless Networks and Applications (WNA), Amrita Vishwa Vidyapeetham, Amritapuri, India

V. Akshay · D. M. Menon · N. B. Sai Shibu (✉)
Center for Wireless Networks and Applications (WNA), Amrita Vishwa Vidyapeetham, Amritapuri, India
e-mail: saishibunb@am.amrita.edu

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_10

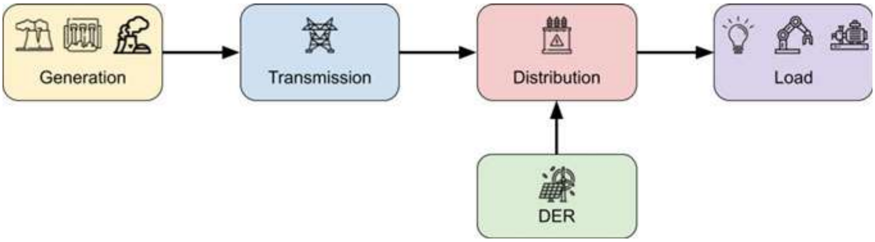


Fig. 1 Single line diagram of the power system

and then transmitted to the residential areas through transmission lines. The power is received at a distribution substation that steps down the power and distributes to houses and factories through distribution transformers. These transformers are placed closed to the residential or industry load which further steps down the power to meet the requirements of the loads. With the growing demand of power, distributed energy resources (DER) are set up closer to the loads. Figure 1 shows the single line diagram of the power system [1].

The DERs are power generators that use renewable energy such as solar photo voltaic, wind and hydro. As they are placed close to the loads, the loss in transmission and distribution is very minimal, and efficiency is also high [2]. DERs also provide real-time data on generation and consumption. By using machine learning algorithms, the power demand can be predicted and plan the load usage based on the predicted energy. This will help the industry to increase their productivity.

One of the main differences between humans and computers is that humans learn from past experience, whereas computers need to be told what to do, i.e., they need to be programmed explicitly. To avoid this contrast, there comes the technique of machine learning. It is a method to teach computers to learn and act like humans do and upgrade their intellect over time in an autonomous fashion by providing them with data and information through real-world interactions [3]. It is used mainly for predictions, suggestions and assistants in social media services, online fraud detection and transportation networks. There are significant applications of machine learning in the field of energy namely time series forecasting, energy system control and fault detection in smart grid. The problems that we address in this paper are as follows:

- To identity the parameters that contribute to the energy demand prediction and remove the redundant parameters from the data set.
- By analyzing the power consumption data from each building, we predict the future power demand of each building and respond to periods of peak demand.
- Understand the significance of holidays and climate in energy demand.

This paper aims to identify a machine learning model that is best suitable for demand prediction application in smart grid. We considered Holt's Linear, Holts Winter, ARIMA and LSTM neural network in this paper. These models are evaluated in Google Colab environments with the data collected from the smart meters deployed in our university campus. Root mean square error of these models were evaluated, and the model with least error is considered best. The paper is organized as follows. Section 2 briefly describes the state of the art and other similar works performed at various institutions. Section 3 explains the process in data collection and processing. Section 4 is about implementation, and the results of each models are discussed. The paper is concluded in Sect. 5.

2 Related Work

The work described in Paper [4] proposes a multiple regression model and an artificial neural network model which is intended to set up a relationship between varying weather conditions and energy consumption. Based on the open values such as wind speed, temperature, humidity, air pressure, and relative atmospheric humidity, the dependency on electricity consumption is being studied from the years 2011 to 2016. The chosen models were assessed based on the coefficient of determination, i.e., *R*-value. According to the results presented, the ANN model best fits the test data.

Deepika and Prakash predicted power consumption in cloud data centers [5] using various machine learning algorithms such as regression types, random forest, forecasting methods, KNN, and multi-layer perceptron. They evaluated Microsoft Azure Virtual machine using these models. Data on CPU utilization, virtual memory read/write, I/O operations is captured for each of these models. Based on their test results, multi-layer perceptron proved to be the best model for this application. The authors in [6] predicted the power consumption for the upcoming day using support vector machine (SVM) and recurrent neural network (RNN) based on historical power consumption which classifies a day into high, low, and moderate consumption periods in a year and daytime temperatures.

In [7], prediction models such as Markov chains, hybrid prediction and stride prediction have been applied to forecast the future electric power values based on the previous values. With a mean absolute error value of 34W, it has been concluded that Markov chains anticipate the lowest error value as compared to other prediction models, hence it can be used to build up an intelligent energy management system.

Wang et al., in [8] proposes power consumption forecast model using LSTM, which represents the power consumption series as well as dependencies about weather and holiday information. This model has increased accuracy rate by 22% when compared with ARIMA, another representative algorithm commonly used for forecasting signals. In (Prediction of Electricity Consumption for Residential Houses in New Zealand), authors used artificial neural network-based model to predict the power usage 24 h before time using ANN-based NARX (nonlinear autoregressive network with exogenous inputs) predictive model by considering temperature, barometric pressure, relative humidity, wind speed, and wind direction.

3 Data Collection and Preprocessing

Smart meters are deployed in our university buildings that provide data on energy consumption at each building. The smart meter has an internal memory which is used to store the data for a limited time. Each parameter is saved to a unique memory address. The data captured by the smart meter is collected using a Raspberry Pi through Modbus protocol. The Raspberry Pi requests the smart meter by mentioning the address specified for each parameter. The smart meter replies back with the requested data. The collected data is sent to our university server over WiFi by using MQTT [9]. The server also captures weather data from Darksky using the provided API [10]. The server receives the data and stores it to MySQL database. APIs are provided by MySQL using with the data can be shared to third party applications. Google Colab is used to evaluate the machine learning models. The Colab environment gets data from the server using the MySQL APIs. The obtained data is in the form of comma separated values (CSV). Figure 2 shows the setup made for data collection and also shows how data is imported to the Colab environment.

3.1 Data Preprocessing

The data that was collected in the previous steps is as follows: The whole dataset consists of four months of data. The 21 columns of each data tuple comprise the following attributes: The timestamp, electricity consumption in kilowatt per hour, holidays, and weather type. The timestamp describes the date and hourly details of the collections. For example, the data collected on April 27, 2019, at 4 pm looks as follows 19-04-27 16:00:00. Holidays describe whether a particular day is a working day or a holiday. The weather type attribute contains 18 types of weather conditions which include humid and mostly cloudy, humid and partly cloudy, humid and overcast, drizzle, light rain, rain, breezy and humid, humid, rain and breezy, heavy rain, possible light rain, and humid, heavy rain and humid, possible drizzle and humid, rain and humid, light rain and humid, rain and windy, possible light rain, overcast. The data which was collected for every 15 s time interval using smart meters is resampled with mean into hourly data by taking the average values of all recordings in a particular hour. There are certain situations where we faced issues of having missing data due to meter errors in smart meters. To solve this problem, we used a data imputation technique called last observation carried forward (LOCF) to interpolate the missing data.

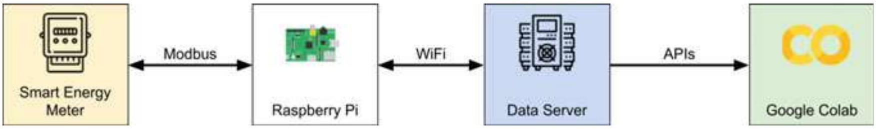


Fig. 2 Data collection setup

As the weather type attribute contains categorical features, we used one hot encoding technique to get one hot encoded vector to each of the corresponding categorical values. The electricity consumption attribute has values varying in different ranges, and the rest of the attributes have a discrete set of values. The training process is not satisfactory if the original measurement units are used, so we used MinMaxScaler as shown in Eq. 1 to normalize the dataset. x_i represents each value in the dataset, whereas $min(x)$ is the minimum value and $max(x)$ is the maximum value in the dataset. The training performance was good when the model was provided with normalized data.

$$MinMaxScaler = \frac{x_i - min(x)}{max(x) - min(x)} \tag{1}$$

4 Implementation and Results Discussion

A total of four different models were implemented which include Holt's Linear [11], Holts Winter [12], ARIMA [13], and LSTM neural network [14]. The former three models are statistical models, while the latter one is a subcategory in the recurrent neural network. The data was split into two different categories as train and test data. The initial 70% corresponds to training data (represented as blue line), and the rest belongs to the test data (represented as orange line) as shown in Fig. 3.

Initially, the Holt's model is provided with the training data, and the results on the test data are compared with the original values. The graph in Fig. 4 shows the

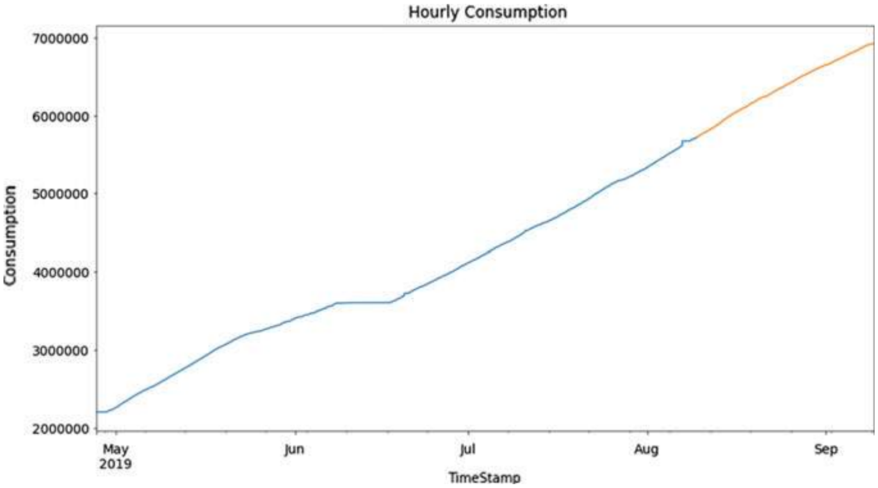


Fig. 3 Visualization of train (blue line) and test data (orange line)

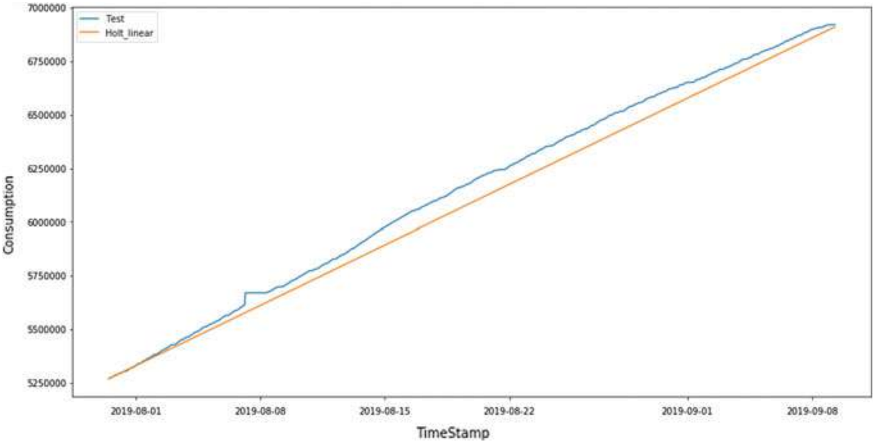


Fig. 4 Prediction on test data using Holt's Linear Model

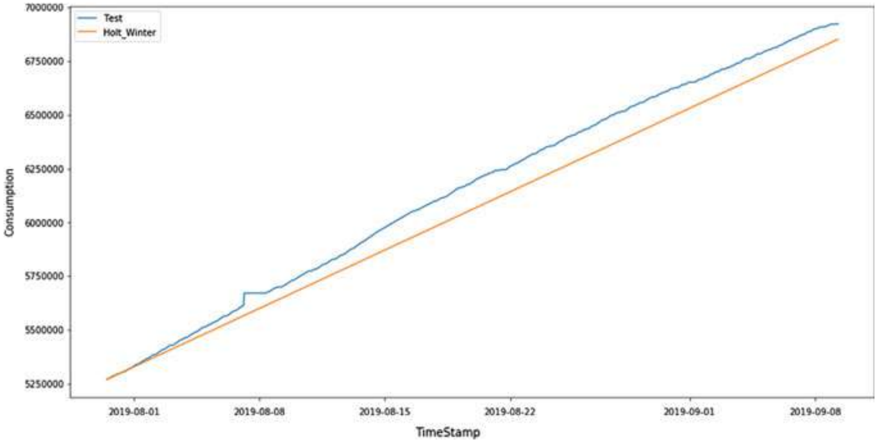


Fig. 5 Prediction on test data using Holt's Winter Model

deviation in the values predicted by the model as of the original values of the test data. The same process is repeated for both Smoothing, ARIMA, and results are plotted in Figs. 5 and 6.

Advancement in deep learning leads to the development of various models to solve a wide variety of real-world problems. RNN is one of the neural network models which can be used for time series data. The reason for the selection of RNN for time series data is because these models will consider the data in previous time steps and input features of present time steps to predict the output. But RNNs suffer from vanishing gradient descent problems. Due to these, weights of some of the input features might not be updated in the long run. To resolve this issue, long short-term memories (LSTM) was implemented.

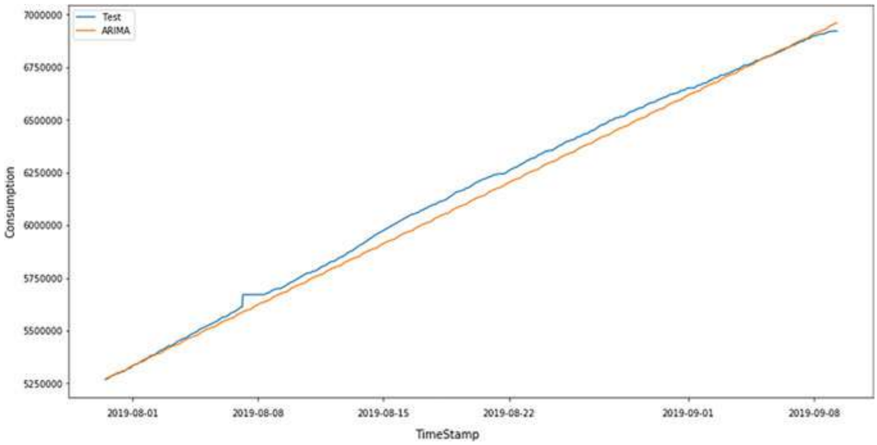


Fig. 6 Prediction on test data using ARIMA Model

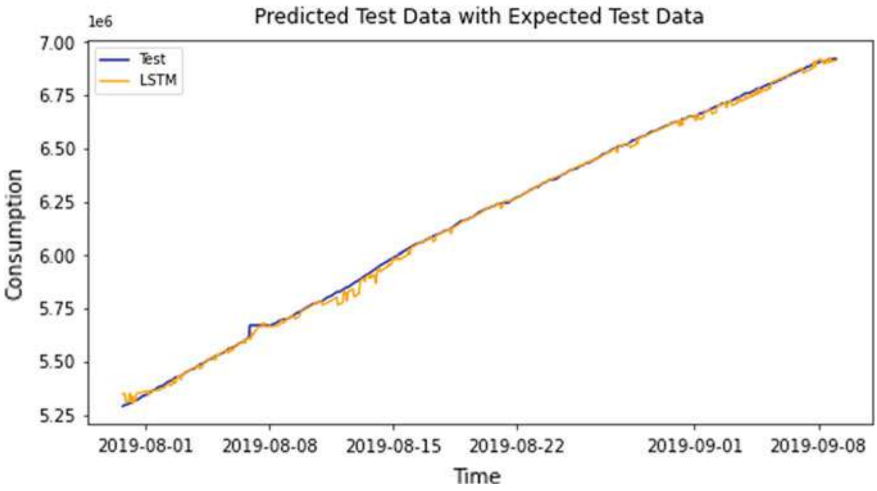


Fig. 7 Prediction on test data without weather and holidays feature using LSTM

A simple three-layer LSTM model was implemented, and the data was modified before passing it to the LSTM model. Two types of LSTM models were considered. The first one only considers the previous 12 h of power consumption as inputs. Figure 7 compares the results of predicted values of the model with the original values. For the second model, the input features such as weather conditions and holidays are also considered along with the power consumption values of the previous 12 h. The training data was passed on to the model to catch the trend in the data. The predictions on the testing data are compared with the true values. Figure 8 shows the deviation of predicted values from the original values.

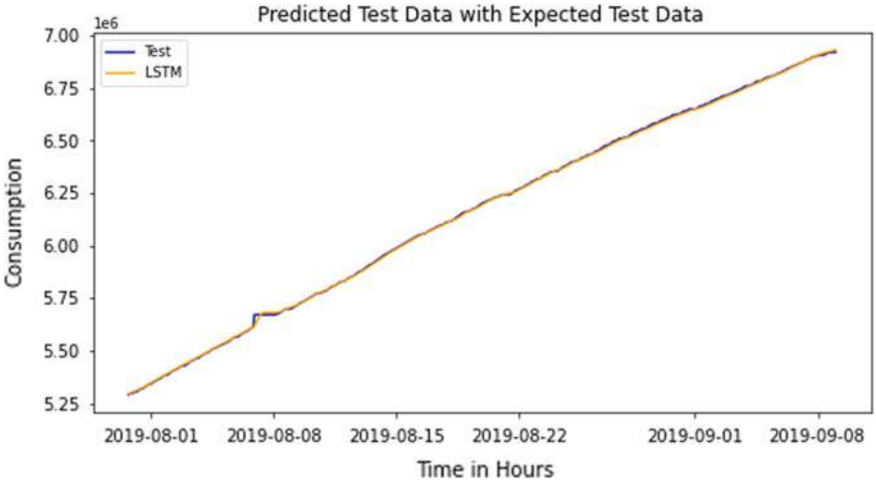


Fig. 8 Prediction on test data with weather and holidays features using LSTM

Table 1 Comparison of machine learning models

Method	RMSE
Holts Linear	68.37
Holts Winter	100.70
ARIMA	45.91
LSTM without weather and holiday	16.28
LSTM with weather and holiday attributes	13.30

4.1 Comparison of all the considered models

The predicted values along with their true values of all the four models are shown in the above plots. Root mean square error (RMSE) metrics value is used for the comparison of these models, and Table 1 shows the RMSE values of all the four models that are implemented. The LSTM model which was trained with weather and holiday attributes produced the minimum RMSE score.

5 Conclusion and Future Work

The data collected and used in this research is from our university, and to avoid any possible outliers in the data collected, we compared the data from the energy department that takes an average of 12 h. The results obtained by considering 12 h are much better than the other values. In this paper, different statistical and machine learning algorithms like Holt's Linear, Holts Winter, ARIMA, and LSTM neural network with and without weather and holiday attributes are implemented. Graphs are plotted in order to compare the predicted patterns with the original results. The models are evaluated with their RMSE scores. The results show that the LSTM model which considers the above-mentioned attributes produces the best results.

The data used in this survey gave much better results with a simple LSTM model, and usage of very complex models results in the over fitting in the data which might lead to the decrease in the efficiency of the system. This reason made us go with the simple LSTM model. This paper also compares the simple version of LSTM that gave better results than many of the statistical models. In future, we would like to work on predicting the demand of other essential utilities such as water and cooking gas in our campus and also extend this finding to deploy energy trade concepts.

Acknowledgements We express our deep gratitude to our beloved Chancellor, staff, and faculty members of the department for the inspiration and motivation. This work was partly done under the Project "Smart Services and Optimization for Microgrids (SSOM)" in the scheme of Project-based Personnel Exchange Program with Indo-German (DST-DAAD) Joint Research Collaboration.

References

1. Kothari, D., Nagrath, I.: Power System Engineering, 3rd edn. McGraw-Hill Education (2019)
2. Shyam, B., Kanakasabapathy, P.: Renewable energy utilization in India policies, opportunities and challenges. In: 2017 International Conference on Technological Advancements in Power and Energy (TAP Energy), pp. 1–6 (2017). https://doi.org/10.1109/TAPENERGY.2017.8397311
3. Alpaydin, E.: Introduction to Machine Learning. MIT Press (2020)
4. Prabakar, A., Wu, L., Zwanepol, L., Velzen, N., Djairam, D.: Applying machine learning to study the relationship between electricity consumption and weather variables using open data (2018). https://doi.org/10.1109/ISGTEurope.2018.8571430
5. Deepika, T., Prakash, P.: Power consumption prediction in cloud data center using machine learning. Int. J. Electr. Comput. Eng. (IJECE) **10**(2), 1524–1532 (2020). https://doi.org/10.11591/ijece.v10i2
6. Theile, P., Towle, A.-L., Karnataki, K., Crosara, A., Paridari, K., Turk, G., Nordstrom, L.: Day-ahead electricity consumption prediction of a population of households: analyzing different machine learning techniques based on real data from rte in France, pp. 1–6 (2018). https://doi.org/10.1109/SmartGridComm.2018.8587591
7. Gellert, A., Florea, A., Fiore, U., Palmieri, F., Zanetti, P.: A study on forecasting electricity production and consumption in smart cities and factories. Int. J. Inf. Manage. **49**, 546–556 (2019). https://doi.org/10.1016/j.ijinfomgt.2019.01.006
8. Wang, X., Zhao, T., Liu, H., He, R.: Power consumption predicting and anomaly detection based on long short-term memory neural network. In: 2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), pp. 487–491 (2019)
9. Sai Shibu, N.B., Hanumanthiah, A., Sai Rohith, S., Yaswanth, C., Hemanth Krishna, P., Pavan, J.V.S.: Development of IotTenabled smart energy meter with remote load management. In: 2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pp. 1–4 (2018)

10. Aziz, A.: Coastal alerting IoT system in response to high tides and turbulent weather. In: 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1–7 (2019)
11. Swamidass, P.M. (Ed.): Forecasting models, Holt's forecasting model, pp. 274–274. Springer US, Boston, MA (2000). https://doi.org/10.1007/1-4020-0612-8-409
12. Gamal, M.D.: Holt-winters forecasting method that takes into account the effect of Eid. Sci. J. Appl. Math. Stat. **3**, 257–262 (2015). https://doi.org/10.11648/j.sjams.20150306.15
13. Liu, T., Liu, S., Shi, L.: ARIMA Modelling and Forecasting, pp. 61–85 (2020). https://doi.org/10.1007/978-981-15-0321-4-4
14. Wang, L.-N., Zhong, G., Yan, S., Dong, J., Huang, K.: Enhanced LSTM with Batch Normalization (2019). https://doi.org/10.1007/978-3-030-36708-4_61

Ranking of Educational Institutions Based on User Priorities Using AHP-PROMETHEE Approach

A. U. Angitha and M. Supriya

Abstract Today, education plays a major role in bringing values to the coming generations. With right to education taken up seriously by the government, there is a vast increase in the number of students taking up various courses and degrees. To cater to these needs, several educational institutions have also come up with the intention of providing a variety of courses to the students. However, as the numbers increase, the need to assess these institutions also increases in order to help students decide the institution of their choice and also for the institutions to better themselves. To accomplish evaluation of such institutions, a number of ranking systems were established with a fixed set of criteria. However, a ranking system which would cater to the needs of a student individually was never developed. This paper aims to develop a ranking system, which ranks the educational institutions based on the needs of an individual, using analytical hierarchy process and the multi-criteria decision-making method PROMETHEE. In order to make this system automized, the data required for this process has been retrieved using the process of web crawling. Web crawlers are automated scripts that are used to browse the World Wide Web in a systematic manner. This work will be useful for parents and students to find institutions based on their set of preferences.

Keywords Web crawling · Multi-criteria decision-making · PROMETHEE · AHP

1 Introduction

The education industry has grown nominally in recent years, so does the number of students. Today, there are a huge number of educational institutions in our country. More and more students prefer higher education and are going abroad to take up various courses. To cater to the need of Indian students, Government of India is focused on getting universal recognition for its education system. Many colleges are

A. U. Angitha (✉) · M. Supriya
Department of Computer Science and Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Bengaluru, India
e-mail: angitha129@gmail.com

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_11

Table 1 Various ranking schemes available in India

National Institutional Ranking Framework (NIRF)	Quacquarelli Symonds (QS)	Times Higher Education (THE)	National Assessment and Accreditation Council (NAAC)
Teaching, learning and resources (TLR)	Academic reputation	Teaching (the learning environment)	Curricular aspects
Research and professional practice (RP)	Employer reputation	Research (volume, income, and reputation)	Teaching learning and evaluation
Graduation outcomes (GO)	Faculty/student ratio	Citations (research influence)	Research, innovations and extension
Outreach and inclusivity (OI)	Citations per faculty	International outlook (staff, students, and research)	Infrastructure and learning resources
Peer perception	International faculty ratio	Industry income (knowledge transfer)	Student support and progression
			Governance, leadership and management
			Institutional values and best practices

trying to get NBA (National Board of Accreditation) accreditation. Due to this, a number of ranking schemes have been introduced to ensure education standards in India. Some of the ranking systems followed for educational institutions along with the parameters they emphasize on are presented in Table 1.

The present ranking systems rank the institutions based on a fixed set of parameters. The weightage assigned to these parameters is also pre-defined. These parameters were decided based on a lot of research, intended to help users know the best and the worst among the huge list of institutions currently present. Since the ranking process involves several criteria, multi-criteria decision-making (MCDM) was introduced. A few popular MCDM techniques that exist are PROMETHEE, TOPSIS, OMDM, MAUT, AHP etc.

This paper's focus is on ranking educational institutions based on user preferences. This work makes use of the parameters and data from National Institutional Ranking Framework (NIRF) and its reports. The criteria are handled using MCDM techniques AHP and PROMETHEE. There also exist sub-criteria under each criterion such that PROMETHEE is carried out at the sub-criteria level, the result of which is used to perform PROMETHEE at the higher level. The criteria and sub-criteria considered are given in Table 2.

The existing work and the problem description are discussed in Sects. 2 and 3. The implementation and the performance details are mentioned in the subsequent section. The paper is concluded in Sect. 5.

Table 2 List of criteria and sub-criteria considered

Teaching, learning and resources(A)	Research and professional practice (B)	Graduation outcomes (C)	Outreach and inclusivity (D)	Placements and higher studies (E)
Student strength	Metric for publications	Metric for university examinations	Percentage of women (women diversity)	No. of students placed through placements
Faculty–student ratio	Metric for quality of publications	Metric for number of Ph.D. students graduated	Economically and socially challenged students	No. of students selected for higher studies
Faculty with Ph.D. ratio	Patents published and granted			

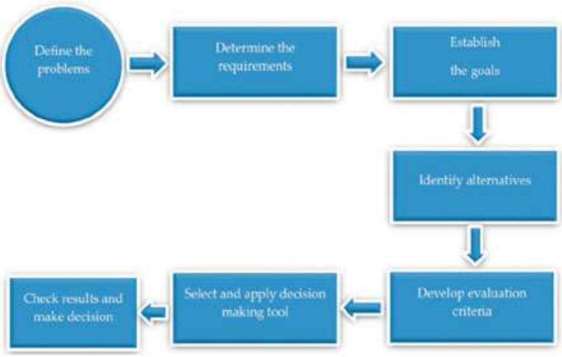
2 Literature Survey

MCDM is a field of Operations Research that is used when a number of criteria are involved. It aims to help choose the best alternative among a set of alternatives based on a set of number of criteria or attributes. Figure 1 shows the activity flow beginning with defining the problem to finding the optimal solution.

MCDM has found its application in various sectors. For instance, it was used to develop an application that would help librarians to rank materials based on PROMETHEE [1]. It has also been used to evaluate the scholarly performance of students based on several criteria such as biographical details, their grades etc, in [2].

PROMETHEE is one of the most preferred MCDM technique. PROMETHEE was further developed into PROMETHEE II which handled situations where the number of attributes that are considered is huge. This process is well explained in [3], and it was used in a digital library management system to rank the collection areas intended for faster access for users.

Fig. 1 Decision-making process


Define the problems
Determine the requirements
Establish the goals
Identify alternatives
Develop evaluation criteria
Select and apply decision making tool
Check results and make decision

This work focuses on PROMETHEE since the ranking involves multiple criteria. For the criteria to be evaluated, we need to assign weights to them. This is accomplished using a systematic approach called analytical hierarchy process (AHP). AHP works using pairwise comparison of the criteria based on mathematics and psychology [4]. The general structure of AHP is depicted in Fig. 2. In this paper, the selection of the alternative is corresponding to the selection of the educational institution and the various parameters are the criteria. Each criterion will have a set of sub criteria under them. Each criterion, constituted as a vector, is multiplied by its weight and results in the score of the alternative with respect to the criterion.

Web crawlers have an interesting history. Crawlers began with collecting statistics about the web. Later, various types of crawlers were introduced for numerous applications. A study of various crawlers and their architectures can be found in [5]. Architecture of a basic crawler is shown in Fig. 3. The architecture was further enhanced in [6]. As crawlers got smarter, their applications also increased across different fields. Various crawlers were used in different scenarios as in [5, 7, 8]. For instance, analyzing customer behavior was performed through opinion mining in [9, 10]. In [9], a parallel crawler is implemented to eliminate the possibility of

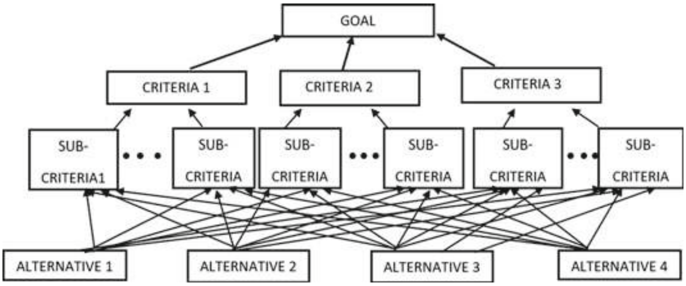
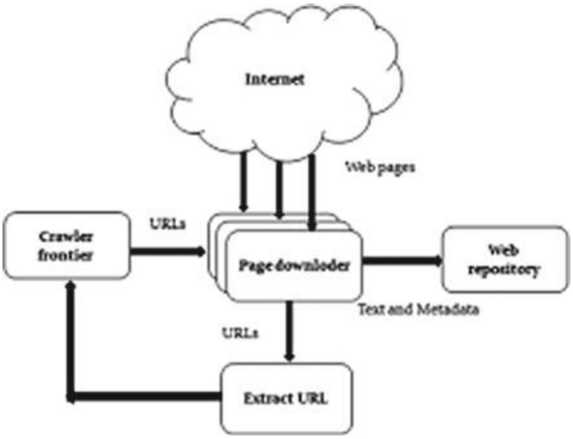


Fig. 2 Structure of AHP

Fig. 3 Crawler architecture



single-point failure by using distributed control pattern. Crawlers in the educational sector are yet to be exploited much. This project focuses on crawlers in educational field. In [10], effective information was extracted through the reviews on the product, whereas in [11], the purchase intention of the user was predicted using the online activity data. The performance of the crawlers can be increased by running crawlers parallelly using multi-threading to make the process of data extraction faster.

Internet service providers were also ranked according to the user preferences using AHP and technique for order of preference by similarity to ideal solution (TOPSIS) methods [12]. A hierarchical trust model was proposed in 2015 to rank service providers along with numerous plans for their infrastructure [13]. A comparative study between AHP-based and fuzzy-based MCDM techniques was conducted to rank the cloud computing services in which fuzzy AHP seemed to give a clearer set of values compared to the AHP mechanism [14]. In the subsequent year, another combination of analytical and fuzzy MCDM methods was used to rank cloud service providers based on trust [15] which was further taken forward in [16] to select the most trustworthy ones.

3 Problem Description

Educational institutions play a major role in transforming the values to the upcoming generations. In recent times, there is a steep increase in the number of students opting for various degrees and one can also observe that the number of institutions offering these degrees is on the rise to cater the need of the younger generations. With the major growth of education sector, it is very essential to evaluate the educational institutions for their own betterment as well as to serve as a reference for choosing a institution.

There exist a number of ranking systems that rank educational institutions based on a fixed set of parameters. However, there exists no such system that provides the rank of institutions based on user defined preferences. This project aims to implement web crawling which extracts data regarding the educational institutions from the web and build a ranking system based on analytical hierarchical process and multi-criteria decision-making method PROMETHEE and ranks educational institutions based on various parameters weighed according to user preferences. It also considers sub-criteria called attributes under these criteria to provide a greater insight. Such a ranking system helps the students and parents to filter out the institutions based on their requirements and find a suitable institution to cater their need.

Ranking of the institutions is based on certain parameters which are of great priority to the users. The association between the criteria, alternatives and the ultimate result is depicted in Fig. 4 at distinct levels of AHP. Figure shows the criteria and sub-criteria enlisted in Table 2. The alternatives correspond to the colleges being considered. The colleges considered for the work are among the top colleges and universities in India. Henceforth, in this work, the colleges considered are renamed as College 1, College 2, etc. The attributes are named Criteria 1, Criteria 2, etc., and the

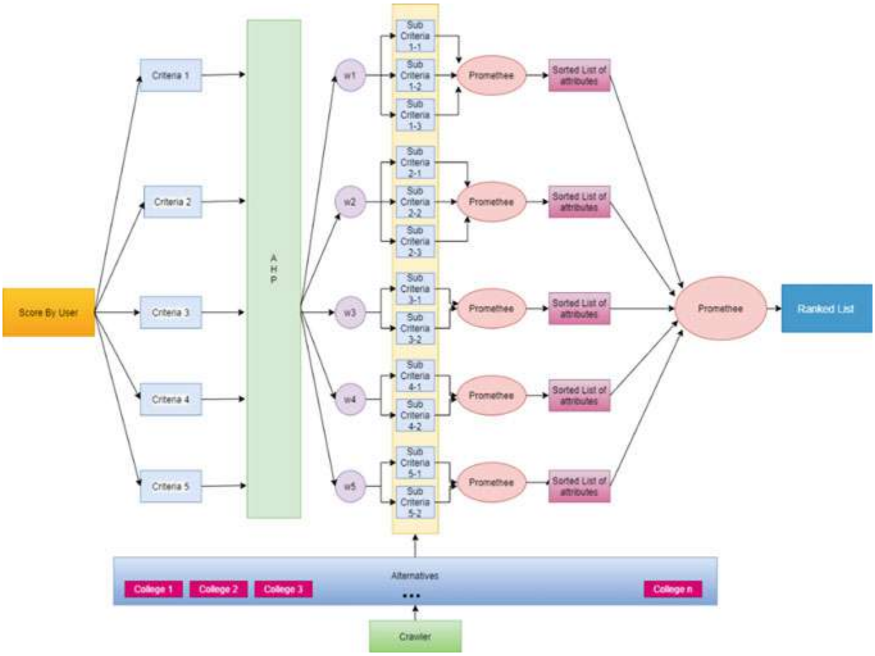


Fig. 4 Working of the proposed system

sub-attributes are named Sub-Criteria1-1, Sub-Criteria1-2, etc., each corresponding to their respective attributes.

The data regarding the colleges is retrieved using web crawler which crawls the Internet and provides data about the educational institutions. The values corresponding to each attribute are taken from the NIRF report submitted by the colleges every year to the NIRF committee which is generally accessible from the NIRF website. These reports hold the data that will be considered for calculations while ranking the institutions. The data is downloaded in a pdf format and converted into an excel format for better parsing, and the required data is collected into a single excel sheet after which the downloaded data maybe deleted or cached as per use.

The user assigns scores to the criteria according to his/her preference. These scores are taken up by the AHP algorithm which processes the scores and using the concept of relative importance, assigns weights to the criteria. The advantage of AHP is that it compares the conflicting criteria and gives a weightage considering their relative importance. The weights assigned to each criterion is distributed among its sub-criteria. These weights can be uniformly distributed among the sub-criteria or they can be distributed based on their importance or any weightage that might be assigned to them explicitly. In this project, we assign an importance value to the sub-criteria, according to which the weights are distributed among them.

Once the weights are assigned to the sub-criteria, PROMETHEE is carried out in a hierarchical fashion where each criterion implements the PROMETHEE algorithm among its sub-criteria. This helps to reduce the complexity as the complexity of the algorithm is directly proportional to the number to criteria being considered. This results in "*n*" number of sorted list of colleges where *n* is equal to the number of criteria. Each attribute, i.e., each college, will have a value associated with it as the output to the hierarchical PROMETHEE performed at the sub-criteria level. These values are taken up by the PROMETHEE algorithm, at the criteria level, as input. On performing PROMETHEE on this data, the sorted list of colleges is obtained which ultimately gives us the ranked list of institutions.

4 Results and Discussions

4.1 Crawler

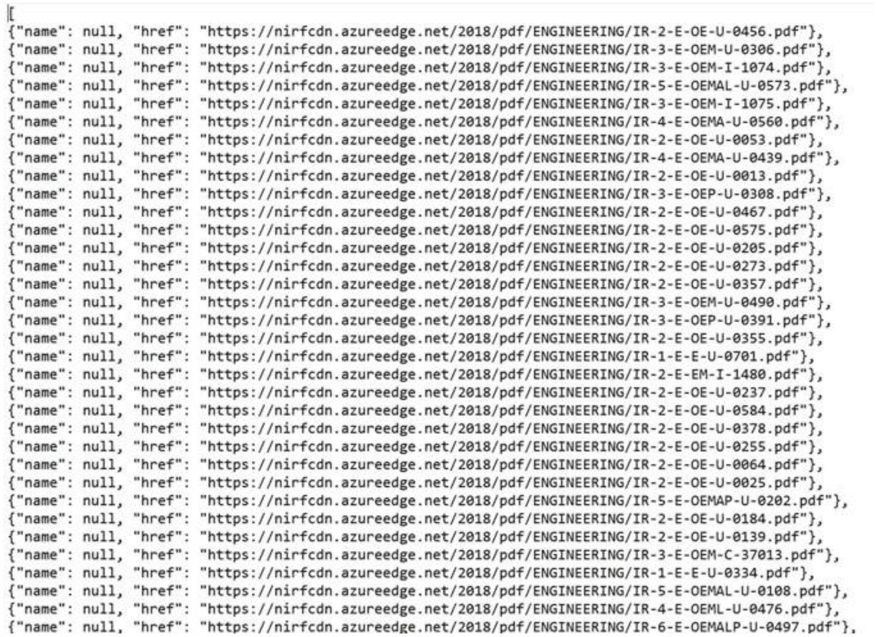
Crawling is carried out by providing a root URL which is the parent URL for the required data. From this page, the crawler searches across the entire page to find the required links. If the link follows a specific pattern, it can be given as an input to make the crawling process faster. The URLs crawled are stored in a json format for easier access. Once the URLs are stored in a json file, these URLs are downloaded to the local folder and the required values can be extracted. Once the values are extracted, the folders may be deleted if not further required. The json file data is shown is Fig. 5. The download process is named according to the session name to ensure its uniqueness. The downloaded data is in the form of pdf document which can be processed and converted to the required form.

4.2 AHP-Based Weight Derivation

Initially, every attribute is assigned a weight by the user based on their preference. The weights are derived from the scores assigned to the attributes. The scores generally range between 1 and 9, 9 being the most preferred attribute. A test case of score assignment is listed in Table 3. AHP performs a pairwise comparison of the scores for each attribute as represented in Table 4. The matrix thus formed in Table 4 is squared in iterations till it matches a threshold as shown in Table 5. The next step is to calculate the row sum of the matrix in Table 5 to normalize the data. These normalized values denote the weights to the attributes corresponding to the preferences given by the user in Table 3.

The weights assigned to the attributes by AHP to the identified criteria are as follows: 0.26666667, 0.13333333, 0.2, 0.23333333, 0.16666667.

With reference to the internal weightage and the weights assigned to the sub-criteria based on user preference is given in Table 6. However, this can be changed as per the user preference.



```
[
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-2-E-OE-U-0456.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-3-E-OEM-U-0306.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-3-E-OEM-I-1074.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-5-E-OEMAL-U-0573.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-3-E-OEM-I-1075.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-4-E-OEMA-U-0560.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-2-E-OE-U-0053.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-4-E-OEMA-U-0439.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-2-E-OE-U-0013.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-3-E-OEP-U-0308.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-2-E-OE-U-0467.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-2-E-OE-U-0575.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-2-E-OE-U-0205.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-2-E-OE-U-0273.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-2-E-OE-U-0357.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-3-E-OEM-U-0490.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-3-E-OEP-U-0391.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-2-E-OE-U-0355.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-1-E-E-U-0701.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-2-E-EM-I-1480.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-2-E-OE-U-0237.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-2-E-OE-U-0584.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-2-E-OE-U-0378.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-2-E-OE-U-0255.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-2-E-OE-U-0064.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-2-E-OE-U-0025.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-5-E-OEMAP-U-0202.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-2-E-OE-U-0184.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-2-E-OE-U-0139.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-3-E-OEM-C-37013.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-1-E-E-U-0334.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-5-E-OEMAL-U-0108.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-4-E-OEML-U-0476.pdf"},
{"name": null, "href": "https://nirfcdn.azureedge.net/2018/pdf/ENGINEERING/IR-6-E-OEMALP-U-0497.pdf"},
```

Fig. 5 List of URLs crawled and stored in json

Table 3 Scores assigned to the attributes

Teaching, learning and resources (C1)	8
Research and professional practice (C2)	4
Graduation outcomes (C3)	6
Outreach and inclusivity (C4)	7
Placements and higher studies (C5)	5

Table 4 Pairwise comparison of the scores available for the five attributes

	C1	C2	C3	C4	C5
C1	1	2.	1.33333333	1.14285714	1.6
C2	0.5	1	0.66666667	0.57142857	0.8
C3	0.75	1.5	1	0.85714286	1.2
C4	0.875	1.75	0.66666667	1	1.4
C5	0.625	1.25	0.83333333	0.71428571	1

Table 5 Squared matrix formed after two iterations

	C1	C2	C3	C4	C5
C1	125.	250.	166.66666667	142.85714286	200.
C2	62.5	125.	83.33333333	71.42857143	100.
C3	93.75	187.5	125.	107.14285714	150.
C4	109.375	218.75	145.83333333	125.	175.
C5	78.125	156.25	104.16666667	89.28571429	125.

Table 6 Weightage to sub-criteria

Teaching, learning and resources (A)	Predefined	AHP assigned
Student strength	0.3	0.08
Faculty–student ratio	0.3	0.08
Faculty with Ph.D. ratio	0.4	0.10666667
Research and professional practice (B)		
Metric for publications	0.4	0.05333333
Metric for quality of publications	0.4	0.05333333
Patents published and granted	0.2	0.02666667
Graduation outcomes (C)		
Metric for university examinations	0.6	0.12
Metric for number of Ph.D. students graduated	0.4	0.08
Outreach and inclusivity (D)		
Percentage of women (women diversity)	0.4	0.09333333
Economically and socially challenged students	0.6	0.14
Placements and higher studies(E)		
No. of students placed through placements	0.5	0.08333333
No. of students selected for higher studies	0.5	0.08333333

4.3 Data Processing

The data collected for the colleges will be used to calculate the required parameters. The data collected for the colleges is shown in Table 10. The required data, representing the sub-criteria, is calculated using certain logic. A few of them are shown in Eqs. (1), (2), and (3).

$$\text{Student strength} = \frac{\text{(Intake of students in all years}}{\text{Total Student Strength}} + \frac{\text{Student pursuing doctoral program)}}{\text{Total Student Strength}} \tag{1}$$

$$\text{Faculty} - \text{student ratio} \doteq \frac{\text{Total no. of Faculty members}}{\text{Total Student Strength}} \tag{2}$$

$$\begin{aligned}&\text{Faculty with Phd ratio}\\&\quad\doteq \frac{\text{No. of Faculty members with Ph.D qualification}}{\text{Total no.of Faculty members}}\end{aligned} \tag{3}$$

The required data, representing the sub-criteria, calculated using the logic described in (1), (2) and (3), is shown in Table 7. The data has been collected for top 100 colleges, and only a sample top five and bottom five details have been listed in the tables henceforth. The data collected for the colleges (listed in Table 8) will be used to calculate the required parameters.

Table 7 Data with the required fields for further calculations

1.875769	323.600574	0.998353	…	3.403125	0.059053	0.059053
1.078963	262.288870	0.328622	…	1.421687	0.115990	0.013532
1.127569	588.877528	0.382530	…	0.498853	0.100244	0.100244
…						
0.892082	317.235258	0.791519	…	1.204248	0.121884	0.005775
0.729562	348.153933	0.893701	…	0.743961	0.160724	0.019242
0.955604	457.302234	0.313501	…	0.462646	0.163752	0.163752

Table 8 List of colleges and their related data

Institution name	UG approved intake	PG approved intake	Current pursuing Ph.D strength	Faculty strength	PhD faculty strength
C1	4360	360	12	349	96
C2	4920	554	5	727	213
C3	1280	638	132	137	61
C4	516	290	99	101	98
C5	3358	1696	1102	340	311
…					
C96	4080	654	19	379	150
C97	2621	2576	2321	644	641
C98	2520	902	27	253	130
C99	6600	618	8	614	182
C100	620	218	145	65	64

4.4 PROMETHEE Calculations for Sub-criteria

PROMETHEE II method promises completeness in ranking unlike PROMETHEE I. Using the weights calculated from AHP technique, PROMETHEE ranks the institutions in multiple steps as described below. These steps will be repeated for all the criteria. The below process is shown for the first criterion which has three sub-criteria.

Step 1: The evaluation matrix (decision matrix) in Table 3 is normalized using the formula (4), and the results of this matrix are listed in Table 7.

$$R[i][j] = \frac{\text{matrix}[i][j] - \text{minInColumns}[j]}{\text{maxInColumns}[j] - \text{minInColumns}[j]} \tag{4}$$

Step 2: Determine the estimated difference in the alternatives with reference to the other alternatives.

So, we perform a pairwise comparison of every institution with every other institution to form a difference matrix R of size (row count*rowcount-1, column count). The number of columns remains the same. However, the number of rows increases by (row-1) times. In this case, we get a matrix of size $R(9900,3)$, i.e., (100*99, 3). The column size will depend on the number of sub-criteria considered under a criterion. In this case, we consider the first criteria which has three sub-criteria under it.

Step 3: Determine the preference Function P.

Further, we modify the matrix so obtained to form a preference matrix P by using the logic (5).

$$\begin{aligned} &\text{if } R[i][j] < 0, P[i][j] = 0 \\ &\text{else} \quad P[i][j] = R[i][[j]] \end{aligned} \tag{5}$$

On doing so, all the negative values in the matrix becomes zero forming the preference matrix.

Step 4: Determine the aggregated preference AggrPref.

After the preference matrix is formed, an aggregated preference matrix AggrPref[][] is calculated using Eq. (6), taking into consideration the criteria weights, which was obtained from the AHP process.

$$\text{AggrPref}[i][j] = \frac{\sum_0^n \text{weight}[j] * P[i][j]}{\sum_0^n \text{weight}[j]} \tag{6}$$

Step 5: Depending on the values of the alternatives, determine the ranks of the alternatives. The leaving and entering flow calculation are shown in Table 9.

Leaving (positive) flow for the ath alternative, Entering (negative) flow for the ath alternative,

$$\frac{1}{m-1}\sum\nolimits_{b=1}^{m} \text{AggrPref}(a, b) \quad (a \neq b) \quad \sum\nolimits_{b=1}^{m} \text{AggrPref}(b, a) \quad (a \neq b) \tag{7}$$

Table 9 Entering flow and leaving flow calculated using Eq. (7)

	C1	C2	C3		C(n−2)	C(n−1)	Cn	Leaving Flow
C1	0	0.136462	0.124255	…	0	0.044815	0.038068	1.08444
C2	0.002957	0	0.015128	…	0.005361	0.024066	0.009563	0.90870
C3	0	0.102161	0	…	0.005222	0.007617	0	1.12084
…								
…								
C(n−2)	0	0	0	…	0	0	0	0.00000
C(n−1)	0	0	0	…	0	0	0	0.00000
Cn	0	0	0	…	0	0	0	0.00000
Entering flow	0.00099	0.07954	0.04646		0.00353	0.02550	0.01588	

Step 6: Calculate the net outranking flow for each alternative

$$\text{LeavingFlow}(a) - \text{EnteringFlow}(a) \tag{8}$$

The net outranking flow for the five criteria are given in Table 10. This table is given as the input to PROMETHEE for the higher-level processing. The weights considered for process are the weights that were calculated by AHP in Sect. 4.2 before they got distributed among the sub-criteria.

Table 10 Net outranking flow of all the criteria

Criteria 1	Criteria 2	Criteria 3	Criteria 4	Criteria 5
1.08346	0.39857	0.23606	0.96286	0.54862
0.82916	0.34089	0.24323	0.97030	0.84835
1.07437	0.52812	0.00000	−0.02143	−0.00553
−0.03179	−0.00753	0.00000	−0.01241	−0.00944
−0.00084	−0.02232	−0.00619	−0.02621	−0.02039
…				
−0.05768	−0.01513	−0.00375	−0.00099	−0.00809
−0.04981	−0.01738	−0.01979	−0.01175	−0.01997
−0.00353	−0.01616	−0.00333	−0.00916	−0.01719
−0.02550	−0.02143	−0.00635	−0.01216	−0.03849
−0.01588	−0.02202	−0.00296	−0.02193	−0.02892

Table 11 Net outranking flow values along with the ranks

College	Leaving flow	Entering flow	Net outranking flow	Rank
C1	1.894921	0.008977	1.88594	1
C2	0.102368	0.026497	0.07587	55
C3	0.083526	0.173502	−0.08998	68
C4	0.095236	0.285865	−0.19063	96
C5	0.083697	0.286096	−0.20240	11
…				
C96	0	0.004555	−0.00456	40
C97	0	0.001859	−0.00186	52
C98	0	0.005072	−0.00507	71
C99	0	0.005068	−0.00507	13
C100	0	0.002764	−0.00276	54

4.5 PROMETHEE Calculations for Criteria

The input for this process is the data shown in Table 10, and the weights have been calculated in Sect. 4.2. The next step is to proceed with the details described in Sect. 4.3. Once the normalized evaluation matrix is calculated, the difference matrix is calculated by comparing the alternatives, i.e., the colleges, pairwise. This results in a matrix corresponding to size 9900*5 where 9900 is the number of colleges multiplied by a number lesser than it to facilitate pairwise comparison. Five represents the number of criteria considered.

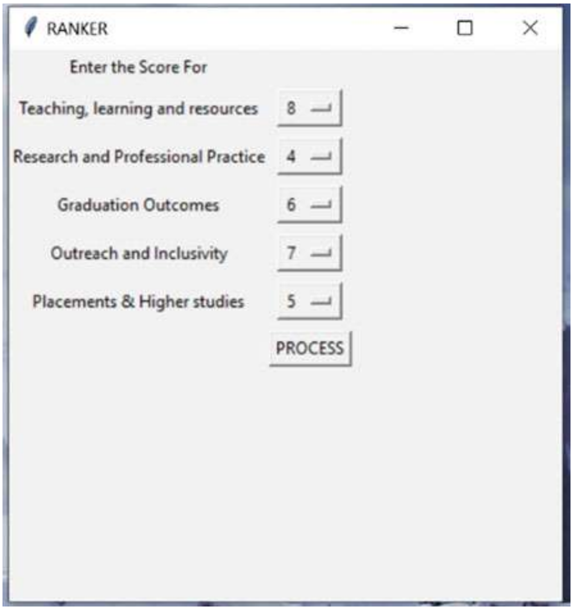
After calculating the difference matrix, preference matrix needs to be calculated using a simple logic mentioned in Eq. 5. The negative values are zeroed, and the positive values remain as it is.

On the basis of the leaving and entering flows calculated, the net outranking flow is calculated using the Eq. 8. Net outranking flow is the difference between the leaving flow and entering flow. Based on this, the alternatives are ranked in the ascending order of the net outranking flow. The final resultant ranked list of colleges along with the calculated net outranking flow is shown in Table 11.

5 Results

The input is the user preference where the user chooses the scores to the criteria considered as shown in Fig. 6. On clicking the “Process” button, the above described calculations take place internally and yield the list of ranked institutions presented in Fig. 7 based on the scores assigned by the user. The proposed model will yield a different set of ranking list for a varied user preference score.

Fig. 6 Input given by the user



Enter the Score For

Teaching, learning and resources 8

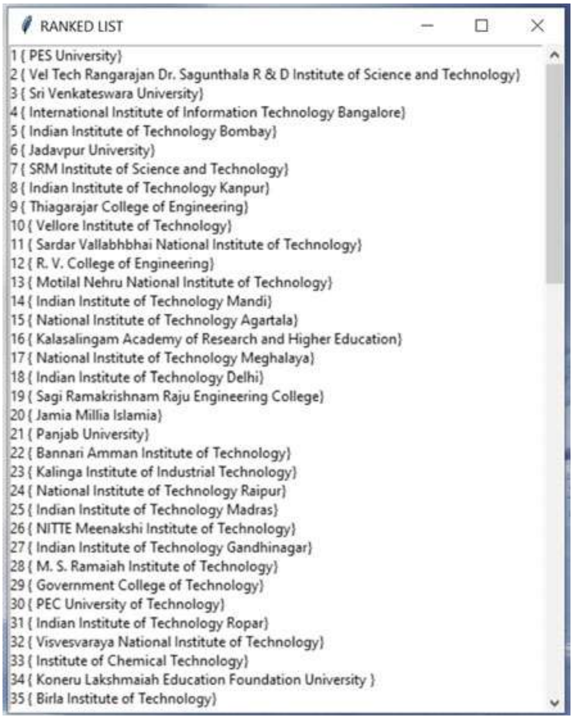
Research and Professional Practice 4

Graduation Outcomes 6

Outreach and Inclusivity 7

Placements & Higher studies 5

Fig. 7 Ranked list of institutions based on user input



```
1 { PES University}
2 { Vel Tech Rangarajan Dr. Sagunthala R & D Institute of Science and Technology}
3 { Sri Venkateswara University}
4 { International Institute of Information Technology Bangalore}
5 { Indian Institute of Technology Bombay}
6 { Jadavpur University}
7 { SRM Institute of Science and Technology}
8 { Indian Institute of Technology Kanpur}
9 { Thiagarajar College of Engineering}
10 { Vellore Institute of Technology}
11 { Sardar Vallabhbhai National Institute of Technology}
12 { R. V. College of Engineering}
13 { Motilal Nehru National Institute of Technology}
14 { Indian Institute of Technology Mandi}
15 { National Institute of Technology Agartala}
16 { Kalasalingam Academy of Research and Higher Education}
17 { National Institute of Technology Meghalaya}
18 { Indian Institute of Technology Delhi}
19 { Sagi Ramakrishnam Raju Engineering College}
20 { Jamia Millia Islamia}
21 { Panjab University}
22 { Bannari Amman Institute of Technology}
23 { Kalinga Institute of Industrial Technology}
24 { National Institute of Technology Raipur}
25 { Indian Institute of Technology Madras}
26 { NITTE Meenakshi Institute of Technology}
27 { Indian Institute of Technology Gandhinagar}
28 { M. S. Ramaiah Institute of Technology}
29 { Government College of Technology}
30 { PEC University of Technology}
31 { Indian Institute of Technology Ropar}
32 { Visvesvaraya National Institute of Technology}
33 { Institute of Chemical Technology}
34 { Koneru Lakshmaiah Education Foundation University }
35 { Birla Institute of Technology}
```

Based on the ranking criteria and sub-criteria considered and with the identified user scores, this paper focuses to develop a ranking model considering user preferences. This model can be fine-tuned to yield a better model with fine-tuned parameter list.

6 Conclusion

This study proposes a system that ranks the educational institutions based on user defined preferences. MCDM methods AHP and PROMETHEE have been implemented to handle the criteria. Web crawling has helped automate the system by extracting the required data automatically. A change in the user preferences results in a different ranked set of institutions.

7 Future Work

The source of data for this is limited to the data from NIRF reports. In future, we can retrieve data from various other ranking reports such as NAAC and QS. There is a huge scope for taking the research of this report further. A model can be developed which compares the various ranking institutions and their reports to give a more analyzed ranking system.

References

1. Hemili, M., Laouar, M.R.: Use of multi-criteria decision analysis to make collection management decisions. In: 2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS), Tebessa (2018)
2. M. Wati, N. Novirasari, H.S. Pakpahan, Evaluation of scholarly performance student using multi-criteria decision-making with objective weight. In: 2018 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC), Bali, Indonesia (2018)
3. Eppe, S., De Smet, Y.: Approximating promethee ii's net flow scores by piecewise linear value functions. Eur. J. Oper. Res. **233**(3), 651–659 (2014)
4. M. Hemili, M.R. Laouar, Use of multi-criteria decision analysis to make collection management decisions. In: 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS), Tebessa (2018)
5. Oguztimur, S.: Why fuzzy analytic hierarchy process approach for transport problems? In: ERSA Conference Papers ERSA, vol. 11, p. 438. European Regional Science Association (2011)
6. Sahu, M.B., Bharne, S.: Study of web crawler and its different types. Int. J. Sci. Eng. Appl. Sci. (IJSEAS) 1–5 (2016)

7. Kang, H., Yoo, S.J., Han, D.: Modeling web crawler wrappers to collect user reviews on shopping mall with various hierarchical tree structure. In: 2009 International Conference on Web Information Systems and Mining, Shanghai (2009)
8. Dong, Q.: Search-engine-oriented theme crawler design. In: 2010 International Conference on System Science, Engineering Design and Manufacturing Informatization, Yi-chang (2010)
9. Ding, Y., Wang, X., Lin, L., Zhang, Q., Wu, Y.: The design and implementation of the Crawler-Inar. In: 2006 International Conference on Machine Learning and Cybernetics, Dalian, China (2006)
10. Gu, J., Wei, F., Yu, K., Cao, R., Shi, Y.: Predicting inquiry and purchase intention of users on automobile websites. In: 2017 IEEE/CIC International Conference on Communications in China (ICCC), Qingdao (2017)
11. Shi, Z., Shi, M., Lin, W.: The implementation of crawling news page based on incremental web crawler. In: 2016 4th International Conference on Applied Computing and Information Technology/3rd International Conference on Computational Science/Intelligence and Applied Informatics/1st International Conference on Big Data, Cloud Computing, Data Science and Engineering (ACIT-CSII-BCD), Las Vegas, NV (2016)
12. Wu, M., Lai, J.: The research and implementation of parallel web crawler in cluster. In: 2010 International Conference on Computational and Information Sciences, Chengdu (2010)
13. Dheeraj, M., Teja, G.S., Yathendra, N., Supriya, M.: Ranking of internet service providers considering customer priorities using multiple criteria decision-making methods. In: 2017 International Conference on Smart Technologies for Smart Nation (SmartTechCon), Bangalore (2017)
14. Supriya, M., Sangeeta, K., Patra, G.K.: A fuzzy based hierarchical trust framework to rate the cloud service providers based on infrastructure facilities (2016)
15. Supriya, M., Sangeeta, K., Patra, G.K.: Comparison of AHP based and fuzzy based mechanisms for ranking cloud computing services. In: 2015 International Conference on Computer, Control, Informatics and its Applications (IC3INA), Bandung (2015)
16. Supriya, M., Sangeeta, K., Patra, G.K.: Trustworthy cloud service provider selection using multi criteria decision making methods. Eng. Lett. **24**, 1–10 (2016)

Using AUDIT Scores to Identify Synbiotic Supplement Effect in High-Risk Alcoholics

Vachrintr Sirisapsombat, Chaiyavat Chaiyasut, Phuttharaksa Phumcharoen, Parama Pratummas, Sasithorn Sirilun, Thamthiwat Nararatwanchai, and Phakkharawat Sittiprapaporn

Abstract Chronic alcohol drinking results in increased intestinal permeability leading to translocation of gut-derived bacterial products. Elevated levels of these products in plasma can induce neuroinflammation probably linking to alcohol's effects on brain function. Prior literature suggests that administration synbiotic may provide intestinal microbial balance and improve gut health. It may show the capacity to ameliorate brain functions in chronic alcohol drinkers. Twenty-one male patients with Alcohol Use Disorders Identification Test (AUDIT) score of 8 or above were administered with synbiotic preparation containing seven probiotics species and three prebiotics once a day before bedtime for eight weeks. There were significantly improved total AUDIT scores ($p = 0.001$), and the data showed significant decreases in scores of the frequency of consuming and blackouts problems from alcohol drinking ($p = 0.011$ and 0.014, respectively). No other differences were observed between trials ($p > 0.05$). These findings suggested that synbiotic consumption could improve the alcohol consumption and addiction levels. The synbiotic may

V.S.-Concept and design of the study, manuscript preparation, statistically analyzed; C.C.-Concept and design of the study, statistically analyzed and interpreted; P.P.-Concept and design of the study; P.P.-Concept and design of the study; SS-Critical revision of the manuscript; T.N.-Critical revision of the manuscript; P.S.-Concept and design of the study, manuscript preparation, critical revision of the manuscript.

V. Sirisapsombat · P. Phumcharoen · P. Pratummas · P. Sittiprapaporn (✉)
Brain Science and Engineering Innovation Research Group, Mae Fah Luang University, Bangkok, Thailand
e-mail: wichian.sit@mfu.ac.th

Neuropsychological Research Laboratory, Department of Anti-Aging and Regenerative Science, School of Anti-Aging and Regenerative Medicine, Mae Fah Luang University, Bangkok, Thailand

C. Chaiyasut (✉) · S. Sirilun
Innovation Center for Holistic Health, Nutraceuticals and Cosmeceuticals, Faculty of Pharmacy, Chiang Mai University, Chiang Mai, Thailand
e-mail: chaiyavat@gmail.com

V. Sirisapsombat · P. Phumcharoen · P. Pratummas · T. Nararatwanchai
Department of Anti-Aging and Regenerative Medicine, School of Anti-Aging and Regenerative Medicine, Mae Fah Luang University, Bangkok, Thailand

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_12

help to prevent and treat alcoholic illnesses. Further investigations for the synbiotic supplement effect on the gut–brain axis lessening the degree of alcohol-induced neuroinflammation in high-risk alcoholics should be studied.

Keywords Synbiotics · Alcoholics · AUDIT · Gut–Brain axis

1 Introduction

Alcohol consumption is associated with several diseases, injuries, and social problems. It is a significant cause of cancer of the mouth, esophagus, and larynx, and long-term, excessive alcohol exposure can lead to liver cirrhosis and pancreatitis. Moreover, gastritis, diabetes, and hypertension, and depression seem to be provoked even by occasional and short-term alcohol exposure [1, 2]. Alcohol Use Disorders Identification Test (AUDIT) is a tool used to screen alcohol consumption levels and addiction treatment efficacy. The ten questions were classified into three domains consisted of recent alcohol use, alcohol dependence symptoms, and alcohol-related problems. Each response has a score ranging from 0 to 4, and the total score sums up a range from 0 to 40. Fleming et al. reported that the AUDIT was the best screening method for the entire range of alcohol troubles than other questionnaires, including the CAGE and the MAS [3]. Hazardous alcohol use implies alcohol ingestion that elevates the risk of detrimental outcomes for the user or others. Hazardous drinking patterns significantly impact public health despite the lack of any current disorders in the individual user. Harmful alcohol drinking is a pattern of alcohol consumption that affects physical and mental health [4]. Alcohol dependence is a behavioral, cognitive, and physiological phenomenon that would expand after repeated alcohol exposure [5].

Many studies demonstrated that gut dysbiosis, an imbalance of the intestinal microbiome, causes several diseases such as diabetes mellitus, allergy, obesity, and inflammatory bowel diseases [6–10]. Recent studies revealed that the gut microbiome could affect the brain functions and emotional behavior involved in the development of psychiatric disorders [10–12]. The microbiota–gut–brain axis shapes the bidirectional communication between the gastrointestinal tract and the brain by the gut microbiome. Both acute and chronic alcohol consumption can alter the microbial composition, bacterial overgrowth, and disturbed the mucosal barrier [13]. Alcohol-damaged cells and microbial debris, especially lipopolysaccharide, are the major factors responsible for alcohol-related inflammation. Ethanol and their toxic metabolites facilitate the generation of reactive oxygen species that can induce a crucial inflammatory transcription factor, nuclear factor-κB (NF-κB) [14]. Moreover, ethanol can change gut permeability and mucosal immunity and stimulate the neuroendocrine hormones [15, 16]. Fukui et al. reported that alcoholic individuals with liver diseases have significantly increased lipopolysaccharide in the blood [17]. At the time of going into the liver from the portal vein, most lipopolysaccharides are detoxified, and only, the eluents can disseminate to the systemic circulation. Alcohol-induced gut inflammation plays a role in liver injury through increasing intestinal permeability and the chance that gut-derived endotoxins, including lipopolysaccharide, translocated to the liver. Keshavarzian et al. suggested that alcoholics with liver disease increased intestinal permeability of more than 40 times compared with healthy subjects and more than 20 times compared with alcoholics without liver disease [18].

Nevertheless, with no notable detoxification organ in transit, the lymphatic route could allow most of the bioactive lipopolysaccharide to leave into the systemic circulation, thereby making it accessible to other organs. Spencer and Hutchison reported that peripheral lipopolysaccharide injection imitating bacterial infection in rodents caused an increase in TNFα in the serum and the CNS. However, lipopolysaccharide could not cross the blood–brain barrier. Generally, the neuroendocrine responses in the central nervous system were triggered by pro-inflammatory cytokines. This interaction caused the activation of the hypothalamopituitary-adrenal axis. The production of neuroendocrine effectors was impaired and modulated the systemic inflammatory condition. Additionally, it led to inflammation in the central nervous system [19]. The study aimed to assess the effect of the synbiotic supplement on the gut–brain axis in high-risk alcoholic patients though AUDIT scores.

2 Materials and Methods

2.1 Study Design

A single-group, pre-test, and post-test experimental design was prepared and conducted during May 2019 and July 2019. The study was approved by and performed under the guidelines of the Research Ethics Committee of Mae Fah Luang University, Thailand, and written consent was obtained from all subjects. Both inclusion and exclusion criteria were applied to all participants while recruiting participants in this study. Informed consent was obtained from all study subjects before initiating study procedures. We recorded all information on demographics and the medical history of all participants.

2.2 Participants

Twenty-four males aged 20–65-years-old participated in this study. All participants were high-risk alcohol drinkers classified by the score of the Alcohol Use Disorders Identification Test (AUDIT) at 8 or above. This classification was measured during both pre-test and post-test clinical trials. In order to get target participants, several criteria were considered. If they met with the following criteria, they were excluded from the study. The exclusion criteria included major abnormal cirrhotic signs, symptoms, and laboratory investigations such as jaundice, ascites, asterixis, vomiting blood, hypoalbuminemia, and coagulopathy. They had a history of the central nervous system and psychiatric disorders such as epilepsy and brain trauma, human immunodeficiency virus, or other immunodeficiency and autoimmune disorder. If they had the following activities, they were also excluded from the study including regular use of drugs other than alcohol, use of antibiotics during the course of this study, consumption of other dietary supplements, and herbs during the course of this study. The history of side effects toward probiotic or prebiotic supplements was excluded.

2.3 Preparation of Synbiotic Intervention

In the present study, all probiotics were manufactured by Lactomason Korea Co., Ltd. Then, they were mixed with prebiotic as a synbiotic by Organic Vita Co., Ltd., Thailand. Synbiotics contained *Bifidobacterium breve* (2.5 × 10^8 CFU), *Bifidobacterium lactis* (7.2 × 10^8 CFU), *Bifidobacterium longum* (4 × 10^7 CFU)*, Lactobacillus paracasei* (9 × 10^7 CFU)*, Lactobacillus reuteri* (2.5 × 10^6 CFU)*, Lactobacillus rhamnosus* (2.5 × 10^8 CFU)*, Lactobacillus salivarius* (1 × 10^7 CFU), respectively. The total of probiotics contained 25 × 10^9 colony forming units per grams including inulin 4 g, fructooligosaccharide 2 g, and galactooligosaccharide 2 g.

2.4 Intervention

All participants were given with aluminum foil sachets containing synbiotics. They consumed one sachet per day for 8 weeks. All participants were reminded to take synbiotics daily by phone. They were allowed to drink normally alcohol. However, they were not allowed to ingest the other including other dietary supplements and herbs while participating in this study. Aluminum foil sachets containing synbiotic preparation were prescribed to all subjects. The subjects took a sachet per day before bed for eight weeks. They were monitored daily by phone and allowed to consume their usual alcoholic beverages. Subjects were requested to keep their routine habits and could notify any illness that occurred during this trial. All subjects were interviewed with the AUDIT at baseline and the end of the study (8th weeks) (see Fig. 1).

2.5 Statistical Analysis

Descriptive analyses for demographic data are presented as mean (standard deviation). Evaluating the mean of differences in the AUDIT score between pre-treatment

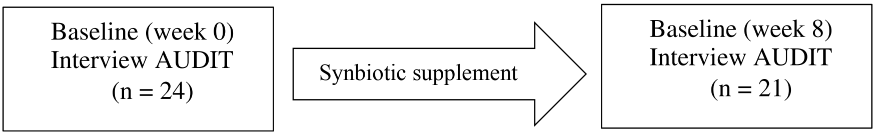


Fig. 1 Illustration of the 8-week single-group synbiotic intervention

and post-treatment, the Wilcoxon signed-rank test was used. In all statistical tests analyzed by a statistical computer program, the null hypothesis was rejected at a $p < 0.05$.

3 Results

Out of twenty-one male participants recruited, three participants were dropped out from the study owing to voluntary loss of follow-up. The mean age of participants was 42.50(±11.66) years (see Table 1). The average weight and height were 62.55 ± 11.51 kg and 166.38 ± 6.91 cm, respectively, and the body mass index (BMI) was calculated. The mean BMI of participants was 22.50 ± 3.36 kg/m^2. Based on the report of the Western Pacific Regional Office of the WHO's Asian BMI classification recommendation, the BMI of participants was categorized as follows: underweight

Table 1 Demographic information of participants

Characteristics	Mean(±SD)
Age (years)	42.50 (±11.66)
Weight (kg)	62.55 (±11.51)
Height (cm)	166.38 (±6.91)
BMI (kg m^{-2})	22.50 (±3.36)
BFP (%)	16.69 (±4.43)
Drinking history (years)	12.05 (±6.79)
BUN (mg/dl)	12.84 (±2.98)
Cr (mg/dl)	1.09 (±0.12)
Alb (g/dl)	4.63 (±0.34)
Underlying disease	Frequency (%)
DM	3 (14.3)
HT	8 (38.1)
DLP	5 (23.8)

Remark: BMI: body mass index; BFP: body fat percent; BUN; blood urea nitrogen; Cr: blood creatinine; Alb: blood albumin; DM: diabetes mellitus; HT: hypertension; DLP: dyslipidemia

(BMI < 18.5) 9.52%, normal weight (BMI 18.5–22.9) 52.38%, overweight (BMI 23–27.5) 28.58%, and obesity (BMI > 27.5) 9.52%. All participants had BUN, creatinine, and albumin levels within normal limits (see Table 1).

According to the accomplishment of compliance forms, all participants consumed assigned synbiotics between the studies about 85%. On the report of general routine activities and food frequency questionnaire data, there were no statistically significant changes in their consumption habits in 8 weeks of trial ($p > 0.05$).

There were significantly improved total AUDIT scores ($p = 0.001$), and the data showed significant decreases in scores of the frequency of consuming and blackout problems from alcohol drinking ($p = 0.011$ and 0.014, respectively). No other differences were observed between trials ($p > 0.05$) (see Table 2).

Table 2 Changes of Alcohol Use Disorder Identification Test (AUDIT) score

Domains	Question content	Pre-intervention	Post intervention	Mean delta[a]	*p*-value[b]
Hazardous	Frequency of drinking	3.35 (0.75)	2.90 (0.91)	−0.45	0.011*
Alcohol use	Typical quantity	2.40 (0.75)	2.45 (0.61)	+0.05	0.564
	Frequency of heavy drinking	2.95 (0.69)	2.65 (0.75)	−0.30	0.096
Dependence symptoms	Impaired control over drinking	3.05 (0.76)	2.90 (0.79)	−0.15	0.180
	Increased salience of drinking	2.95 (0.83)	2.80 (0.70)	−0.15	0.317
	Morning drinking	2.90 (0.72)	2.60 (0.75)	−0.30	0.058
Harmful alcohol use	Guilt after drinking	2.60 (0.68)	2.65 (0.67)	+0.05	0.317
	Blackouts	3.05 (0.51)	2.60 (0.60)	−0.45	0.014*
	Alcohol-related injuries	2.20 (0.62)	2.20 (0.52)	0	1.00
	Others concerned about drinking	2.20 (0.62)	2.10 (0.55)	−0.10	0.157
Total		27.50 (4.07)	25.65 (3.98)	−1.85	0.001*

SD = standard deviation

[a] calculated by using Wilcoxon signed-rank test for both paired pre-test and post-test

[b] represents the mean of pre-test to post-test score delta for all individual paired data

*indicates the significant difference between samples ($p < 0.05$)

4 Discussion

Several pathophysiological mechanisms of chronic alcohol consumption are represented. First, ethanol increases gut permeability and alters microbiome balance. This results in the translocation of lipopolysaccharide across the gut epithelium. Second, ethanol triggers lipopolysaccharide dissemination through the lymphatic route. Third, chronic exposure to alcohol negatively impacts the liver's ability to detoxify lipopolysaccharide and decreases its capacity to produce a main anti-inflammatory cytokine, interleukin (IL)-10. The TLR4 pathway is activated through LPS in the portal, and systematic circulation leads to the activation of Kupffer cells and multiple pro-inflammatory cytokines. The production and release of cytokines, including TNF-α, IL-1β, IL-6, and IL-8 by Kupffer cells, will also stimulate immune cells such as neutrophils, monocytes, and lymphocytes. These cells will respond to Toll-like receptors activation of Kupffer cells through secreting anti-microbial peptides, conduction ROS, and increase phagocytic activity [20]. Liver cells react to inflammation via the creation of acute-phase proteins, including LPS-binding protein, fibrinogen, C-reactive protein, and ceruloplasmin [21–23]. Eventually, chronic alcohol ingestion impairs the production of neuroendocrine effectors that modulate the systemic inflammatory condition as well as lead to inflammation in the central nervous system [19]. Among the concept of the microbe–gut–brain axis, the gut function is not only impacted by the brain, but also brain function can be affected through gut factors covering that of its microbiome [24, 25].

The previous study in alcoholic, non-alcoholic cirrhosis, and hepatic encephalopathy patients showed that probiotic supplement for twenty-four weeks decreased risk of hepatic encephalopathy and improved liver disease severity [26]. The past studies have demonstrated that both alcoholic liver disease patients and experimental animal models increase plasma lipopolysaccharide levels [17, 27]. The study in alcoholic hepatitis patients administrated with *Lactobacillus subtilis*, and *Streptococcus faecium* resulted in a decrease of gut-derived microbial lipopolysaccharide [28]. Bang et al. also showed that the supplement of probiotics helped to reduce pro-inflammatory cytokines, IL-1β, and TNF-α [29]. Depression and psychological stress were previously reported by several clinical studies to show that they were related to exacerbations of inflammatory bowel disease and with the pathogenesis of irritable bowel syndrome [30]. In addition, serotonergic dysregulation and the decline of gut barrier function through mast cell-dependent and mast cell-independent mechanisms might be concerned [31, 32]. Many studies indicate that probiotics favorably affect brain function in healthy individuals, holding out a declaration that focusing on the gut microbiome in alcohol-dependent individuals could help defray alcohol's effect on brain function in the CNS related to alcohol-associated behaviors [33–35]. Likewise, studies support probiotic supplementation can improve anxiety and depressive symptoms considered to induce an inclination toward drinking in alcoholic patients [36]. One previous study also revealed that the improvement of the intestinal integrity with the decrease of LMER could be happened by consuming of synbiotics supplement for eight weeks [37]. As a result, lipopolysaccharide's translocation into systemic circulation and lymphocyte to monocyte ratio, one of the systemic inflammatory markers, decreased significantly. Heyers and et al.'s findings showed quinolinic acid, neuroactive kynurenine pathway metabolites through the direct transformation of L-tryptophan by brain tissue, happened in states of CNS inflammation, yet not by normal brain condition [38]. The improvement of some cognitive functions in high-risk alcoholic participants, which were recognized from the diminishing of the AUDIT scores, might be associated with the decline of lipopolysaccharide level and neuroinflammation after the synbiotic supplement. Further studies should be done to clarify why synbiotic supplementation could ameliorate some brain functions in high-risk alcoholic participants.

5 Conclusion

The results supported that consumption of synbiotics for 8 weeks helps to improve some domains of AUDIT scores in high-risk alcoholic participants. Our results supported that the gut–brain axis plays a vital role in the prevention and treatment of alcohol-induced liver disease and alcohol dependence. However, this recent study has several factors to be concerned including limited sample size, short duration of the study, subjective outcome measurement, and no-placebo control. Hence, future studies should be required to confirm the results of the present study.

Acknowledgements The authors would like to acknowledge the National Council of Thailand (Grant No. 126/2561); Tobacco Authorities of Thailand (Grant No. 4/2560); Postgraduate Studies of Mae Fah Luang University; School of Anti-Aging and Regenerative Medicine, Mae Fah Luang University for the supports. The authors gratefully acknowledge the Chiang Mai University grant (CMU-grant) and Brain Science and Engineering Innovation Research Group, Mae Fah Luang University (MFU-grant) for the supports and necessary provision. The authors thank all participants involved in this research as the main data provider of this study.

References

1. Anderson, P., Cremona, A., Paton, A., Turner, C., Wallace, P.: The risk of alcohol. Addiction **88**, 1493–1508 (1993)
2. Edwards, G., Anderson, P., Babor, T.F., Casswell, S., Ferrence, R., Geisbrecht, N.: Alcohol policy and the public good, pp. 1–240. Oxford University Press, Oxford (1994)
3. Fleming, M.F., Barry, K.L., MacDonald, R.: The alcohol use disorders identification test (AUDIT) in a college sample. International Journal of the Addictions **26**(11), 1173–1185 (1991)
4. Babor, T., Campbell, R., Room, R., Saunders, J.: Lexicon of alcohol and drug terms, pp. 1–65. World Health Organization, Geneva (1994)
5. World Health Organization: The ICD-10 classification of mental and behavioural disorders: diagnostic criteria for research. World Health Organization, Geneva (1993)
6. Noverr, M.C., Huffnagle, G.B.: Does the microbiota regulate immune responses outside the gut? Trends Microbiol. **12**, 562–568 (2004)

7. Karlsson, F.H., Tremaroli, V., Nookaew, I., Bergström, G., Behre, C.J., Fagerberg, B., Nielsen, J., Bäckhed, F.: Gut metagenome in European women with normal, impaired and diabetic glucose control. Nature **498**(7452), 99–103 (2013)
8. Manichanh, C., Rigottier-Gois, L., Bonnaud, E., Gloux, K., Pelletier, E., Frangeul, L., Nalin, R., Jarrin, C., Chardon, P., Marteau, P., Roca, J., Dore, J.: Reduced diversity of faecal microbiota in Crohn's disease revealed by a metagenomic approach. Gut **55**, 205–211 (2006)
9. Seksik, P., Sokol, H., Lepage, P., Vasquez, N., Manichanh, C., Mangin, I., Pochart, P., Dore, J., Marteau, P.: Review article: the role of bacteria in onset and perpetuation of inflammatory bowel disease. Alimentary Pharmacology and Therapeutics 24 (Suppl.), 11–18 (2006)
10. Sivamaruthi, B.S., Kesika, P., Suganthy, N., Chaiyasut, C.: A review on role of microbiome in obesity and antiobesity properties of probiotic supplements. BioMed Research International 1–20 (2019)
11. Finegold, S.M., Dowd, S.E., Gontcharova, V., Liu, C., Henley, K.E., Wolcott, R.D., Youn, E., Summanen, P.H., Granpeeshen, D., Dixon, D., Liu, M., Molitoris, D.R., Green, J.A.: Pyrosequencing study of fecal microflora of autistic and control children. Anaerobe **16**(4), 444–453 (2010)
12. Chaiyasut, C., Sivamaruthi, B.S.: Influence of Probiotic Supplementation on Brain Function: Involvement of Gut Microbiome, Inflammation, and Stress Pathway. In: Evrensel, A., Ünsalver, B.Ö. (eds.) Gut Microbiota - Brain Axis, pp. 20–33. IntechOpen, London (2018)
13. Leung, C., Rivera, L., Furness, J.B., Angus, P.W.: The role of the gut microbiota in NAFLD. Nature Reviews Gastroenterology and Hepatology **13**(7), 412–425 (2016)
14. Gloire, G., Legrand-Poels, S., Piette, J.: NF-kappaB activation by reactive oxygen species: fifteen years later. Biochem. Pharmacol. **72**(11), 1493–1505 (2006)
15. McClain, C.J., Antonow, D.R., Cohen, D.A., Shedlofsky, S.I.: Zinc metabolism in alcoholic liver disease. Alcohol. Clin. Exp. Res. **10**(6), 582–589 (1986)
16. Lambert, J.C., Zhou, Z., Wang, L., Song, Z., McClain, C.J., Kang, Y.J.: Preservation of intestinal structural integrity by zinc is independent of metallothionein in alcohol-intoxicated mice. Am. J. Pathol. **164**(6), 1959–1966 (2004)
17. Fukui, H., Brauner, B., Bode, J.C., Bode, C.: Plasma endotoxin concentrations in patients with alcoholic and non-alcoholic liver disease: reevaluation with an improved chromogenic assay. J. Hepatol. **12**(2), 162–169 (1991)
18. Keshavarzian, A., Holmes, E.W., Patel, M., Iber, F., Field, J.Z., Pethkar, S.: Leaky gut in alcoholic cirrhosis: A possible mechanism for alcohol-induced liver damage. Am. J. Gastroenterol. **94**(1), 200–207 (1999)
19. Wang, J.H., Zakhari, S., Jung, M.K.: Alcohol, inflammation, and gut-liver-brain interactions in tissue damage and disease development. World J. Gastroenterol. **16**(11), 1304–1313 (2010)
20. Beier, J.I., McClain, C.J.: Mechanisms and cell signaling in alcoholic liver disease. Biol. Chem. **391**(11), 1249–1264 (2010)
21. Mackiewicz, A., Kushner, I., Baumann, H.: Acute phase proteins molecular biology, biochemistry, and clinical applications. CRC Press, Boca Raton (1993)
22. Baumann, H., Gauldie, J.: The acute phase response. Immunol. Today **15**(2), 74–80 (1994)
23. Wright, S.D., Ramos, R.A., Tobias, P.S., Ulevitch, R.J., Mathison, J.C.: CD14, a receptor for complexes of lipopolysaccharide (LPS) and LPS binding protein. Science **249**(4975), 1431–1433 (1990)
24. Bravo, J.A., Forsythe, P., Chew, M.V., Escaravage, E., Savignac, H.M., Dinan, T.G., Bienenstock, J., Cryan, J.F.: Ingestion of *Lactobacillus* strain regulates emotional behavior and central GABA receptor expression in a mouse via the vagus nerve. Proceeding of the National Academy of Sciences of the United States of America **108**(38), 16050–16055 (2011)
25. Fetissov, S.O., Déchelotte, P.: The new link between gut-brain axis and neuropsychiatric disorders. Current Opinion in Clinical Nutrition and Metabolic Care **14**(5), 477–482 (2011)
26. Dhiman, R.K., Rana, B., Agrawal, S., Garg, A., Chopra, M., Thumburu, K.K., Khattri, A., Malhotra, S., Duseja, A., Chawla, Y.: Probiotic VSL#3 reduces liver disease severity and hospitalization in patients with cirrhosis: a randomized, controlled trial. Gastroenterology **147**(6), 1327–1337 (2014)

27. Rivera, C.A., Bradford, B.U., Seabra, V., Thurman, R.G.: Role of endotoxin in the hypermetabolic state after acute ethanol exposure. Am. J. Physiol. **275**(6), 1252–1258 (1998)
28. Han, S.H., Suk, K.T., Kim, D.J., Kim, M.Y., Baik, S.K., Kim, Y.D., Cheon, G.J., Choi, D.H., Ham, Y.L., Shin, D.H., Kim, E.J.: Effects of probiotics (cultured Lactobacillus subtilis/Streptococcus faecium) in the treatment of alcoholic hepatitis: Randomized-controlled multicenter study. Eur. J. Gastroenterol. Hepatol. **27**(11), 1300–1306 (2015)
29. Bang, C.S., Hong, S.H., Suk, K.T., Kim, J.B., Han, S.H., Sung, H., Kim, E.J., Kim, M.J., Kim, M.Y., Baik, S.K., Kim, D.J.: Effects of Korean Red Ginseng (*Panax ginseng*), urushiol (Rhus vernicifera Stokes), and probiotics (*Lactobacillus rhamnosus* R0011 and *Lactobacillus acidophilus* R0052) on the gut-liver axis of alcoholic liver disease. Journal of Ginseng Research **38**(3), 167–172 (2014)
30. Mawdsley, J.E., Macey, M.G., Feakins, R.M., Langmead, L., Rampton, D.S.: The effect of acute psychologic stress on systemic and rectal mucosal measures of inflammation in ulcerative colitis. Gastroenterology **131**(2), 410–419 (2006)
31. Wallon, C., Yang, P.-C., Keita, A.V., Ericson, A.-C., McKay, D.M., Sherman, P.M., Perdue, M.H., Söderholm, J.D.: Corticotropin-releasing hormone (CRH) regulates macromolecular permeability via mast cells in normal human colonic biopsies in vitro. Gut **57**(1), 50–58 (2008)
32. Demaude, J., Levêque, M., Chaumaz, G., Eutamène, H., Fioramonti, J., Bueno, L., Ferrier, L.: Acute stress increases colonic paracellular permeability in mice through a mast cell-independent mechanism: involvement of pancreatic trypsin. Life Sci. **84**(23–24), 847–852 (2009)
33. Bercik, P., Denou, E., Collins, J., Jackson, W., Lu, J., Jury, J., Deng, Y., Blennerhassett, P., Macri, J., McCoy, K.D., Verdu, E.F., Collins, S.M.: The intestinal microbiota affects central levels of brain-derived neurotropic factor and behavior in mice. Gastroenterology **141**, 599–609 (2011)
34. Tillisch, K., Labus, J., Kilpatrick, L., Jiang, Z., Stains, J., Ebrat, B., Guyonnet, D., Legrain-Raspaud, S., Trotin, B., Naliboff, B., Mayer, E.A.: Consumption of fermented milk product with probiotic modulates brain activity. Gastroenterology **144**(7), 1394–1401 (2013)
35. Gorky, J., Schwaber, J.: The role of the gut-brain axis in alcohol use disorders. Prog. Neuropsychopharmacol. Biol. Psychiatry **65**, 234–241 (2016)
36. Tang, Y., Banan, A., Forsyth, C.B., Fields, J.Z., Lau, C.K., Zhang, L.J., Keshavarzian, A.: Effect of alcohol on miR-212 expression in intestinal epithelial cells and its potential role in alcoholic liver disease. Alcohol. Clin. Exp. Res. **32**(2), 355–364 (2008)
37. Sirisapsombat, V., Pratummas, P., Phumcharoen, P., Nararatwanchai, T., Chaiyasut, C., Sirilun, S., Sittiprapaporn, P.: The effect of synbiotic supplement on alcohol use disorders identification test and biochemical parameters, gamma glutamyl transferase, lipopolysaccharide and immunoglobulin A levels, in high risk alcoholics. Asian Journal of Medical Sciences **11**(1), 1–6 (2020)
38. Heyers, M.P., Saito, K., Major, E.O., Milstien, S., Markey, S.P., Vickers, J.H.: A mechanism of quinolinic acid formation by brain in inflammatory neurological disease. Attenuation of synthesis from L-tryptophan by 6-chlorotryptophan and 4-chloro-3-hydroxyanthranilate. Brain 116(Pt. 6), 1425–1450 (1993)

Learning-Based Macronutrient Detection Through Plant Leaf

Amit Singh and Suneeta V. Budihal

Abstract The paper proposes a deep learning framework using two deep learning architectures, Keras and Pytorch to analyze the three macronutrients present in the plants basically nitrogen (N), phosphorous (P), potassium (K), i.e., NPK by the convolutional neural network (CNN). Agriculture is the backbone for the economy of a country, especially in the developing nations. Demand for food increases with an increase in population. To meet the increasing need for food, farmers need to maximize the productivity and balance the economy to reduce the losses. The plants require various minerals and nutrients for healthy growth and fruit development. Plant nutrients should be in proper proportion to keep plant healthier and less susceptible to pests. The nutrient analysis can be done by two techniques invasive and non-invasive techniques with their own advantages and disadvantages. Invasive or traditional methods are time-consuming and are costly, whereas non-invasive methods have proved its significance in recent years. The proposed methods are cost-effective and consume less time compared to conventional methods. The proposed framework provides an accuracy of 91% using Keras and 95% using Pytorch.

Keywords Keras · Non-invasive · CNN · Deep learning · NPK

1 Introduction

The increasing population, global warming, less advanced technical equipment or devices are some of the challenges for farmers. They need to increase the food productivity in turn profits with minimal losses. Therefore, farmers need to update to smart farming techniques to increase productivity and minimize losses.

A. Singh (✉) · S. V. Budihal
KLE Technological University, Hubballi, India
e-mail: amitsingh11097@gmail.com
URL: http://kletech.ac.in

S. V. Budihal
e-mail: suneeta_vb@kletech.ac.in

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_14

The increasing demand for food had led to the enhanced use of fertilizers. To increase crop yield, plants should contain predominantly NPK components. Proper nutrients' proportions are needed for healthy plants and crops. Improper use of fertilizers can result into poor quality fruits and vegetables, lacking in size, color, and most of the times the quantity. The nutrients associated with the plants are categorized into two sections, i.e., macronutrients and micronutrients. The concentration of micronutrient is generally higher on the surface soil and reduces as we move deeper. These are required in small quantities for plant growth, but any deficiency of it will affect the plants. Some of the common micronutrients are iron (Fe), manganese (Mn), copper (Cu), zinc (Zn), and molybdenum. Macronutrients are needed in a large amount and are essential elements used in plant growth. The major macronutrients are NPK along with sulphur (S), magnesium (Mg), calcium (Ca), etc.

To make agriculture an intelligent and smart system, new technologies can help to address challenges of farmers. Precision agriculture is the new emerging technology consisting of Global Positioning System, sensors and other technologies for better crop management. Concentrations of macronutrients in the soil are generally determined before the site is distributed. The majority of macronutrients is absorbed from the soil. The micronutrients are part of basic substances present in plant cell. The quantity of NPK needed depends on the type of crop and status of growth. The amount of fertilizer to be used is based on the contents of NPK nutrients present in the soil. Optimization of plant yield by minimizing the fertilizer requirement is the topic interest in present researchers. Recently, deep learning (DL) and artificial intelligence (AI) have been applied in many fields such as industry, automation, agriculture, medicine, and biology. DL is specifically applied for speech synthesis and recognition, visual object recognition and tracking, image to image translation, etc. The authors in [1] proposed deep convolutional network (DCN) to achieve results in the processing of various modalities of information. DL models are used to forecast [2] and identify plant diseases and their classification, weed detection, farmland management, etc. The authors surveyed 40 cases [3] using DL frameworks on agriculture.

Based on image processing, studies have focused on agriculture, analyzing the nutrient deficiency and symptoms of diseases in plants using an artificial neural network (ANN) [4]. In [5], the authors reviewed ANN for agriculture under various conditions. In [6], the authors discussed image processing, where the background of image, image capture situations, segmentation, etc., are elaborated. Pathological analysis of plants is discussed using DL in most of the literature. The authors [7] experimented using CNN models to detect plant diseases in 29 crops, which consisted of 87, 848 dataset of images, with 99.53% of accuracy. In [8], the authors used NN model for tomato plants to segregate the plants as diseased or healthy, using mobile devices. A CNN model was built that used 10, 413 images for training and tested on 10,413 images comprising of 22 species of crops and weeds with accuracy of 86.2%.

A model was developed [10] that can recognize 13 plant diseases. By using a DL framework Caffe, authors achieved a precision from 91 to 98% for separate class tests. In [11], they proposed a robust DL-based disease detector and pest recognition for tomato plants, where three families of architecture are combined. In [12], ML approaches w.r.t. nitrogen (N) estimation in precision agriculture was surveyed. Crop N status was estimated during side-dressing operations in [13]. Maize leaf was assessed using NN model based on reflectance of a maize canopy in three channels in the camera. With the help of aerial images and ANN, authors applied radial basis function and multilayer perceptron to predict N content in real time [14]. In [15], genetic algorithm (GA) was used and they described identification of N and K deficiency in tomato plant and texture features with an accuracy of 82.5%.

An ANN model was developed in [16] to estimate the nutrients in an area of tomato leaf, under changing climatic conditions. Support vector machine (SVM) was analyzed in [17] with improved methodology by introducing the near IR spectrogram. In [18], the authors estimated N status with a reflectance spot sensor and Bayes' theorem using multi-spectral imaging for the prediction model. Furthermore, for wheat leaves a method was proposed in [19] where computational intelligence in vision sensing for estimation was discussed. A method of fusing the DL-based multilayer perceptron (DL-MLP) with committee machines was proposed in [20]. In , the authors achieved image base, low cost, simple, and accurate N estimation. The authors also presented a comparison of many techniques w.r.t. nitrogen prediction.

The survey over the state-of-the-art techniques has led to an investigation of macronutrient deficiency detection using non invasive methods. To address the identified challenges in the previous work, a model was developed using CNN architecture for the classification of plants based on deficiency of macronutrients NPK. Since the images of nutrient deficiency leaves were present in very small scale, it was difficult to train the model to get better accuracy and to train with other DL algorithm as other DL algorithm need dataset in large scale. One of the challenges we faced while collecting the image dataset was to take an image in different weather conditions like windy, rainy, and sunny condition. Images with different weather condition help to train the model to make real-life scenario. Since the background of the image is heterogeneous, which is an essential feature for real-life images.

To train the model with DL architecture has a number of trainable parameters. For example, VGG16 is an architecture which has 138 million parameters. To train the CNN architecture from beginning or fine-tuning to these types of architecture take a lot of time in a CPU environment. Hence, the use of Google Colab to train the model to get fairly good training speed was preferred. Google Colab is a free cloud service by Google, which supports free GPU and has 12GB NVIDIA Tesla K80 GPU. We have considered total four classes such of nutrient deficiency, three classes for nitrogen, phosphorous, and potassium deficiency and one class is for healthy leaf images. Symptoms of deficiency of nutrient in the plant are analyzed with the help of the leaves. Sample images for each nutrient deficiency are shown in Figs. 1, 2, 3 and a healthy plant leaf in Fig. 4.

A total of 2411 images of rice plant were considered, where some of the images have nitrogen deficiency, phosphorous deficiency, potassium deficiency, and some healthy leaves. The total number of images are distributed as shown in Table 1. Various mixtures of the image augmentation in order to increase the images in each category are realized. For transformation, image augmentation techniques such as rotating the image by 30°, width shift, height shift, shear transform, random distor-

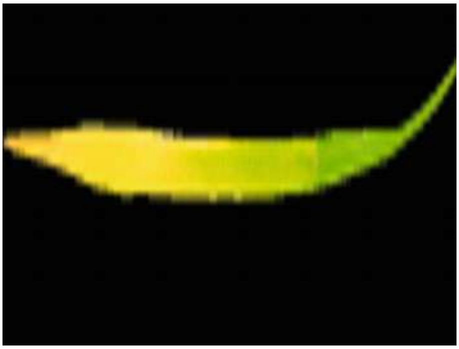


Fig. 1 A rice plat leaf indicating nitrogen deficiency through the color change

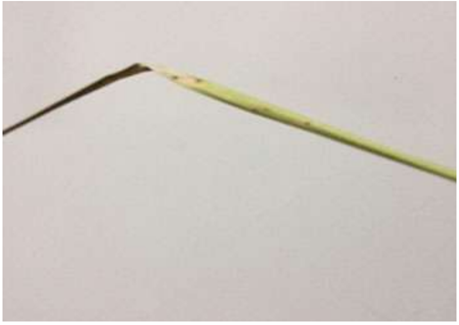


Fig. 2 A rice plat leaf indicating phosphorous deficiency through the color change

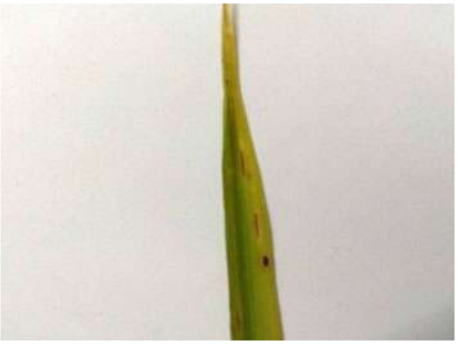


Fig. 3 A rice plat leaf indicating potassium deficiency through the color change

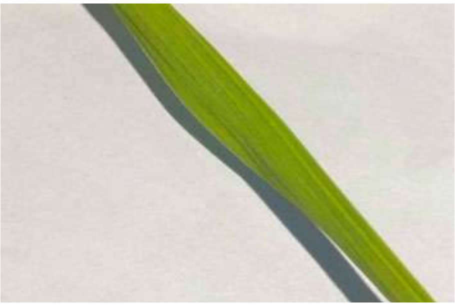


Fig. 4 A healthy rice plant leaf considered for classification using AI solutions while training the proposed model

Table 1 Images of plant leaf for various classes

Name of the class	Number of collected images
Nitrogen	407
Phosphorous	481
Potassium	523
Healthy	1000

tion, vertical flip, and horizontal flip are used. Every image augmented output is the subset of original dataset. The variation of images helps the model in training phase, to learn the detailed features of nutrient deficiency, which can overcome the problem of under fitting and over fitting and can lead to lower accuracy. We have annotated the images by putting images of different class into different folders.

2 Proposed Framework

Plants and crops are sensitive to the disorders due to deficiency of nutrients. There are several factors influencing nutrients' deficiency such as environmental conditions, temperature, nutritional excess or shortage, humidity, and light. The deficiency leads to change in physical characteristics of the leaves such as a color, shape, and size. The similarities in patterns block us from identifying which makes monitoring and classification a challenging task. It is very important to detect deficiency of nutrients and treatment at the initial phase, to avoid losses in yields.

We proposed a framework for macronutrient, i.e., NPK deficiency in the leaf by CNN in Keras and FastAI in Pytorch. CNN is a DL architecture capable of providing efficient, high-quality output and can deal with many different types of datasets. To get the best result for the classification of images using DL algorithm, we proposed a framework that used TensorFlow with Keras and the other is Pytorch. The detailed information of CNN is discussed in further sections.

CNN is a family of DNNs, which Keras utilize the spatial structure of the data, learn from that data and comes out with some useful output. CNN has total three layers. The first layer is a convolutional layer, which is used for extracting the features from data. Pooling layer is the next layer used to reduce the size, number of parameters and decrease the computational time. This layer is operated on the feature maps. The last one is fully connected layer, used to take result of the pooling process, and classify into different classes. If CNN is provided with the data, it learns some local features from the data. In convolutional layers, these local feature are learnt. After looking for local features, CNN will generate some specific activation patterns which will represent global existence of these local features. In CNN, these activation vectors are generated by the fully connected layers. For multi-layer features, each layer has its own activation function. The convolutional layers consist of many convolutional filters also called as Kernels. The Kernels are used for learning local features from the dataset.

In mathematics, convolutional function is operation of two functions f() and g() to generate a new function using 2D, which shows the modification of one function with the other. 2D convolutional layer usually takes three-dimensional input. The convolutional filters are passed over the images to check every pixel of the input and moves the filter until the filter scans the entire image. Basically, convolution is a dot product of the pixel value of image and weight defined in the filter. If an input image is represented by X and a filter is represented by f, then

$$Z = X * f \tag{1}$$

To predict the size of an output for complex inputs or filters:
Dimension of image = (n, n), Dimension of filter = (f, f), Dimension of output will be $((n - f + 1), (n - f + 1))$ A feed-forward neural network is a composition of a number of functions,

$$f(x) = f_L(\ldots f_2(f_1(x; w_1); w_2)\ldots), w_L) \tag{2}$$

Each function of f_l takes as an input a datum x_l, a parameter vector w_l and produces an output datum x_{l+1}. The parameters w = $(w_1, \ldots, w_L)$ are learnt from the data for solving a target problem, like classifying images. The next step is to apply filter to image with output y of convolution layer. The filters are usually 3D. Hence, they operate on a tensor with x and K as a channels. Furthermore, there are K' filters that generate a K' dimensional map y. The equation is given by,

$$Y_{i'j'k'} = \sum_{ijk} W_{ijkk'} X_{i+i',j+j',k} \tag{3}$$

By mapping several linear functions, in addition to nonlinear functions, CNNs are built. The simplest nonlinear function is rectified linear unit (ReLU). $F(z)$ is zero when z value is less than zero and $F(z)$ is equal to z when z value is equal or above zero with a range between zero to infinity.

$$F(z) = max(0, z) \tag{4}$$

There are two pooling techniques, maximum pooling and average pooling. In maximum pooling, it extracts the maximum number in the filters or windows. In average pooling, the average number is extracted. The formula for maxpooling layer is given by Eq. 5,

$$Y_{ijk} = max\{y_{i'j'k'} : i < i^{'} < i + p, j < j^{'} < j + p\} \tag{5}$$

The last layer of CNN is used for combining of features learnt by different filters. It takes the output from the previous pooling layers, flatten them, and convert into a single vector, which will be the input for next layers. It takes the input after analyzing the features, and it applies weights for classification application. Fully connected layer works on backpropagation technique to determine the accurate weights of the

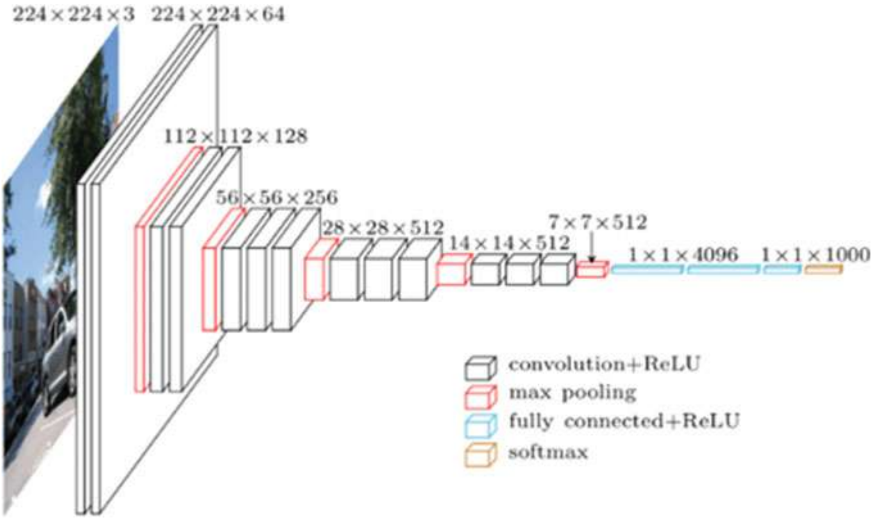


Fig. 5 Network architecture of CNN

neurons. Each neuron has weights that prioritize the classes. The neurons finally vote on each label and the label with highest vote is the winner and is the classification decision. Now we need to put all these together to form a DNN model, from the raw images till final classification decision. After weaving all the layers, we get the model as shown in Fig. 5.

3 Convolution Neural Network in Keras

Keras is a high-level NN application program interface (API), which is basically written in Python and is used for running the TensorFlow library. It was developed for the large-scale deployment with the focus on consuming less computational time and fast experimentation compared to other DL algorithms. Since the computational time is less in this framework, it is a better option for the researcher to train and build the model. Keras supports both convolutional and recurrent networks or the combination of both the networks. Before training the model, first we need to preprocess the image to enhance the images. After preprocessing, next step is image segmentation. Image segmentation is used to focus on the infected region else the non-infected region will decrease the accuracy and will increase loss error in the model. After image segmentation, the next step is feature extraction performed in the segmented region of the leaf, where the important features are extracted and the final step is to train the model for further classification process.

Keras framework is used with TensorFlow to train the model. We observed the accuracy of the model after each epoch and checked for the best accuracy. If for any epoch the accuracy exceeds the highest limit, the model was saved. Likewise, the model was saved with best accuracy. After 50 epochs, we stopped to train the model and evaluated the accuracy of the model. VGG16 CNN architecture is used for training. VGG16 is a sequential type CNN architecture, which uses convolutional filter of $n \times n$ matrix size. After each pooling layer, the convolution filter gets doubled. Fine-tuning was performed to get highest accuracy. In fine-tuning method, pre-trained ImageNet weights of convolution layers were taken. We then initialized weights randomly for densely connected layers.

Transfer learning techniques were applied where all the convolutional layers are not trained. Instead, a pre-trained model was taken, modified and trained the last convolutional layer only. While training the model, we evaluated the hyperparameters. With the help of fine-tuning, hyperparameters are tuned and set the environment for conducting an experiment. Accuracy is taken as the performance metric of the model. Accuracy can be a good performance evaluator, when the samples are not biased towards any individual class. Categorical cross-entropy was used as a loss function for the model. The mathematical formula of SoftMax and cross-entropy loss is shown by Eqs. 6 and 7. We are not training the base model; hence, we freeze all the convolutional layers. Then top three convolutional layers were removed from the base model. Then output of the last convolutional layers is flattened from the remaining convolutional layers. The remaining layers have the pre-trained weights from the ImageNet.

$$f(s)_i = \frac{e^{s_i}}{\sum_j^c e^{s_j}} \tag{6}$$

$$CE = -\sum_i^C t_i log(f(s)_i) \tag{7}$$

Hence, the proposed architecture already becomes familiar with the basic features of the image. Three dense convolutional layers with ReLU activation function and add one denser layer with SoftMax activation function are added at the end. The hyperparameter value is fixed as two at the starting of training as they need not be trained again. The first hyperparameter is a dropout rate. A dropout rate of 0.5 was fixed, means the model will ignore 50 % of neurons of the previous convolutional layers. It is used to reduce over fitting. The second hyperparameter is the learning rate. The learning rate is used to determine the speed of the model to adjust to get local and global minima of the loss function. The learning rate is tuned to 0.01. Adaptive moment estimation (Adam) optimizer was used for training the model. Optimizer is a method, which is used to compute the adaptive learning rates for all parameters.

3.1 FastAI in Pytorch

To identify nutrient deficiency in plants, we can create artificial intelligence web applications using FastAI. It is built on the top of Facebook DL platform Pytorch. FastAI is a DL library used in Pytorch for its backend. It is modern and more robust than the TensorFlow library Keras. The area of plant nutrient, deficiency diagnosis had progressed a lot with time: from the visual visualization to a molecular level diagnosis. We proposed a classifier to detect the nutrient deficiency in plants from the images of leaves, which consists of healthy leaves and nutrient-deficient leaves. In training the classifier first, we have loaded the data. After loading, normalization of data was done to get the ImageNet parameters. We need to normalize the image to make the pixel value of the image equal to mean and standard deviation. Hence, it makes the model easier to train and makes it faster. Transformation of the images is done that is needed for the image segmentation by cropping the images, centring it and by zooming the images. The size of the images should be resized. After preprocessing, defining labels and generating some images, we train the model (Fig. 6).

To train the model, transfer learning technique is used, where we use the pre-trained model to handle the task. The transfer learning takes data and network and shows in the output in the form of metrics. Matrices print the output to show the performance of training. It helps the model to train faster. We have used pre-trained ResNet34 CNN architecture. It has 34 convolutional layers in its architecture and is trained under ImageNet dataset. The residual network or ResNet is known because of various shortcut networks used on the top of feed-forward networks. We will make the model to learn to classify in 10 epochs. FastAI will randomly select training sets and validation sets to minimize the human interface and make the code simple.

Intuitively, a deep neural network should not be hard to fit. If there is a certain number of layers 'N' that has achieved the optimal accuracy on a dataset, then the layers after N can just learn the identified mapping, and then the network will generate

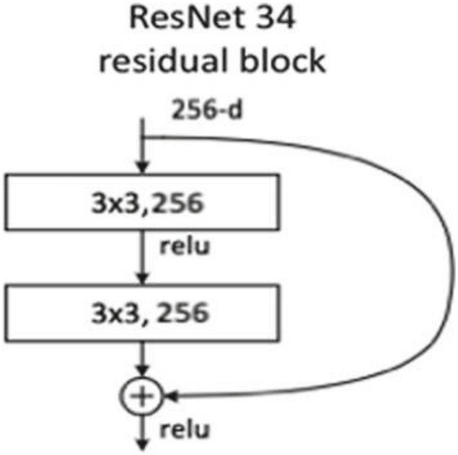
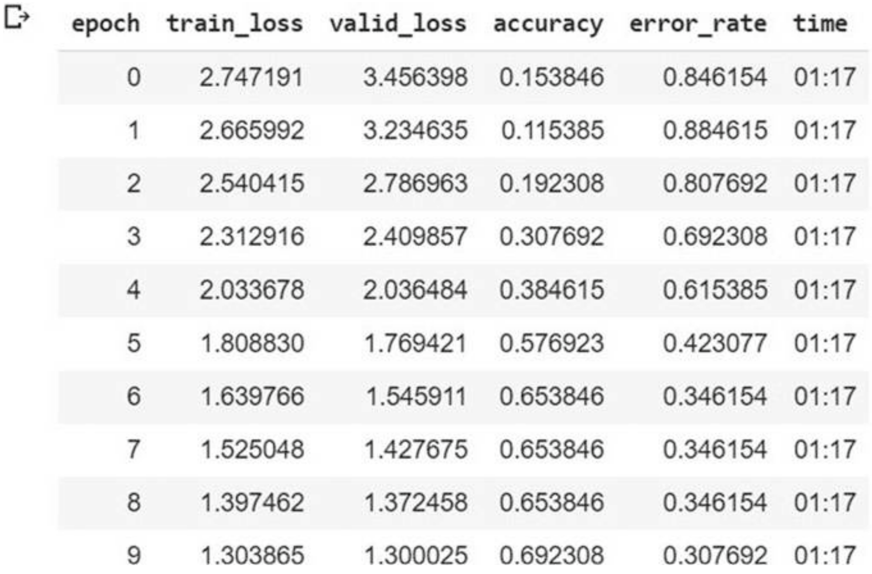


Fig. 6 Block diagram of ResNet34



epoch	train_loss	valid_loss	accuracy	error_rate	time
0	2.747191	3.456398	0.153846	0.846154	01:17
1	2.665992	3.234635	0.115385	0.884615	01:17
2	2.540415	2.786963	0.192308	0.807692	01:17
3	2.312916	2.409857	0.307692	0.692308	01:17
4	2.033678	2.036484	0.384615	0.615385	01:17
5	1.808830	1.769421	0.576923	0.423077	01:17
6	1.639766	1.545911	0.653846	0.346154	01:17
7	1.525048	1.427675	0.653846	0.346154	01:17
8	1.397462	1.372458	0.653846	0.346154	01:17
9	1.303865	1.300025	0.692308	0.307692	01:17

Fig. 7 Accuracy and error rate of trained model before fine-tuning

final output at layer *N*. However, it is not easy for the weights to be pushed to get the exact identified mapping. In Eq. 8, $H(x)$ is the mapping of the layer to be learned (Fig. 7).

$$H(x) = x \tag{8}$$

Hence, we used a residual function instead of directly approximating the underlying mapping,

$$y = F(x) + x \tag{9}$$

where $F(x)$ is the output of the layers (before the ReLU of the last layer) and then the original input x is element-wise added.

4 Results and Analysis

In the first framework, we trained the model with VGG16 CNN architecture. We used transfer learning, fine-tuning techniques to train the model from the scratch. We trained the model twice to get high accuracy. In first attempt, we got 87% accuracy, and on the second attempt, we got 91% accuracy on the testing set. We have noticed that on applying techniques like fine-tuning from pre trained ImageNet weight the accuracy increased. In fine-tuning, we did not train the whole base model, instead we start from the pre-trained weights of ImageNet and train the model by giving our dataset. In second framework, we have used a DL library known as FastAI. FastAI is a free open-source project, which is built on a Pytorch framework. We will make the model to learn for classifying in 10 epochs. FastAI will randomly select the training sets and the validation sets to minimize the human interface and make the code simple.

The model gains high accuracy in just ten epochs, where we check for the wrong prediction done by the trained model so that we can further fine-tune. By checking the wrong prediction, we get an idea that how confident the model is to classify output. The trained model predicts the output correctly most of the times but, sometime the model gives the wrong output with a high confidence value.

We can observe that, if the error rate is slightly higher, there is a need to tune the model and to find the optimal learning rate at which the trained model has highest accuracy rate. Learning rate is one of the most important hyperparameters while training. We should set the learning rate at very optimal value where it should not be either too slow or too fast. To find the perfect learning rate, we have learning rate finder. It will generate automatically the optimal value. After fine-tuning, the model becomes less confused compared to earlier tuning the model. From graph shown in Fig. 8, it was clear that the losses increase very fast w.r.t. the learning rate beyond 0.001. So, learning rate should be kept within a range of 0.000001–0.001.

Now the model is more confident to provide accurate predicted output. The graph is plotted to analyze the relation between Loss function and learning rate. We noticed that the learning rate finder has chosen the higher learning rate for which there should be minimum loss. The higher learning rate means the machine will learn faster. Finally, after fine-tuning, the accuracy of the model is slightly increasing and the model becomes more confident to provide correct output with very high accuracy (Fig. 9).

Figure 10 shows the final output of training and validation set. Initially, the model was over fitting, after training the model, the model reached to optimal and after

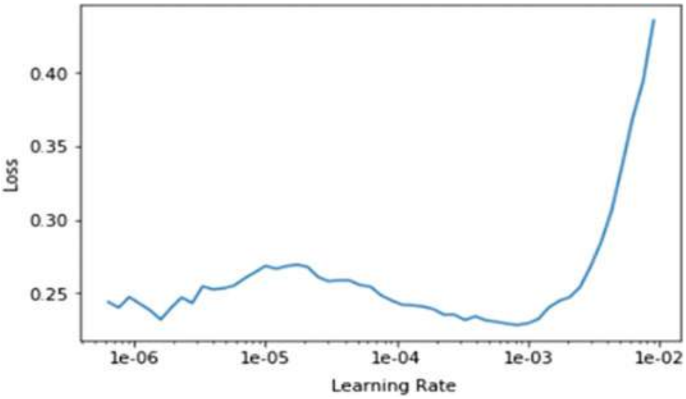
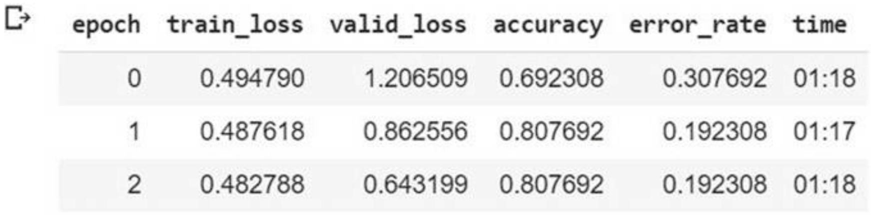


Fig. 8 Graph between loss and learning rate



```
epoch  train_loss  valid_loss  accuracy  error_rate  time
    0    0.494790    1.206509  0.692308    0.307692  01:18
    1    0.487618    0.862556  0.807692    0.192308  01:17
    2    0.482788    0.643199  0.807692    0.192308  01:18
```

Fig. 9 Accuracy and error rate of trained model after fine-tuning

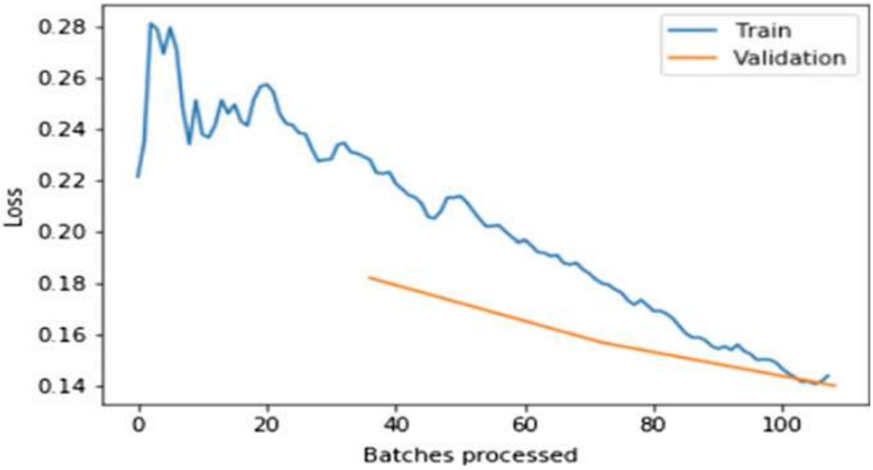


Fig. 10 Loss score for training and validation set

Table 2 Comparison of both prediction models: Keras and Pytorch

Parameter	Keras	Pytorch
Epoch	25	10
Accuracy (%)	91	95
Error (%)	9	7
Total layers	23	34

applying the fine-tuning technique, training and the validation sets coincide meaning the model is neither under fitting nor over fitting.

We have proposed the model with two different frameworks, Keras and Pytorch. Using both the frameworks, we got good results. In Table 2, we have compared some important parameters, for the classification and prediction. Pytorch was able to execute faster than Keras with better efficiency. Pytorch forms dynamically computational graph. Keras form a statically computational graph. Keras is usually used for small datasets as it is comparatively slower. On the other hand, Pytorch is used for large datasets that require fast execution. Hence, Pytorch is better for rapid pro-

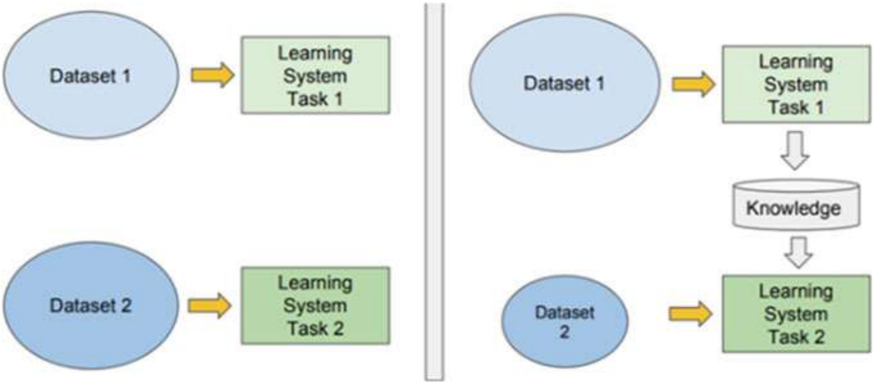


Fig. 11 Difference between the traditional method versus proposed method

totyping in research. Pytorch is good for starting with small-scale projects. Pytorch utilizes and optimizes better for using GPU. While Keras is used for large-scale deployment projects, Keras is the most popular framework in the bigger community, and it has more number of users than Pytorch.

Traditional learning methods are completely isolated processes and occur depending on particular tasks, datasets and training have separated the isolated models on them as shown in Fig. 11. No knowledge is retained back, which can be used for transferring the learning parameters from one model to another. In transfer learning, we can hold the knowledge like features and weights from previously trained models for training new models and can even deal with problems like having lesser data for a new task. Linear Kernel is used for data, which are linearly separable. It is mostly used when a large number of parameter or features are there in a particular dataset. K-means is an unsupervised machine learning algorithm, which separate observations of features into *k* clusters. Since we can identify the amount of clusters, we can use this algorithm easily in classification where we need to divide data into clusters, which can be equal or more than the number of classes.

Since the traditional machine learning algorithms require labeled data, they are not suitable for solving complex queries, which need a huge amount of data. The proposed model uses deep learning algorithms like transfer learning and ResNet-34 architecture. However, deep learning techniques required a huge amount of data than a traditional machine learning algorithm as this is only able to identify the edges like concepts and differences of the layers of neural networks when exposed over a million data. In machine learning algorithms, they are able to learn through pre-programmed defined parameters. It is observed that as the data is increased, the performance of DL algorithms is increased compared to traditional algorithms where the performance used to be almost constant after a while even the data was increased.

5 Conclusion

An ideal DL model for identifying the amount of nutrient in plants should have high accuracy rate and low error rates. We studied the nutrient deficiency including four classes NPK deficiency and healthy plants. We tried to get high accuracy from a limited sample of images. In this paper, we applied DL models based on the CNN architecture. The comparison results between different frameworks demonstrated how efficiently our DL model is able to recognize different classes of nutrient deficiency. The proposed model is experimented with one species of plants but it can extend to other plant species too. The proposed framework provides an accuracy of 91% using Keras and 95% using Pytorch.

References

1. LeCun, Y., et al.: Deep learning. Nature 521(7553). Google Scholar Google Scholar Cross Ref Cross Ref., pp. 436–444 (2015)
2. Mavridou, E., et al.: Machine vision systems in precision agriculture for crop farming. J. Imag. **5**(12), 89–99 (2019)
3. Andreas, K., et al.: Deep learning in agriculture: a survey. Comput. Electron. Agric. **147**, 70–90 (2018)
4. Madhogaria, S., et al.: Pixel-based classification method for detecting unhealthy regions in leaf images. In: InGI-Jahrestagung, pp. 482–490 (2011)
5. Samborska, I.A., et al.: Artificial neural networks and their application in biological and agricultural research. J. NanoPhotoBioSciences **2**, 14–30 (2014)
6. Barbedo, J.G., et al.: Factors influencing the use of deep learning for plant disease recognition. Biosyst. Eng. **172**, 84–91 (2018)
7. Liu, B., et al.: Identification of apple leaf diseases based on deep convolutional neural networks. Symmetry **10**(1), 11–21 (2018)
8. Chaivivatrakul, S., et al.: Automatic morphological trait charact erization for corn plants via 3D holographic reconstruction. Comput. Electron. Agric. **109**, 109–123 (2014)
9. Dyrmann, M., et al.: RoboWeedSupport-Detection of weed locations in leaf occluded cereal crops using a fully convolutional neural network. Adv. Animal Biosci. **8**(2), 842–847 (2017)
10. Mohanty, S.P., et al.: Using deep learning for image-based plant disease detection. Front. Plant Sci. **7**, 1419–1428 (2016)
11. Ferentinos, K.P.: Deep learning models for plant disease detection and diagnosis. Comput. Electron. Agric. **145**, 311–318 (2018)
12. Chlingaryan, A., et al.: Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: a review. Comput. Electron. Agric. **151**, 61–69 (2018)
13. Noh, H., et al.: Dynamic calibration and image segmentation methods for multispectral imaging crop nitrogen deficiency sensors. Trans. ASAE **48**(1), 393–401 (2012)
14. Gautam, R., et al.: Neural network optimisation of remotely sensed maize leaf nitrogen with a genetic algorithm and linear programming using five performance parameters. Biosyst. Eng. **95**(3), 359–370 (2006)
15. Xu, G., et al.: Use of leaf color images to identify nitrogen and potassium deficient tomatoes. Pattern Recogn. Lett. **32**(11), 1584–1590 (2011)
16. Ahmadian-Moghadam, H., et al.: Prediction of pepper (Capsicum annuum L.) leaf area using group method of data handling-type neural networks. Int. J. Agri Sci. **2**(11), 993–999 (2012)
17. Tran, T.-T., et al.: A comparative study of deep CNN in forecasting and classifying the macro nutrient deficiencies on development of tomato plant. Appl. Sci. **9**(8), 1601–1611 (2019)

18. Jones, C.D., Jones, J.B., Lee, W.S.: Diagnosis of bacterial spot of tomato using spectral signatures. Comput. Electron. Agric. **74**(2), 329–335 (2010)
19. Asraf, M.H., et al.: A fuzzy inference system for diagnosing oil palm nutritional deficiency symptoms. ARPN J. Eng. Appl. Sci. **12**(10), 3244–3250 (2017)
20. Sulistyo, S.B., et al.: Ensemble neural networks and image analysis for on-site estimation of nitrogen content in plants. In: Proceedings of SAI Intelligent Systems Conference, pp. 103–118. Springer (2016)
21. Xin, X., et al.: Non-invasive sensing of nitrogen in plant using digital images and machine learning for brassica campestris. Sensors **19**(11), 2448–2458 (2019)
22. Singh, A., Budihal, S.V.: Non-invasive techniques of nutrient detection in plants. In: Intelligent Computing and Applications. Advances in Intelligent Systems and Computing, vol. 1172. Springer, Singapore, pp. 407–417
23. Pavaskar, S., et al.: Real-time vehicle-type categorization and character extraction from the license plates. In: Cognitive Informatics and Soft Computing, pp. 557–565. Springer, Singapore (2019)
24. Suneeta, V.B., et al.: Facial expression recognition using supervised learning. In: Computational Vision and Bio-Inspired Computing, (ICCVBIC), Advances in Intelligent Systems and Computing, vol. 1108, pp. 275–285

Characteristics of Karawitan Musicians' Brain: sLORETA Investigation

Indra K. Wardani, Djohan, Fortunata Tyasrinestu, and Phakkharawat Sittiprapaporn

Abstract The fast development of music research prompts numerous interdisciplinary issues. In the neuroscience field of study, music is being examined related to its impact on the cognitive process or the psychological process behind it. These investigations inspire music's integration to numerous subjects, for example, neuroscience and neuropsychology. A few previous studies showed the distinction between musicians and non-musicians regarding brain structure and brain activities. Instead of differentiating brain activity between musician and non-musician, the present study demonstrated the different brain activity while musicians listened to music regarding their musical experience. Applying the electroencephalography (EEG) recording and source localization in the exploratory methodology toward Karawitan musicians (N = 20), the outcomes demonstrated higher brain activities in tuning into recognizable music, *Gendhing Lancaran*, Javanese traditional music. In addition, the dominant brain activities happened in the temporal lobe while Karawitan musicians listened to *Gendhing Lancaran*, Javanese traditional music.

Keywords Electroencephalography · sLORETA · Brain · Karawitan musician · Music · *Gendhing Lancaran*

I. K. Wardani · Djohan · F. Tyasrinestu
Faculty of Performing Arts, Indonesia Institute of the Arts of Yogyakarta, Yogyakarta, Indonesia

F. Tyasrinestu
Graduate School of Indonesia Institute of the Arts of Yogyakarta, Yogyakarta, Indonesia

P. Sittiprapaporn (✉)
Brain Science and Engineering Innovation Research Group, Mae Fah Luang University, Bangkok, Thailand
e-mail: wichian.sit@mfu.ac.th

Neuropsychological Research Laboratory, Department of Anti-Aging and Regenerative Science, School of Anti-Aging and Regenerative Medicine, Mae Fah Luang University, Bangkok, Thailand

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_15

1 Introduction

Neuroscience is known as a discipline focusing on brain mechanisms. It has been examined music and its cognitive process for quite a long time. Not just has it attempted to explain the mechanism that lies behind perception and reception of music, however, this discipline additionally attempted to comprehend the impact of specific music (or melodic activity) toward the brain structurally and functionally. Many pieces of research had been done to understand the effect generated by particular music activity toward cognitive function. Listening to music passively as a background of certain activity has been proved to improve cognitive performance [1]. Meanwhile, certain treatments of music in terms of tempo and mode were able to stimulate arousal and mood to increase cognitive abilities in spatial tasks [2]. Previous study revealed how musical training is beneficial to alter brain function and enhance cognitive performance outside music [3]. Using the visuospatial task applied to orchestral musicians and non-musicians subject, this previous study tried to provide evidence on how complexity in musical training done by orchestral musicians can affect the activation of Broca's area during the test that enhanced their performance [3]. By comparing naïve, amateur, and professional musicians, Gaser, and Schlaug [4] exhibited how the brain structure is diverse between those subjects.

As to past investigations, researchers assumed a relative contribution of long-term musical practice to be done by both amateur and professional musicians as the main cause. Previous studies compared musicians and non-musicians where the musical experience became an important consideration as prior conditioning to differentiate subjects [1, 5–7]. The primary argumentation of this assumption was depending on the thought that the contribution of long-lasting musical training and certain musical skills acquirement experienced by subjects empowered them to have a different cognitive process. Musicians experienced complex rehearsal processes including physical and mental activities to interpret musical notation as a piece of visual information and change it as a motoric reaction, to remember the melodic expression, to do melodic improvisation, and to distinguish a note with no referential note [5–10]. Those complexities in music rehearsal were considered as brain stimuli to train its capacity and improve its performance as well as develop its perception and recognition. Those previous investigations which utilized both musicians and non-musicians as the primary subject to distinguish melodic experience in terms of 'had received' and 'had not received' melodic educating. It is quite clear to create such a distinction between the subject and to hypotheses the possible difference occurred [11–13].

After all, instead of examined brain activity differences between musicians and non-musicians, the present study aimed to describe brain activity of Karawitan musicians who play traditional Javanese music. Referring to one previous study of musical preference and cognitive style showed that a tendency of individuals who have certain cognitive style as a base of their personality traits and how musical genre became a distinctive variable. Here, we would like to examine how musicians with a traditional different musical background in terms of the genre will perceive audio stimuli. The propose of the current investigation was, consequently, to utilize electroencephalographic activities and the standardized low-resolution electromagnetictomography (sLORETA) methods to quantify the level of cortical activation and to localize the brain area contributing to the scalp recorded via auditory stimuli. Thus, the objective of the present study was to identify the lateralization of musical experiences by means of listening to *Gendhing Lancaran*, Javanese traditional music.

2 Materials and Methods

2.1 Participants

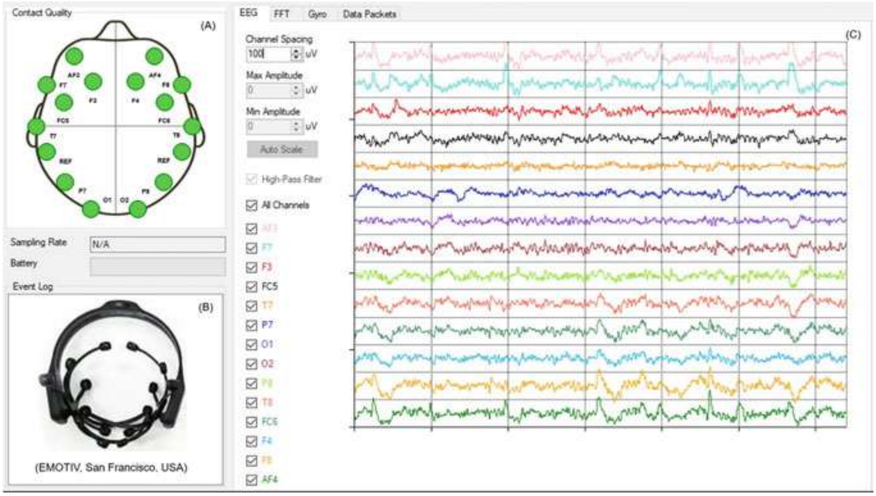
Twenty Karawitan musicians, aged 23–29 (mean 28.25 ± 1.41), were participated in this study. All participants had been explained and had approved the applied procedure. All Karawitan musicians employed in the present study actively learned practical music lessons of *Gendhing Lancaran*, Javanese traditional music, for consecutive three years. Duration of music learning was considered as an aspect of musical experience. The approval of the Graduate School of Indonesia Institute of the Arts, Yogyakarta, Indonesia, on experiment and written consent from each participant were also obtained.

2.2 Stimuli

Stimuli consisted of *Gendhing Lancaran*, Javanese traditional music. The sound was one minute's long. It was presented binaurally via headphones at 85 dB. EEG signals were time-locked to the onset of stimuli presentation. *Gendhing Lancaran*, Javanese traditional music, was chosen as participants' unfamiliar music; participants were told to focus on the auditory stimuli presented via earphones.

2.3 Electroencephalographic Recording

For electroencephalographic recording, the electroencephalographic activities were recorded via wireless neuroheadset (EMOTIV Epoc Plus, EMOTIV, San Francisco, USA) from 14 active electrodes including AF3, F3, F7, FC5, T7, P7, O1, O2, P8, T8, FC6, F8, F4, and AF4, respectively, positioned according to the 10–20 international system of electrode placement, plus ground were applied (see Fig. 1). Reference electrodes were physically applied to the left and right mastoids. EEGs were acquired as continuous signals and amplified with a gain of 30,000 and filtered with a bandpass of 0.1–100 Hz. In addition, the recordings were filtered and carefully inspected for



Sampling Rate N/A

Battery

Channel Spacing 100 uV

Max Amplitude 0 uV

Min Amplitude 0 uV

☑ High-Pass Filter

☑ All Channels

☑ AF3
☑ F7
☑ F3
☑ FC5
☑ T7
☑ P7
☑ O1
☑ O2
☑ P8
☑ T8
☑ FC6
☑ F4
☑ F8
☑ AF4

Fig. 1 **a** 14-channels electrode montage. **b** EEG device (EMOTIV Epoc Plus). **c** Electroencephalographic recording from 14 channels were selectively analyzed in this experiment

eye movement and muscle artifacts. Epochs with voltage variation exceeding ± 100 μV at any EEG channel were also rejected from further analysis. All responses were then recalculated offline against average reference for additional examination.

2.4 Data Pre-processing and Feature Extraction

The TestBench program was used to collected data. Those collect data were generated as.edf extension file and were then converted into.csv extension file. This file was compatible with Microsoft Excel where we reduced unnecessary tables and make sure that collected data would be compatible with spatial analysis.

2.5 Spatial Analysis

Electrical activities of Karawitan musicians' brains while listening to *Gendhing Lancaran*, Javanese traditional music, were defined as a moment of the global field power and were segmentally filtered for each of frequency bands, e.g., ɗ ɵ α β1 β2 β3 and ɤ, respectively, related to stable scalp potential topography [14]. In the subsequent stage, standardized low-resolution electromagnetic tomography (sLORETA) was applied to estimate the current source density distribution in the brain, which contributed to the electrical scalp field [14]. Maps were then processed with sLORETA. The sLORETA then computed the smoothest of all conceivable source configurations throughout the brain volume by minimizing the total squared Laplacian of source qualities [14].

2.6 Statistical Analysis

While Karawitan musicians listening to *Gendhing Lancaran*, Javanese traditional music, electric activities of the Karawitan musicians' brain were continuously recorded. The smoothest of all conceivable source configurations throughout the brain volume was obtained by minimizing the total squared Laplacian of source qualities. The statistical significance of electrical activities was tested with one sample *t*-test.

3 Results

Electrical activities of Karawitan musicians' brain computed by sLORETA showed that theta (ɵ) wave had highest electrical activity (7.624 ± 0.15 μV, $t(19) = 2.74$; $p < 0.05$) compared to other waves (e.g., delta (ɗ) wave: 5.289 ± 0.47 μV, $t(19) = 2.47$; $p < 0.05$, alpha (α) wave: 2.969 ± 0.52 μV, $t(19) = 1.96$; $p < 0.05$, beta 1 (β1) wave: 1.624 ± 0.81 μV, $t(19) = 2.63$; $p < 0.05$, beta 2 (β2) wave: 5.518 ± 0.04 μV, $t(19) = 2.77$; $p < 0.05$, beta 3 (β3) wave: 1.369 ± 0.61 μV, $t(19) = 2.29$; $p < 0.05$), gamma (ϒ) wave: 7.259 ± 0.74 μV, $t(19) = 2.17$; $p < 0.05$, respectively, while listening to *Gendhing Lancaran*, Javanese traditional music (see Table 1 and Fig. 2).

Source localization analyses were performed using sLORETA [8]. Figure 1 exhibits the *xyz*-values in Talairach space as determined with sLORETA at theta wave while Karawitan musicians tuning into *Gendhing Lancaran,* Javanese traditional music. A single source was estimated to be situated in the middle temporal

Table 1 Electroencephalographic activities of Karawitan musicians' brain while listening to *Gendhing Lancaran*, Javanese traditional music

Frequency Bands	Electroencephalographic activities (μV)
ɗ	5.289 ± 0.47
ɵ	**7.624 ± 0.15**
α	2.969 ± 0.52
β1	1.624 ± 0.81
β2	5.518 ± 0.04
β3	1.369 ± 0.61
ϒ	7.259 ± 0.74

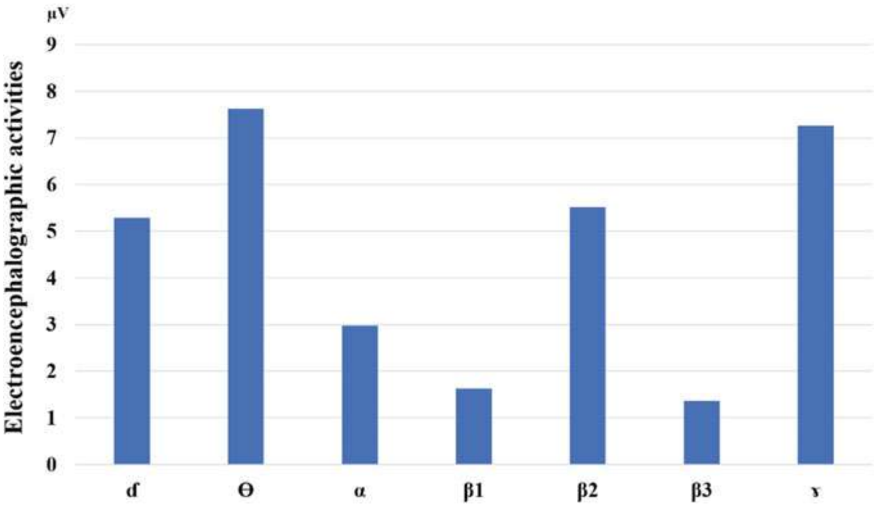


Fig. 2 Electroencephalographic activities (µV) (Mean ± SD) of Karawitan musicians' brain while listening to *Gendhing Lancaran*, Javanese traditional music as computerized by standardized low-resolution electromagnetic tomography (sLORETA)

gyrus (MTG) (Brodmann area 21 (BA21); $x = 65$, $y = -55$, $z = 0$; MNI coords; best match at 5 mm; 7.624 ± 0.15 µV) of right hemisphere (see Fig. 1).

4 Discussion

In Karawitan musician brains, the temporal lobe was the location where predominant frequency band happened when Karawitan musicians tuned into music familiar to them. The temporal lobe was the region of the brain responsible for a memory processing task. It could demonstrate that memory processing happened when Karawitan musicians were tuning into familiar music.

In terms of brain activity, as Karawitan musicians listened to *Gendhing Lancaran*, Javanese traditional music, and Karawitan musicians achieved higher brain activity when listening to *Gendhing Lancaran*, Javanese traditional music. This might be indicated that a higher brain activity occurred when the Karawitan musicians were more familiar with the familiar music, they listened to it. One previous study showed the importance of familiarity instead of liking in increasing brain activity of individuals [15]. Using fMRI toward the subject, the result showed certain emotion-related areas activation during familiar music listening such as the amygdala, putamen, anterior cingulate cortex, and thalamus. It was contributing a marginal effect on brain activation. In the present study, familiar music tends to activate Karawitan musicians' brain in-memory processing area: temporal lobe, specifically in the middle temporal gyrus. Meanwhile, a meta-analysis study [16] showed different results where the location activated by familiar music at left superior frontal gyrus as the most activated area, followed by the ventral lateral as the second-highest area activated by familiar music. The assumption is that the frontal gyrus being activated by the semantic memory of certain familiar music, and meanwhile, the ventral lateral, which is related to the motor cortex, might be stimulated by the motoric response given by the subject when they are anticipating rhythms of familiar music they listened to. In addition, the different results done by Freitas et al. [16] showed ventral lateral as an area activated by familiar music might not happen in our present study because Karawitan musicians were not allowed to make any movement during the listening session to avoid artifacts caused by body movement (Fig. 3).

In terms of musical experience and perception, some previous studies demonstrated through a cultural difference approach [6–8]. One previous study showed that the sequence memory of auditory mechanism was used to promote the early musical experience role in musicians [7]. Musicians, gymnasts, video game players, and psychology students were participated in this previous study. The memory task on audio, visual, and audio-visual stimuli was applied to all participated subjects. The results showed an insignificant difference in the visual or audio-visual task but the significant difference in the audio task where musicians scored higher. Finally, a longitudinal study in early childhood also showed a change of brain structure in children after only fifteen months of musical training [8].

Several previous studies of both EEG BOLD-based studies discovered more prominent initiation to familiar things in the left posterior parietal cortex (lPPC), which increases with the degree of memory of details of those things [17–19]. In any case, an ongoing human intracranial recording investigation [20] revealed the presence of two kinds of memory-specific neurons in the posterior parietal (PPC). These specific neurons expanded their terminating rates for recognizable stimuli, and specific neurons that favored novel stimuli [20]. The presence of memory sensitive cells that react to new things suggests that the coding of "newness" is related to processing that is not quite the same as a simple absence of a familiarity signal [20]. The previous examination utilizing electroencephalography (EEG) and pupillometry to reveal the temporal marks of the brain processes that permit separation between a natural, popular, and new piece of music [21]. This previous examination reported that brain responses measured with EEG demonstrated a later separation among natural and new piece of music from 350 ms post-onset. The cluster pattern identified in the EEG response was similar to that ordinarily found in the classic old/new memory retrieval paradigms. The findings of this previous study proposed that the recognition of brief, randomly presented, music snippets, draws on comparable processes [21]. Consequently, the human auditory system displays a marked sensitivity to recognizable or familiar music [22–27]. The idea of music familiarity intensely depends on long-term memory traces [27], auditory mental imagery [28–31], and is additionally connected to autobiographical memories, particularly for emotionally relevant music [31].

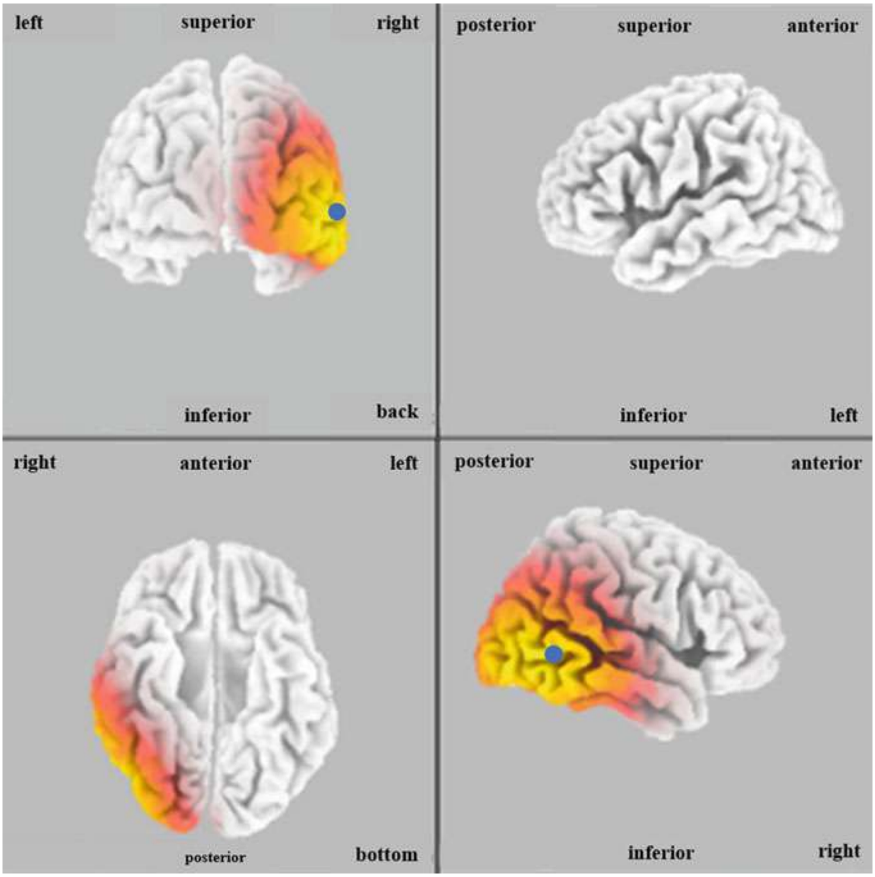


Fig. 3 Graphical representation of the low-resolution electromagnetic tomography (sLORETA) *t*-statistic at the theta wave while Karawitan musicians listening to *Gendhing Lancaran*, Javanese traditional music at middle temporal gyrus (MTG)—temporal lobe (Brodmann area 21; X = 65, Y = −55, Z = 0; MNI coords; best match at 5 mm; 7.62 μV) in the right hemisphere. Red color indicates local maxima of increased electrical activity in right hemisphere (middle-to-back region) through the reference brain. Blue dot marks the center of significantly increased electric activity of theta wave

5 Conclusion

Applying the electroencephalographic recording and source localization techniques, the results showed higher brain activities of Karawitan musicians' brain while listening to familiar music, *Gendhing Lancaran*, Javanese traditional music. The dominant brain activities happened in the temporal lobe. Instead of differentiating brain activity between musician and non-musician, the present study demonstrated the different brain activities while musicians listened to music regarding their musical experiences. This result will evoke music's integration study to further consideration, such as neuroscience of music.

Acknowledgements This study was technically supported by Mae Fah Luang University (electroencephalogram laboratory 2019), and Brain Science and Engineering Innovation Research Group, Mae Fah Luang University, Thailand. We thank all Karawitan musicians who participated in this study.

Contributors
I.K.W. designed the study; conducted the research; obtained funding. D. supervised the research. F.T. supervised the research. P.S. designed the study; analyzed the data; obtained funding; wrote the paper. All authors approved the final version of the paper. The authors declare no competing interests.

References

1. Patston, L.L.M., Tippett, L.J.: The effect of background music on cognitive performance in musicians and nonmusicians. Music. Percept. **29**(2), 173–183 (2011)
2. Husain, G., Thompson, W., Schellenberg, E.: Effects of musical tempo and mode on arousal, mood, and spatial abilities. Music. Percept. **20**(2), 151–171 (2002)
3. Sluming, V., Brooks, J., Howard, M., Downess, J.J., Roberts, N.: Broca's area supports enhanced visuospatial cognition in Ochestral musicians. J. Neurosci. **27**(14), 3799–3806 (2007)
4. Gaser, C., Schlaug, G.: Brain structures differ between musicians and non-musicians. J. Neurosci. **23**(27), 9240–9245 (2003)
5. Schlaug, G.: The brain of musicians. In: Peretz, I., Zatorre, R.J. (eds.) The Cognitive Neuroscience of Music, pp. 366–381. Oxford University Press, New York (2003)
6. Cross, I.: Music, cognition, culture, and evolution. In: Peretz, I., Zatorre, R.J. (eds.) The Cognitive Neuroscience of Music, pp. 42–56. Oxford University Press, New York (2003)
7. Tierney, A.T., Bergeson-Dana, T.R., Pisoni, D.B.: Effects of early musical experience on auditory sequence memory. Empir. Music. Rev. **3**(4), 178–186 (2008)
8. Hyde, K.L., Lerch, J., Norton, A., Forgeard, M., Winner, E., Evans, A.C., Schlaug, G.: The effects of musical training on structural brain development: A longitudinal study. Ann. N. Y. Acad. Sci. **1169**, 182–186 (2009)
9. Greenberg, D.M., Baron-Cohen, S., Stillwell, D.J., Kosinski, M., Rentfrow, P.J.: Musical preferences are linked to cognitive styles. PLoS ONE **10**(7), e0131151 (2015)
10. Bos, D.P.O.: EEG-based emotion recognition: The influence of visual and auditory stimuli. 1–17 (2006)
11. Tanji, K., Suzuki, K., Delorme, A., Shamoto, H., Nakasato, N.: High-frequency gamma band activity in the basal temporal cortex during picture-naming and lexical-decision tasks. J. Neurosci. **25**(13), 3287–3293 (2005)
12. Grahn, J.A., Brette, M.: Rhytm and beat perception in motor areas of the brain. J. Cogn. Neurosci. **19**(5), 893–906 (2007)
13. Gur, R., Turetsky, B.I., Matsui, M., Yan, M., Bilker, W., Hughett, P., Gur, R.E.: Sex differences in brain gray and white matter in healthy young adults: Correlations with cognitive performance. J. Neurosci. **19**(10), 4065–4072 (1999)
14. Pascual-Marqui, R.D.: Standardized low-resolution brain electromagnetic tomography (sLORETA): Technical Details. Methods Find. Exp. Clin. Pharmacol. **24**(Suppl. D), 5–12 (2002)
15. Pereira, C.S., Teixeira, J., Figueiredo, P., Xavier, J., Castro, S.L., Brattico, E.: Music and emotions in the brain: Familiarity matters. PLoS ONE **6**(11), e27241 (2011)

16. Freitas, C., Manzato, E., Burini, A., Taylor, M.J., Lerch, J.P., Anagnostou, E.: Neural correlates of familiarity in music listening: A systematic review and a neuroimaging meta-analysis. Front. Neurosci. **12**, 686 (2018)
17. Klostermann, E.C., Loui, P., Shimamura, A.P.: Activation of right parietal cortex during memory retrieval of nonlinguistic auditory stimuli. Cogn., Affect., Behav. Neurosci. **9**, 242–248 (2009)
18. Sestieri, C., Shulman, G.L., Corbetta, M.: The contribution of the human posterior parietal cortex to episodic memory. Nat. Rev. Neurosci. **18**, 183–192 (2017)
19. Hutchinson, J.B., Uncapher, M.R., Wagner, A.D.: Posterior parietal cortex and episodic retrieval: Convergent and divergent effects of attention and memory. Learn. Mem. **16**, 343–356 (2009)
20. Rutishauser, U., Aflalo, T., Rosario, E.R., Pouratian, N., Andersen, R.A.: Single-neuron representation of memory strength and recognition confidence in left human posterior parietal cortex. Neuron **97**, 209–220.e3 (2018)
21. Jagiello, R., Pomper, U., Yoneya, M., Zhao, S., Chait, M.: Rapid brain responses to familiar vs. unfamiliar music—An EEG and pupillometry study. Scientific Reports **9**, 15570 (2019)
22. Fujioka, T., Trainor, L.J., Ross, B., Kakigi, R., Pantev, C.: Automatic encoding of polyphonic melodies in musicians and nonmusicians. J. Cogn. Neurosci. **17**, 1578–1592 (2005)
23. Trainor, L.J., Marie, C., Bruce, I.C., Bidelman, G.M.: Explaining the high voice superiority effect in polyphonic music: Evidence from cortical evoked potentials and peripheral auditory models. Hear. Res. **308**, 60–70 (2014)
24. Filipic, S., Tillmann, B., Bigand, E.: Erratum to: Judging familiarity and emotion from very brief musical excerpts. Psychon. Bull. Rev. **17**, 601 (2010)
25. Halpern, A.R., Bartlett, J.C.: Memory for melodies. In: Jones, R.M., Fay, R.R., Popper, A.A. (eds.) Music Perception, pp. 233–258. Springer, New York (2010)
26. Schellenberg, E.G., Iverson, P., Mckinnon, M.C.: Name that tune: Identifying popular recordings from brief excerpts. Psychon. Bull. Rev. **6**, 641–646 (1999)
27. Koelsch, S.: Investigating the neural encoding of emotion with music. Neuron **98**, 1075–1079 (2018)
28. Halpern, A.R., ZATORRE, R.J.: When that tune runs through your head: A pet Investigation of auditory imagery for familiar melodies. Cereb. Cortex **9**, 697–704 (1999)
29. Kraemer, D.J.M., Macrae, C.N., Green, A.E., Kelley, W.M.: Sound of silence activates auditory cortex. Nature **434**, 158 (2005)
30. Martarelli, C.S., Mayer, B., Mast, F.W.: Daydreams and trait affect: The role of the listener's state of mind in the emotional response to music. Conscious. Cognition **46**, 27–35 (2016)
31. Janata, P., Tomic, S.T., Rakowski, S.K.: Characterisation of music-evoked autobiographical memories. Memory **15**, 845–860 (2007)

Automatic Detection of Parkinson Speech Under Noisy Environment

R. Janani Jayashree, Sneha Ganesh, Sanjana C. Karanth, and S. Lalitha

Abstract This work primarily aims to automatically detect patients who are suffering from Parkinson's disease (PD) in comparison to the individuals who are healthy, through voice samples under clean and different noisy environmental conditions. The dataset was subjected to colored noises, electronic noises and natural noise. A feature vector comprising seven mean spectral features and two mean temporal features have been extracted. The performance of the PD detection model, configured by different classifiers of K- nearest neighbor (KNN), Extreme Gradient Boost, and Classification and Regression Trees (CART) have been analyzed under varying noisy environments. The proposed model for PD detection offers 97.01% accuracy for noise free dataset with KNN classifier and it also performs optimally even in the presence of varying noises. All colored noise samples gave superior classification accuracy with KNN classifier and all electronic and natural noises gave best accuracy with Extreme Gradient Boost classifier.

Keywords Parkinson's disease · Speech · Noise · K- nearest neighbors (KNN) · eXtreme gradient boost (XGBoost) · Classification and regression trees

1 Introduction

Parkinson's disease is a chronic neurodegenerative disease which affects the central nervous system of the body [1]. Around 10 million people suffer from this disease all over the world [2]. It is more commonly seen in men than women [3]. Detecting Parkinson's disease by the loss of dopamine containing neurons may take much longer as majority of these dopaminergic neurons are generally degenerated at the time of clinical diagnosis [1]. Therefore, early detection of this disease through non-invasive techniques helps the patient retain the remaining number of neurons [2]. The symptoms of PD can be both motor and non-motor. Among the major motor

R. J. Jayashree · S. Ganesh · S. C. Karanth · S. Lalitha (✉)
Department of Electronics and Communication Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Bengaluru, India
e-mail: s_lalitha@blr.amrita.edu

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_16

symptoms of PD, the inability to articulate the phonemes of a speech called Dysarthria and the heavy involuntary aperiodic breathing which reduces the volume and pitch of the voice known as Dysphonia play a major role in speech analysis as these can be observed almost 5 years prior to the clinical diagnosis [4]. These non-invasive methods can be used to suppress nerve impulses and improve the sense of control [2]. Although PD is a type of disease which cannot be cured completely, early detection and appropriate medication may not make the symptoms worse for the patient [5].

This work mainly uses sustained vowel phonations /a/and /o/to analyze the speech symptoms. These phonations make the patient hold a particular frequency for a longer time. This makes identifying hoarseness in the breathing, variation in pitch and variation in voice easy and effective. This also significantly verifies the ability of memory to remember the phonations.

In practical scenarios, collection of voice samples includes noise, which is an unavoidable component in our day-to-day activities. Hence, detection of PD accurately along with the noise in the voice samples is challenging.

In this experimental work, the samples are subjected to:

1. Coloured noise which includes white, pink, brown, grey and violet noises
2. Electronic noise which includes AC noise and Fan Noise
3. Natural noise included is Rain noise

Among the coloured noises, pink noise is the most commonly found noise in audio samples as this noise maintains a balance across the full range of audible frequencies by shifting the energy across frequencies. White noise is used in variety of engineering applications and its ability to cut through the background noise makes it unique. Brown noise has decrease in power density with increasing frequency. All the above three noises can be generated using the Audacity Software easily. Violet noise being the inverse of brown noise has an increase in the power density with increasing frequency. Grey noise has power at the top and bottom of frequency spectrum and has a less energy at the center. Thus, these coloured noises with its unique power spectrum makes their addition to the voice samples important in order to study how accurately PD can be detected under their influence.

Considering the electronic noises used in this study, these kinds of noises are most commonly found in real time scenarios. Although natural noise like rain noise is considered similar to pink noise, it has its own modulation in the frequency. Including them in this work makes the proposed model more relevant to practical situations.

This paper is divided into 5 sections. Section 2 mentions the key points about the work previously reported in this area. Section 3 explains the methodology with which this experiment was carried out. Section 4 points out the experimentation results and Sect. 5 states the future scope.

2 Literature Survey

Considering the samples recorded in a noise free background, Sai Jahnavi et al. derived speech feature set of size 28, which were classified using SVM and Random Forest yielding an performance accuracy of 94.77% [1].

In the work proposed by Shaohua Wan et al., about 23 features were extracted and were classified using 5 different classifiers. The Deep Multilayer Perceptron-10 (DMLP-10) classifier outperformed the other classifiers resulting with an accuracy of 80% [6]. Luca Parisi et al., attained a remarkable accuracy of 100% with only 3 epochs and the computational time being 0.01 s prior to convergence under noiseless condition with the usage of MLP-SVM classifier [2]. Both the works in [2, 6] considers not more than 20 PD patients whereas the healthy individuals are 20 and 28 respectively. In another work proposed by Zhennao Cai1 et al., 22 vocal features were selected and were classified using BFO SVM, Grid SVM, PSO SVM, RF, KLEM classifiers. BFO SVM performed the best with an accuracy of 97.3% [7].

Similarly, work proposed by Achraf Benba et al., mainly aimed at recognizing PD patients from other patients suffering from different other neurological diseases. Among the different classifiers used maximum classification accuracy of 90% was achieved using PLP, followed by RASTA-PLP with 86% and then MFCC with 82% [8].

Athanasios Tsanas et al., proposed detection of PD under the influence of pink noise and white noise. Random forests and SVM Classifiers were used for the task. It was observed that SVM classifier performs better compared to RF. Also, addition of noise has no much effect on the accuracy rate. It was also demonstrated that about 90% accuracy was achieved using a feature subset consisting of eight dysphonia measures [9].

Through the survey conducted, it is observed that various speech features and classifiers have been applied in various works for PD detection. However, very limited work is reported on the task under discussion in the presence of background noise. Hence, this work is focused to put forth a PD detection model that uses a compact feature set of 9 features with the performance validated through the speech samples in noise free background as well as subjected to coloured noises such as grey, brown and violet, electronic noise such as Fan and AC noise and natural noise such as rain noise.

3 Experimental Approach

The proposed work is represented as a flowchart in Fig. 1. The performance of the PD Detection model with various datasets which includes noise free dataset, dataset with added coloured noise, dataset with added electronic noise and dataset with added natural noise has been analyzed. Further, a feature set comprising of 7 spectral and 2

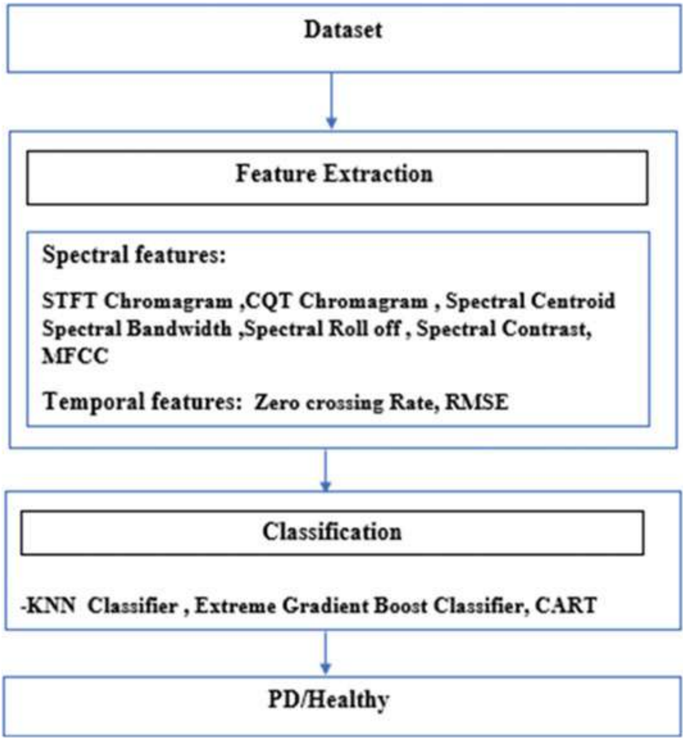

Dataset
Feature Extraction
Spectral features:
STFT Chromagram ,CQT Chromagram , Spectral Centroid Spectral Bandwidth ,Spectral Roll off , Spectral Contrast, MFCC
Temporal features: Zero crossing Rate, RMSE
Classification
-KNN Classifier , Extreme Gradient Boost Classifier, CART
PD/Healthy

Fig. 1 Complete methodical approach of Parkinson's disease detection

temporal features are extracted from each speech dataset followed by classification using the proposed classification techniques.

3.1 Database

The dataset was extracted from UCI machine learning repository and consists of 267 speech samples among which 177 speech samples are that of people suffering from Parkinson's disease and 90 speech samples of healthy controls. All the speech samples extracted are sustained vowels. The speech samples are clean i.e., they are noise-free samples [1].

In addition to the original dataset which consists of speech samples of PD patients and healthy controls, this work furthermore constitutes noise database which comprises three types of noises namely, Coloured noise, Electronic noise and Natural noise.

Five coloured noises are used, they are pink noise, white noise, grey noise, brown noise and violet noise. The electronic noises used are fan noise and Air Conditioner (AC) noises. Rain noise is the Natural noise used for the comparative study.

Pink noise has a frequency spectrum such that the power spectral density is inversely proportional to the frequency of the signal i.e., the spectral power density decreases with 3 dB per octave. White noise has equal intensity per frequency interval i.e., it has a flat frequency spectrum when plotted as a linear function of frequency. Grey noise is random white noise that is subjected to a psychoacoustic equal loudness curve. Grey noise differs from white noise in the power for each frequency. While white noise has equal power for each frequency, grey noise doesn't have equal power for each frequency. Brown noise is a deeper version of pink noise because its spectral power density is inversely proportional to the frequency by 1/f^2 i.e., its spectral power density decreased by 6db per octave. Violet noise is the opposite of brown noise. Its spectral power density is proportional to frequency by f^2 i.e., its spectral power density increases by 6db per octave. Whirring fan noise is similar to white noise. Humming Air Conditioner noise is also similar to white noise. Steady rain noise is similar to pink noise.

3.2 *Feature Extraction*

To detect Parkinson's Disease (PD) and to analyze detection performance under noisy environment from voice samples consisting of sustained phonation of vowels from healthy and PD patients with Dysarthria and Dysphonia symptoms a feature vector comprising 7 mean Spectral features and 2 mean Temporal features have been extracted. The same set of features are derived from the original dataset as well as the dataset with added noises. The feature set includes the following speech features:

3.2.1 Spectral Features

Short Time Fourier Transform (STFT) Chromagram

Combines spectral information from the Short Time Fourier Transform (STFT) of the speech signal that relates to a single pitch class into a single coefficient. 12 Chroma STFT coefficients are obtained from the power spectrogram [10].

Constant Q Transform (CQT) Chromagram

Combines spectral information from the Constant Q Transform (CQT) of the speech signal that relates to a single pitch class into a single coefficient. 12 Chroma CQT coefficients are obtained from the power spectrogram [10, 11].

Spectral Centroid

The Spectral Centroid is a spectral feature that computes the frequency for each frame in the signal at which the spectral energy is centered on [12].

Spectral Bandwidth

Spectral bandwidth gives the weighted mean amplitude difference between frequency magnitude and brightness [12].

Spectral RollOff

Spectral Rolloff represents the fraction of bins in the power spectrum where 85% of its power is at lower frequencies. It is a reliable feature used in the computation of shape of input voice signal [12].

Mel Frequency Cepstral Coefficients (MFCC)

The MFCC of a signal concisely describes the overall shape of the envelope of the short time power spectrum of the speech signal which is the representative of the vocal tract contractions [12–14].

Spectral Contrast

Spectral Contrast depicts the difference in the decibel level between the highest and lowest points in the spectrum of the signal. It accounts for the power variations in the voice sample [12].

3.2.2 Temporal Features

Zero Crossing Rate

Zero Crossing rate is an effective parameter to analyze the noise in an audio sample. This temporal feature is a measure of the number of times in a given time frame the amplitude of the speech signals passes through zero value [12].

Root Mean Square Energy (RMSE)

A measure of loudness of the speech signal. It is defined as

$$\mathrm{RMSE} = \sqrt{\left(\frac{1}{N}\right)\sum_{n}|x(n)|^{2}} \tag{1}$$

All the 9 features have been extracted using Python's Librosa Package.

3.3 Classification

The proposed PD Detection model in the work is modelled with different classifiers i.e. K-Nearest neighbor(KNN) [3, 6, 15], Extreme Gradient Boost [16], Classification and Regression Trees (CART) [17], Logistic Regression, Gaussian Naive Bayes(GNB) [18], Support Vector Machine (SVM) [1, 6], for the classification of feature set extracted from the original as well as dataset with added noises. Performance of these models for different dataset (original and noised dataset) have been analyzed.

From the experimental results, it was observed that KNN, XGBoost and CART performed superior for different datasets under consideration. Hence this work gives a comparative study of the performance of PD Detection model designed based on the stated three classifiers during the detection of PD in noise free and noisy environments.

K-Nearest Neighbor (KNN) Classifier based on supervised learning machine learning algorithm is easy to implement, quite versatile, and offers relatively high accuracy. The input consists of the k closest training examples in the feature space and the output is assignment of a class. An object is classified based on the vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small) [3, 6, 15].

Extreme Gradient Boost Classifier implements optimized Machine Learning Algorithms under Gradient Boosting framework and is flexible and efficient. These algorithms combine many weak learning models together to create a strong predictive model using decision trees [16].

Classification and Regression Tree Classifier (CART) is simple yet powerful and a predictive model obtained by recursive partitioning of data space and fitting a prediction model in each partition. The partitioning can be illustrated as a Decision Tree [17].

3.4 Performance Metric

The performance of the proposed PD model has been evaluated using the performance metric of average accuracy (%) evaluated using Eq. (2) as follows,

$$\text{Average Accuracy}(\%) = \frac{\text{No. of correct predictions}}{\text{Total no. of predictions}} \times 100 \tag{2}$$

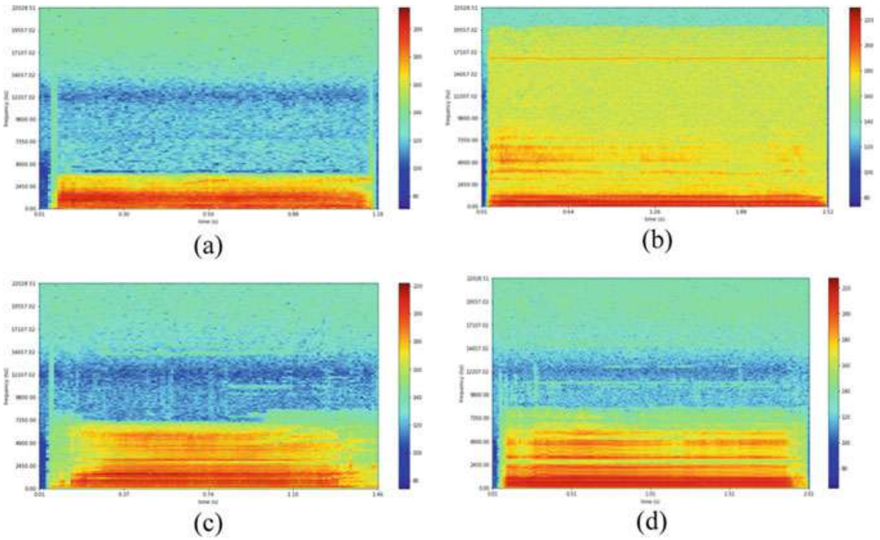
4 Experimental Analysis and Results

Speech spectrograms of male and female patients suffering from PD as well as male and female healthy controls with the utterance of /a/phonation are shown in Fig. 2.

By observing the spectrograms it is observed that the speech spectrograms of female patients suffering from PD have higher frequency distribution when compared with the speech spectrograms of female healthy controls, and the speech spectrograms of male patients suffering from PD have lower frequency distribution when compared with the speech spectrograms of male healthy controls.

Figure 3, depicts the spectrograms of different noises which depict various spectral characteristics across the frequency bands. White noise has uniform energy distribution while violet noise apart from low frequency band, has a similar characteristic as white noise. Rain noise has no energy spread in the mid frequency bands. Pink, grey, brown and fan noise has energy in the low frequency band, while the AC noise has most energy in the low frequency band.

With the observation of the spectral energy across the frequency bands of various noises, the next stage in the experimental work involves with the feature extraction procedure for both clean and noise samples. The feature vector with 7 mean spectral



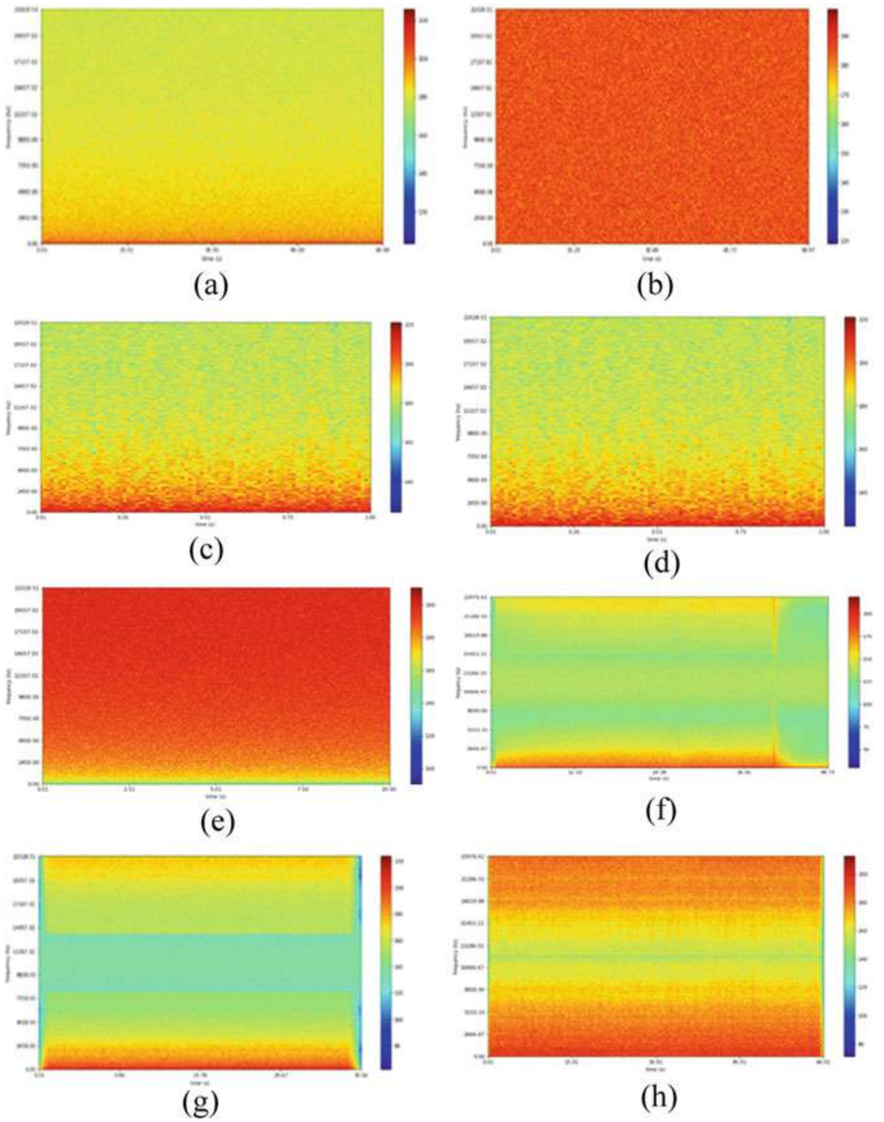
(a)

(b)

(c)

(d)

Fig. 2 Spectrogram representation of voice samples of **a** Female control **b** Male Control **c** Female PD Patient **d** Male PD Patient



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 3 Spectrogram of noises **a** Pink noise **b** White noise **c** Grey noise **d** Brown noise **e** Violet noise **f** Fan noise **g** AC noise **h** Rain noise

features namely STFT Chromagram, CQT Chromagram, Spectral Centroid, Spectral Bandwidth, Spectral Roll Off, MFCC, Spectral Contrast and 2 mean temporal features of RMSE and Zero crossing rate has been extracted from 267 noise free voice samples consisting of 90 healthy and 177 PD speech samples and extrapolated the analysis of Parkinson's speech detection with added noise (−12 dB) to

Table 1 Accuracy (%) analysis of the proposed PD detection model using various classifiers

Dataset	Dataset Named	Accuracy (%)		
		Using KNN	Using XGBoost	Using CART
Noise free Dataset (referred as OD)	Dataset 1	97.01	91.04	89.55
OD + Pink Noise	Dataset 2	80.59	77.61	70.14
OD +White Noise	Dataset 3	83.58	76.11	71.64
OD +Grey Noise	Dataset 4	77.61	77.61	77.61
OD +Brown Noise	Dataset 5	80.59	74.62	70.14
OD +Violet Noise	Dataset 6	92.53	92.53	88.05
OD +Fan Noise	Dataset 7	83.58	89.55	74.62
OD +AC Noise	Dataset 8	86.57	91.04	82.08
OD +Rain Noise	Dataset 9	82.08	83.58	82.08

the above dataset. The noises, of pink noise, white noise, brown noise, grey noise, violet noise, fan noise, air conditioner (AC) noise and rain noise were added to the dataset. Feature vectors derived from these various datasets, have been followed by classification using the three proposed classification techniques, namely KNN Classifier, XGBoost Classifier and Classification and Regression Trees classifier. Post classification. The model is evaluated using 10-fold cross-validation. Feature extraction, classification and cross validation have been done in python platform. All the above-mentioned procedures have been done for both the clean and noisy databases mentioned in Sect. 3.1.

The results obtained with the proposed method using three different classification methods on various datasets are shown in Table 1. Noise free dataset or original dataset (OD) are better recognized with 97.01% using KNN classifier for PD detection.

Results with grey and brown noise are nearly similar around 77−80% probably because of the similarity in the spectral distribution. Also with AC and rain noise, the recognition is above 80% proving the suitability of the model for electronic noises.

From the results of Table 1, a further investigation of the proposed method with PD detection on samples from coloured, electronic and natural noises compared to noise free samples is carried out.

Table 1 depicts the comparative performance of the PD detection model for the original dataset with added coloured noise. An optimal accuracy has been observed with KNN Classifier as in the case of original dataset with added pink noise (80.59%), white noise (83.58%) and brown noise (80.59%) the best accuracy is given by KNN classifier and with grey noise (77.61%) and violet noise (92.53%) best accuracies are given by both KNN and XGB classifiers.

Table 1 also depicts the comparative performance of the PD detection model for the original dataset with added electronic noise and the original dataset with added natural noise. The best accuracies in both original dataset with added fan noise

(89.55%) and AC noise (91.04%) are given by XGB classifier. The best accuracy with respect to the original dataset with added rain noise (83.58%) is given by XGB classifier.

Hence it can be concluded that KNN Classifier outperforms the other classifiers in terms of average accuracy in the detection of PD from noise free dataset as well as for dataset with added coloured noise. For dataset with added electronic noise like Fan and AC noise and natural noise like rain noise XGBoost classifier offers a good accuracy.

4.1 Comparative Analysis with Existing Work on PD Detection

The first comparative analysis has been performed with work of Sai Jahnavi et al., [1] where PD detection is performed with UCI Machine Learning Repository database which is also the database used in the proposed work. The second analysis is done with the work of Athanasios Tsanas et al., [9] where the PD detection involves samples with addition of two noises.

From Table 2, it is observed that the existing work uses a feature set that almost is 3 times the size of the feature set in the proposed work. Additional mechanisms of feature selection is also incorporated. The proposed method achieved around 2.3% increased accuracy compared to the work [1].

The comparative work in Table 3, demonstrates Athanasios Tsanas et al., [9], is validated on a database with a total of 156 speech samples collected from 14 PD patients and only two noises got incorporated in the work followed by a feature selection method. This work involves 8 different noises with no feature selection and an accuracy performance attainment with minimum of 70.14% to a maximum of 92.53%.

Table 2 Comparative analysis of the existing literature and the proposed work using the same database

Parameters	Sai jahnavi et al. [1]	Proposed work
Accuracy	94.77%	97.01%
Size of feature vector	28	9
Feature selection	Correlation based Feature Selection, Best First Selector	NA
Classifiers used	SVM, Random Forest (RF)	KNN, XGBoost and CART

Table 3 Comparative analysis of the work with the dataset added with pink and white noise and the proposed study

Parameters	Athanasios Tsanas et al. [9]	Proposed work
Features	10 Dysphonia measures	7 spectral and 2 temporal features
Feature Selection	LOGO feature selection algorithm used	NA
Classifier	RF and SVM	KNN, XGBoost and CART
Addition of noise	Pink and White noise	1. Coloured noises—pink, white, grey, brown and violet noises 2. Electronic noise—Fan and AC noises 3. Natural noise—rain noise
Accuracy	90%	Coloured noise—70.14−92.53% Electronic noise—74,62−91.04% Natural noise—82−83%

5 Conclusion and Future Scope

This work presents the detection of Parkinson's disease using machine learning algorithms and also compares the classification performance in the presence noise environment. The proposed model is simple with a very compact feature set with no additional preprocessing or feature selection methods applied. Real-time speech signals may contain any of the noises considered in this work. So, it is important to analyze how the presence of noise affects the accuracy of PD detection. The comparison of classification performance of 3 different classifiers has also been done. KNN classifier provides optimized performance results with respect to original dataset and with the dataset with added coloured noise. XGBoost classifiers outperforms in the case of dataset with added electronic noise and dataset with added natural noise.

It would be interesting if the data for detection of PD is collected with an ambient surrounding for the patient, which would give the study a more practical approach. As the dataset used in this work is limited, it could be made larger by adding more number of phonations which includes other sustained vowels and consonants. Additionally, words and numeric characters can also be included. Different feature extraction techniques can also be included. With the larger dataset, deep learning architectures can be implemented. Further, other types of noises can be further added as these play an important role during the detection of PD in real time environments.

References

1. Sai Jahnavi, B., Sai Supraja, B., Lalitha, S.: A vital neurodegenerative disorder detection using speech cues. J. Intell. Fuzzy Syst. (2020)
2. Luca, P., Narrendar, R.C., Marianne, L.M.: Feature-driven machine learning to improve early diagnosis of Parkinson's disease. ELSEIVER- Expert. Syst. Appl. J. (2018)

3. Hamid Karimi, R., Mohammad Reza D.: Diagnosis of Parkinson's disease in human using voice signals. Basic Clin. Neurosci. J. (2011)
4. Athanasios, T.: Novel speech signal processing algorithms for high-accuracy classification of Parkinson's disease. IEEE Trans. Biomed. Eng. (2012)
5. Michal, N., Jan, R., Roman, Č., Evžen, R.: Automatic evaluation of articulatory disorders in Parkinson's disease 1366 IEEE/ACMTransactions On Audio, Speech, And Language Processing, **22**
6. Shaohua, W., Yan, L., Yin, Z., Mohsen, G.: Deep multi-layer perceptron classifier for behavior analysis to estimate Parkinson's disease severity USING smartphones. IEEE Access (25 July, 2018)
7. Zhennao, C., Jianhua, G., Hui-ling, C.: A new hybrid intelligent framework for predicting Parkinson's disease. IEEE Access (19 September, 2017)
8. Achraf, B., Abdelilah, J., Ahmed, H.: Discriminating between Patients with Parkinson's and neurological diseases using Cepstral analysis. IEEE Trans. Neural Syst. Rehabil. Eng. **24**, 10 (2016)
9. Athanasios, T., Max, A.L., Cynthia. F., Lorraine, O.R.: Objective automatic assessment of rehabilitative speech treatment in Parkinson's disease. IEEE Trans. Neural Syst. Rehabil. Eng. **22**, 1 (2014)
10. Shah, A., Kattel, M., Nepal, A., Shrestha, D.: Chroma feature extraction (2019)
11. Manzo-Martínez, A., Camarena-Ibarrola, A.: A robust characterization of audio signals using the level of information content per Chroma. 2011 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Bilbao (2011), pp. 212–217, https://doi.org/10.1109/isspit.2011.6151562
12. Ahmet, E., Hilmi, B.Ç., Mehmet, E.İ., Berkay, Ö., Nizamettin, A.: Music genre classification and recommendation by using machine learning techniques. 2017 25th Signal Processing and Communications Applications Conference (SIU) (May, 2017)
13. Lalitha, S., Tripathi, S., Gupta, D.: Enhanced speech emotion detection using deep neural networks. Int. J. Speech Technol. **22** (2018). https://doi.org/10.1007/s10772-018-09572-8
14. Sreeram, L., Gupta, D.: An encapsulation of vital non-linear frequency features for various speech applications. J. Comput. Theor. Nanosci. **17**, 303–307 (2020). https://doi.org/10.1166/jctn.2020.8666
15. Krishna, P., Reddy, R., Narayanan, Veena, Sreeram, Lalitha, Gupta, Deepa: Affective state recognition using audio cues. J. Intell. & Fuzzy Systems **36**, 1–8 (2019). https://doi.org/10.3233/JIFS-169926
16. Iyad Lahsen, C., Abdesselem, K.: IEEE On using eXtreme gradient boosting machine learning algorithm for home network traffic classification (June, 2019)
17. Zhang, B., Wei, Z., Ren, J., Cheng, Y., Zheng, Z.: An empirical study on predicting blood pressure using classification and regression trees. IEEE Access **6** (2018)
18. Jebakumari, V.S., Shanthi, D., Sridevi, S., Meha, P.: Performance evaluation of various classification algorithms for the diagnosis of Parkinson's disease. 2017 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), Srivilliputhur (2017)

Voice Conversion Using Spectral Mapping and TD-PSOLA

Srinivasan Kannan, Pooja. R. Raju, R. Sai Surya Madhav, and Shikha Tripathi

Abstract In this paper, we propose a novel approach for a voice conversion system that makes effective use of spectral characteristics and excitation information, to optimally morph voice. This work addresses some key issues that are not adequately addressed in reported literature and achieves a more holistic voice conversion system. This is achieved using a strategic combination of line spectral frequencies (LSFs) to minimize the effects of over smoothing, a neural network for performing nonlinear spectral mapping and time-domain pitch synchronous overlap add to account for the interaction of excitation signal with the vocal tract. Within this proposed system, two different methods of pitch modification have been suggested, and the performance of these is compared with existing models of comparable complexity. The proposed methods have an average LSF performance index of 0.4082 and 0.4008, respectively, which is higher than existing similar work reported.

Keywords LSFs · Spectral modification · Neural network · Pitch modification · Time domain pitch synchronous overlap add (TD-PSOLA)

S. Kannan · Pooja. R. Raju · R. S. S. Madhav
Department of Electronics and Communication Engineering, PESIT Bangalore South Campus, Bangalore, India
e-mail: srinivasankannan98@gmail.com

Pooja. R. Raju
e-mail: pooja.rraju98@gmail.com

R. S. S. Madhav
e-mail: saisurya1603@gmail.com

S. Tripathi (✉)
PES University, Bangalore, India
e-mail: shikha.eee@gmail.com

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_17

1 Introduction

Speech has been a prominent way of communication among humans. Human speech conveys not only a message, but also provides an insight into the identity of the speaker. Speaker identity can be characterized using different linguistic, segmental and supra-segmental features. Voice conversion is a technique under voice manipulation that involves transformation of a given source speaker's utterance in such a way that it sounds like an intended target speaker, without losing any language details. This paper describes a voice conversion system which accomplishes morphing of a given utterance by manipulating segmental level features.

Voice conversion, which is also known as voice morphing, has large number of applications in commercial as well as industrial sectors. Voice conversion frameworks can play a significant role in personalized and natural sounding TTS synthesis techniques for use in E-Book reading apps and E-Readers. It is of great importance in the medical field, for patients who have lost, or, are on the verge of losing their voice due to vocal disorders like laryngeal cancer, by helping them to substitute their lost speech. It can also aid in concealing a person's voice for security reasons or to even run spoofing attacks for verifying robustness of speaker verification systems. It has immense scope in the entertainment industry for applications like movie dubbing, virtual voice generation for animation movies and video games. Parametric speech synthesis, one of the popular techniques used in voice conversion, is employed in the proposed system. Linear predictive coding (LPC) parameters, mel-frequency cepstral coefficients (MFCCs), and line spectral frequencies (LSFs)/line spectral pairs (LSPs) are few of the well-known parameters that can be used. The linear predictive coding technique, which gives a set of predictor coefficients represented by linear prediction coefficients (LPCs) as well as estimates of the excitation source, has been considered for the parameterization in this work, because research shows that using LPCs/LSPs, instead of MFCCs results in better performance for voice conversion [9, 12], there is also an advantage of relative simplicity of computation. Along with the parameterization method, the choice of mapping technique also plays a vital role in achieving a reliable conversion model. Some of the prominent techniques include codebook-based approach of vector quantization and spectrum mapping [1, 2], vector field smoothing [5], using hidden Markov models (HMMs) [6], Gaussian mixture models (GMMs) [7, 13], and artificial neural networks (ANNs) [4, 7, 11, 13, 14]. Artificial neural networks have been used for mapping in this work, as they have been shown to outperform GMM- and HMM-based voice conversion models [7, 13], with respect to the intelligibility and quality of the converted voice. Overall, an effective combination of LSFs, ANN, and TD-PSOLA is used for performing nonlinear spectral mapping and alteration of excitation characteristics. The key contributions of the paper are the proposal of a holistic voice conversion system that accounts for the effects of interaction of the excitation source with the vocal tract, by transforming excitation features along with spectral features and the use of computationally efficient signal processing algorithms and a relatively simple feedforward neural network architecture. The manipulation of excitation features is done using efficient time-domain algorithms, which reduces the artifacts in the generated speech. Additionally, the LPC order used for parameterization is varied based on the type of conversion, and an optimal neural network architecture is chosen according to it.

This paper is organized as follows. Section 2 briefly outlines the methods proposed in the reported literature. Section 3 describes the proposed methodologies for the voice conversion framework. Section 4 details the experimental setup used for testing and validating the proposed model, and Sect. 5 illustrates the performance of the proposed model with results of both subjective and objective tests. The final section serves as the conclusion of this paper and also provides an insight into the future scope of this work.

2 Literature Review

Abe et al. [1, 2] were one of the first to research voice conversion and did pioneering work by employing a combination of spectrum mapping and vector quantization, in a codebook-based approach for voice conversion. The authors of [4] were among the initial researchers who experimented with the use of neural networks for converting formants of a source speaker to that of a target speaker, but did not provide adequate results to validate effectiveness of the conversion model. The authors in [7, 13] conducted comparative studies on performance of voice conversion models based on ANN, GMM, and PSOLA and concluded that ANN-based models performed better, especially, in the presence of noise. Avarachan et al. [13] describe a voice conversion method that uses wavelet decomposed pitch residuals for mapping excitation features. The papers [9, 11] and [12] propose different variants of neural network-based voice conversion models. The authors of [14] proposed a neural network-based model that also made use of pitch information to augment the conversion process. Over the years, extensive research has been done in concatenative speech synthesis-based voice conversion models [1, 2, 5, 6], which require an appreciable amount of training data to create reasonably good conversion frameworks. Models that make use of MFCCs are quite popular [7, 9, 10], but the use of MLSA filters often result in artificial sounding voice. Further, many of the recently proposed voice conversion systems make use of computationally intensive machine learning algorithms and deep neural networks [8, 10], which also need a substantial amount of training data to prevent overfitting. As an improvement over existing models, we propose a voice conversion system that makes use of computationally efficient signal processing algorithms and a relatively simple feedforward neural network architecture. While the reported works mainly considered transformation of filter features, this work also considers modification of excitation features in the time domain, using two different approaches, with the help of the simple, reliable and efficient TD-PSOLA algorithm. Thus, the proposed system comprehensively takes care of the major aspects pertaining to ensuring good quality voice conversion.

3 Proposed Methodologies

The voice conversion system described here is based on the source filter discrete time system model of speech production, in which the sampled speech is separated into two components that are expected to be independent of each other. This separation is done by representing the speech signal as the response of a time varying filter to an excitation signal. Nonlinear spectral mapping using a neural network does a good job in transformation of the filter parameters while modification of the pitch contour using TD-PSOLA gives good results in the transformation of the excitation component. Since each of these two techniques primarily addresses the transformation of only one of the two components, in a complimentary manner, they have been combined in this model. The voice conversion model has two distinct sequential phases, viz., a training phase followed by a transformation phase. In the training phase, speaker dependent attributes represented by LPCs are extracted from several pairs of parallel training utterances spoken by a pair of speakers, using linear prediction analysis. The LPCs are then converted to LSPs and time aligned using dynamic time warping [7, 13], before training the neural net. This is done in order to reduce the number of unstable synthesis filters (generated after modification of the parameters using an ANN) and to account for any differences in timing and speed of pronunciation between a pair of speakers, respectively. The capability of ANNs to discern nonlinear relationships is exploited to establish a suitable mapping function between the vocal tract parameters of a pair of speakers. In addition to this, average fundamental frequency information from multiple pairs of training utterances is also used to develop a logarithm Gaussian normalization transformation model. In the transformation phase, manipulation of the vocal tract filter parameters (spectral modification) is done by modifying the extracted LSPs using a trained neural net. Additionally, the excitation source signal is also modified using the log linear transformation model developed in the training phase and the time domain pitch synchronous overlap add (TD-PSOLA) algorithm [3]. Two methods have been proposed for the pitch manipulation. The algorithms used in both methods are based on the source-filter speech production model and the LPC analysis and synthesis framework. The LPC framework was used for the separation of the spectral and excitation features, because of its accuracy in parameterization and also because it comparatively minimizes the complexity involved in homomorphic filtering. LPC analysis parameterizes the excitation source as well as the time-varying filter representing the vocal tract, by separating them into residual signals and LP coefficients, respectively. In the first method, the excitation signal for each frame is initially extracted using LP analysis and combined using overlap add, after which, the pitch contour of the combined excitation signal is modified according to the log linear transformation model, using TD-PSOLA. In the second method, the pitch contour of the test utterance is directly manipulated after which the LSPs are changed. Finally, the modified LSPs and excitation signal are used to synthesize the converted speech. The steps involved in both methods are explained further below.

3.1 Method 1

The flow of this method is illustrated in Fig. 1.

Training Stage
A pair of training utterances are subjected to pre-emphasis, frame blocking (frame size of 30 ms), and windowing. Frame blocking is carried out in such a manner that the successive frames have an overlap of 66% corresponding to a 20 ms overlap, and each frame is weighted using a Hanning window. LPC analysis using LPC prediction orders of either 16, 20, or 24 is carried for each windowed frame. It has been found empirically that an order of 20 gives better results for female to male voice conversion while an order of 16 gives better results for male to female voice conversion. For all the cases of voice conversion, an order of 24 will give reasonably good results.

The LP analysis step generates a set of predictor coefficients and a residual signal for each frame. The LP coefficients are converted to line spectral pairs (LSPs) in order to reduce the number of unstable synthesis filters, in the conversion stage and also to better preserve spectral details after conversion. Further, these LSPs extracted from a pair of source and target parallel utterances are time aligned using dynamic time warping in order to minimize distortion that may occur due to differences in timing and speed of pronunciation. A feed forward neural network with one hidden layer is then trained using these time aligned LSPs. The predictor order chosen in LP analysis determines the number of neurons in the hidden layer, as illustrated in Table 1. The scaled vonjugate gradient algorithm which is a supervised learning

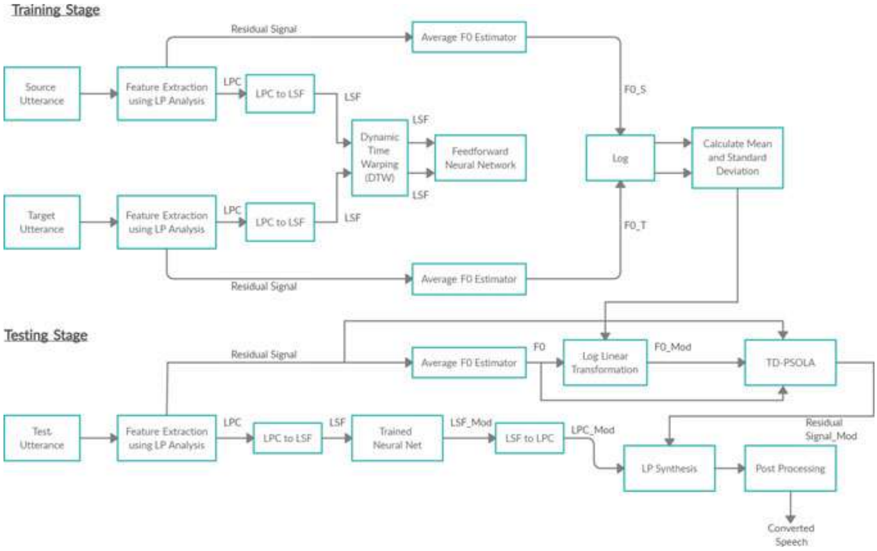


Fig. 1 Block diagram of method 1

Table 1 No. of neurons for different LPC orders

LPC order	No. of neurons
16	27
20	34
24	50

algorithm is used to train the neural network, to approximate a mapping function from the source speaker's LSP vector space to that of the target speaker (spectral mapping).

Finally, the residual signals of each frame are combined using overlap add, and the autocorrelation method of pitch estimation is used to find the average fundamental frequency of the combined residual signal, for multiple pairs of training utterances. A normalized Gaussian logarithmic transformation model (log linear model) is then developed using measures of mean and standard deviation of the average fundamental frequency of the residual signals in the log domain [7].

Testing Stage

The test utterance is subjected to pre-emphasis, frame blocking, and windowing after which LP analysis is carried with the parameters same as in the training stage. The obtained LP coefficients are further converted to LSPs, and the residual signals of each frame are combined using overlap add.

Spectral modification is performed by altering the LSPs using the trained neural neural. The altered LSPs of each frame are arranged in an ascending order, and if the LSPs of any frame are found to lie outside the range $[0, \pi]$, the respective set is normalized and scaled to lie in the range $[0, \pi]$. The modified LSPs are then converted back to LP coefficients. Although the LPCs are indirectly altered by first modifying the equivalent LSPs, some synthesis filters still remain unstable. This issue is resolved by identifying the filter poles that lie outside the unit circle in the Z-Plane and reflecting them back into it.

The residual signals of all the frames are combined using the overlap add method. The average fundamental frequency 'F0' of the combined signal is then estimated using the auto-correlation method. The log linear transformation model developed in the training stage uses this information to calculate the expected average fundamental frequency of the morphed residual signal according to the equation

$$\log(f0_{\text{conv}}) = \mu_{\text{tgt}} + \frac{\sigma_{\text{tgt}}}{\sigma_{\text{src}}}(\log(f0_{\text{src}}) - \mu_{\text{src}}) \tag{1}$$

where σ_{src} and μ_{src} are the variance and mean of the logarithmic representation of the fundamental frequencies for the source speaker, $f0_{\text{src}}$ is the average fundamental frequency of the source speaker, and $f0_{\text{conv}}$ is the expected average converted fundamental frequency of the target speaker [7]. Pitch contour of the residual signal is then modified using TD-PSOLA [3] such that the new average fundamental frequency matches $f0_{\text{conv}}$. The modified residual signal is then framed and windowed to create residual signals for each frame. The modified residual signals for each frame are then filtered using the corresponding modified LPCs as poles of the synthesis filter, to generate the frames of the converted speech. These are then combined using overlap add to get the morphed utterance.

3.2 Method 2

The flow of this method is illustrated in Fig. 2. The differences in the the steps over method 1 are highlighted below.

Training Stage
Average F0 is estimated directly from a pair of training utterances. The pitch contour of the source utterance is then modified to match that of the target utterance, using this information with the help of the TD-PSOLA algorithm. LP analysis is done similar to method 1, and the time aligned LSP parameters are used to train the neural net with the same parameters as method 1. The estimated average F0s of a pair of speakers is converted to the equivalent log representation. Mean and standard deviation measures are then calculated using the data from multiple training utterances, which is used to develop the log linear transformation model.

Testing Stage
Average F0 of the test utterance is estimated and used to calculate the expected average F0 of the converted utterance, using log linear transformation. This information

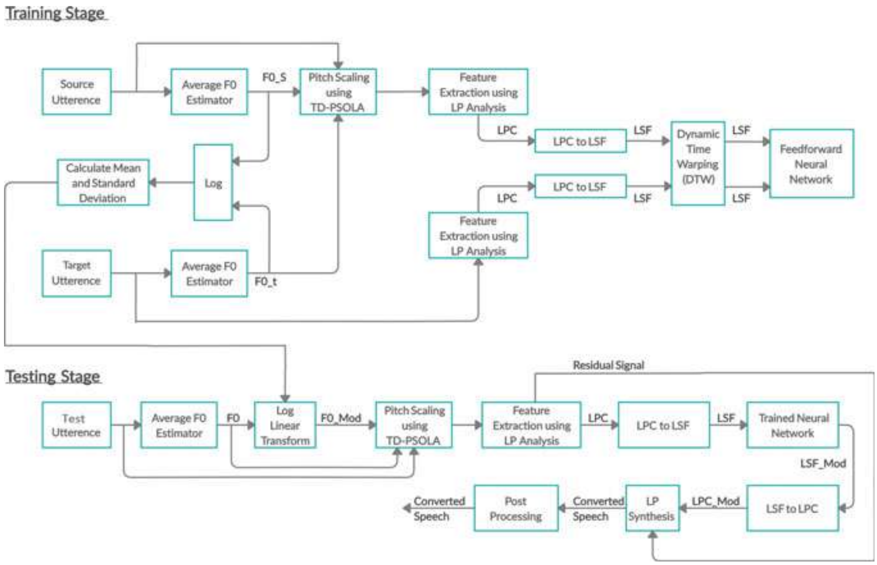


Fig. 2 Block diagram of method 2

is then used to perform pitch contour modification on the test utterance, using TD-PSOLA. The procedure for spectral modification remains the same as in method 1. The residual signals of each frame are left unchanged and are filtered using the modified LPCs as poles of the synthesis filter, to reconstruct the converted utterance (LP synthesis).

3.3 Summary

The first method proposed provides an intuitive way to address the modification of excitation features along with spectral modification since it separates the speech signal into its spectral and excitation components. The LSPs of each frame which are a good representation of the spectral features are modified using a neural network. The modification of the excitation features is done by manipulating the pitch contour of the combined excitation signal since the effects of formants are filtered out in this signal. This leads to better pitch estimation and consequently to better pitch contour manipulation, especially, since the modification of the spectral features is done on the time domain waveform (TD-PSOLA).

The second method uses a form of indirect pitch manipulation similar to the approach taken in [14], which relies on lesser number of overlap adds than method 1 (lesser reconstruction perturbations or artifacts). The pitch contour of the main signal is directly modified before the spectral modification, which indirectly leads to the modification of the excitation features after filtering of the untouched residual signals using the modified LPCs, in the conversion stage. The pitch contour manipulation is carried out in the time domain, in order to reduce the number of artifacts in the converted speech and also to better preserve spectral details.

Both the proposed methods perform well in a voice conversion framework and generate good results, as illustrated by both subjective and objective tests. The converted speech generated using method 2 typically has lesser artifacts and sounds clearer, because of lesser number of overlap add reconstructions. But, the characteristics of the voice of the target speaker are better represented in the voice generated using method 1. Method 1 also performs slightly better for male to female voice conversion while method 2 performs better for female to male voice conversion.

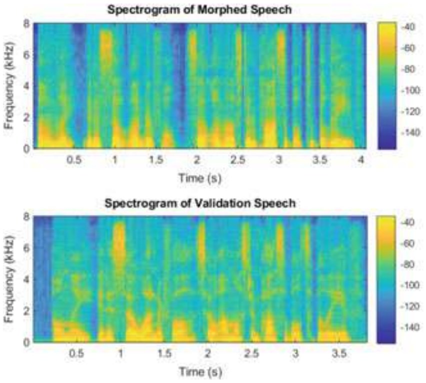
4 Experimental Setup

The proposed framework was tested using speech samples from the CMU arctic database, which consists of 1132 phonetically balanced utterances of 7 different speakers (both male and female), recorded at a sampling frequency of 16000 Hz. Training the proposed framework for about two and a half minutes with at least 30 seconds of parallel training data produces a reasonably good voice conversion framework. Experimentation on the framework was carried out using speech samples from two speakers BDL (US male) and SLT (US female) of the CMU arctic database. 50 parallel utterances of approximately 3 seconds each were used to train the model.

A feedforward neural network with a tangent sigmoid transfer function in the hidden layer and a mean squared error performance function was used for spectral mapping. The model was trained for 7 min. Both methods of conversion were tested using different orders of LPC.

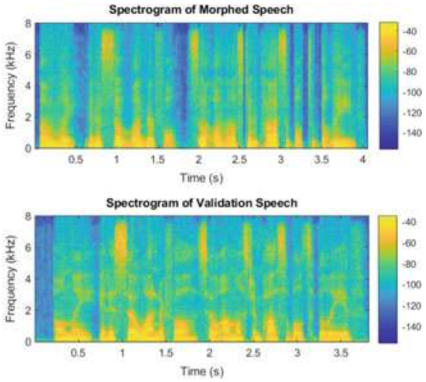
5 Results and Performance Evaluation

The results are explained further below and are illustrated using the spectrograms shown for the various test cases. All the spectrograms correspond to the sentence "Only, it is so wonderful, so almost impossible to believe." The performance of the proposed methods has been evaluated using both objective and subjective tests. The results are discussed further below and also compared with that of the earlier approaches having similar complexity.


Spectrogram of Morphed Speech

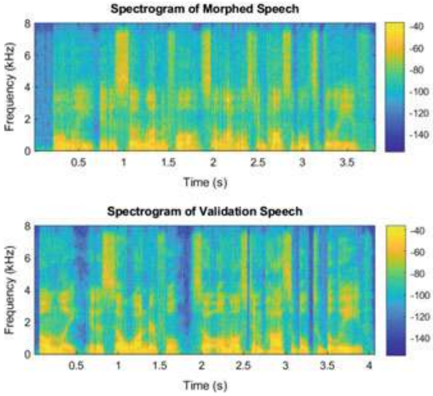
Spectrogram of Validation Speech

Spectrogram: Method 1; BDL to SLT (Male to Female); LPC order-16


Spectrogram of Morphed Speech

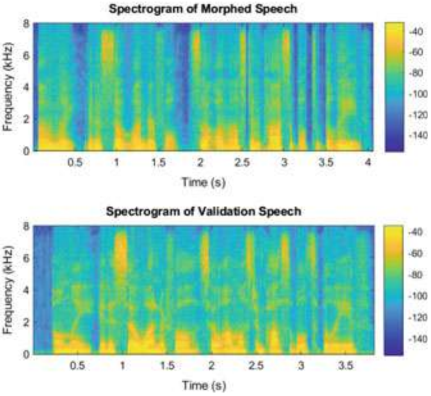
Spectrogram of Validation Speech

Spectrogram: Method 2; BDL to SLT (Male to Female); LPC order-16


Spectrogram of Morphed Speech

Spectrogram of Validation Speech

Spectrogram: Method 1; SLT to BDL (Female to Male); LPC order-20


Spectrogram of Morphed Speech

Spectrogram of Validation Speech

Spectrogram: Method 2; SLT to BDL (Female to Male); LPC order-20

5.1 Performance Evaluation

Objective evaluation metric: LSF performance index LSF performance is evaluated for a set samples as part of the objective assessment. The LSF performance index is given by

$$P_{\text{lsf}} = 1 - \frac{D_{\text{lsf}}(d(n), t(n))}{D_{\text{lsf}}(d(n), s(n))} \tag{2}$$

where $d(n)$ and $s(n)$ represent desired speaker and source speaker utterances, respectively, while $t(n)$ is the converted utterance. $D_{\text{lsf}}(d(n), t(n))$ represents the spectral distortion between desired and transformed utterances, and $D_{\text{lsf}}(d(n), s(n))$ evaluates inter spectral distortion between speakers. $D_{\text{lsf}}(u, v)$ is given by

$$D_{\text{lsf}}(u, v) = \frac{1}{N} \sum_{i=1}^{N} \sqrt{\frac{1}{p} \sum_{j=1}^{p} (\text{LSF}_u^{i,j} - \text{LSF}_v^{i,j})^2} \tag{3}$$

where N is the number of frames, p is the order of LSF, and $LSF_u^{i,j}$ is the j_{th} LSF component in the i_{th} frame. When $P_{\text{LSF}} = 1$, it indicates that the conversion is perfect, whereas $P_{\text{LSF}} = 0$ indicates that the conversion has completely failed (Tables 2 and 3).

Table 2 Comparison of LSF performance index of both methods

Conversion	Pre-emphasis	LPC order	Method 1	Method 2
BDL to SLT (male to female)	Yes	16	0.4416	0.4309
		24	0.4216	0.3975
SLT to BDL (female to male)	Yes	20	0.3911	0.3924
		24	0.3783	0.3823

Table 3 Comparison with PLSF of existing models

Approach	PLSF
Method 1	0.40815
Method 2	0.400775
Bhuyan et al. (2015) [9]	0.28415
Ayodeji et al. (2016) [11]	0.39945
Yathigiri et al. (2017) [14]	0.15425

Subjective evaluation metrics

1. **ABX Test**
The listeners are presented with the source, target, and the converted speech samples. They asked to indicate whether the converted speech sounds closer to the natural utterance of the source speaker or to that of the target speaker.
2. **Mean Opinion Scoring (MOS) Test**
The listeners are asked to evaluate the speech quality of the converted sample using a five-point scale which indicates the following:
5—Excellent; 4—Good; 3—Fair; 2—Bad; 1—Poor (Tables 4 and 5).

It has been found that method 1 typically performs better when it comes to accurate conversion of spectral details, as indicated by its superior PLSF scores; thus, the converted voice generated using method 1 will have a better representation of the target speaker's voice characteristics. Method 2 on the other hand produces cleaner converted speech with lesser artifacts and perturbations (due to lesser number of overlap add reconstructions), which is evident from its better subjective evaluation metrics (MOS and ABX test results). For any arbitrary voice conversion case, in order to achieve a higher quality of voice conversion, method 1 is recommended, as in this method, the modeling and subsequent modification of excitation features is typically more accurate. This is because the frequency of erroneous pitch estimations is reduced by filtering out the effects of the formants. But, since both techniques have

Table 4 Subjective evaluation results

Conversion	Pre-emphasis	LPC order	Method 1		Method 2	
			MOS	ABX (%)	MOS	ABX (%)
BDL to SLT (male to female)	Yes	16	3.36	86	3.5	84
		24	2.94	78	3.46	82
SLT to BDL (female to male)	Yes	20	3.28	84	3.84	92
		24	3.12	82	3.7	88

Table 5 Comparison with subjective evaluation results of existing models

Approach	MOS	ABX
Method 1	3.175	82.5%
Method 2	3.625	86.5%
Desai et al. (2009) [7]	2.549	_
Bhuyan et al. (2015) [9]	3.05	81.56%
Yathigiri et al. (2017) [14]	_	85%

their merits, the best performance of the voice conversion framework can be achieved by selecting the LPC parameters and the conversion method at runtime, according to the conversion case. Both the methods of the proposed system use optimal neural network architectures for spectral mapping which leads to better performance than other reported approaches, as seen from the higher PLSF scores. Comparison of the subjective evaluation results with the ones available in the reported literature indicates that the quality of the converted voice generated using the proposed system is also perceptually better.

6 Conclusion

In this paper, a voice conversion system has been proposed which makes use of pitch and spectral information. A neural network is employed for non-linear spectral mapping, and TD-PSOLA is used for implementing pitch manipulation (for which two different methods have been proposed), to enhance the quality of conversion. The proposed methods are computationally efficient and show better conversion performance as compared to existing models, which is evidenced by their respective average PLSF scores of 0.4082 and 0.4008. They also obviate any need for careful preparation of training data; minimal training time and data are required for the proposed models, since a simple neural network architecture is used.

The paper addresses some of the problems overlooked previously and shows better performance than the models described in the reported works, but it still has some limitations which are primarily the requirement for parallel training data as well as the preference that both the source and target speakers have similar accents. Future work can be done in removing the need for parallel training data and in using speaker adaption techniques on readily available pre-trained models, to adapt for new pairs of speakers. Additional work can be done in enhancing the naturalness of the converted speech by considering more prosodic characteristics and in developing zero shot algorithms.

References

1. Abe, M., Nakamura, S., Shikano, K., Kuwabara, H.: Voice conversion through vector quantization. In: International Conference on Acoustics, Speech, and Signal Processing, ICASSP-88. New York, NY, USA, vol.1, pp. 655–658 (1988). https://doi.org/10.1109/ICASSP.1988.196671
2. Abe, M.: A segment-based approach to voice conversion. In: [Proceedings] ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing, Toronto, Ontario, Canada, vol. 2, pp. 765–768 (1992). https://doi.org/10.1109/ICASSP.1991.150451
3. Valbret, H., Moulines, E., Tubach, J.P.: Voice transformation using PSOLA technique. In: [Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing, San Francisco, CA, USA, vol. 1, pp. 145–148 (1992). https://doi.org/10.1109/ICASSP.1992.225951

4. Narendranath, M., Murthy, H.A., Rajendran, S., Yegnanarayana, B.: Transformation of formants for voice conversion using artificial neural networks. Speech Commun. **16**(2), 207–216 (1995). https://doi.org/10.1016/0167-6393(94)00058-I
5. Hashimoto, M., Higuchi, N.:Training data selection for voice conversion using speaker selection and vector field smoothing. In: Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP '96, Philadelphia, PA, USA, vol. 3, pp. 1397–1400 (1996). https://doi.org/10.1109/ICSLP.1996.607875
6. Kim, E.-K., Lee, S., Oh Y.-H.: Hidden Markov model based voice conversion using dynamic characteristics of speaker. In: EUROSPEECH-1997, pp. 2519–2522 (1997)
7. Desai, S., Raghavendra, E.V., Yegnanarayana, B., Black, A.W., Prahallad, K.: Voice conversion using artificial neural networks. In: 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, pp. 3893–3896 (2009). https://doi.org//ICASSP.2009.4960478
8. Mohammadi, S.H., Kain, A., Voice conversion using deep neural networks with speaker-independent pre-training. In: IEEE Spoken Language Technology Workshop (SLT). South Lake Tahoe, NV, vol. 2014, pp. 19–23 (2014). https://doi.org/10.1109/SLT.2014.7078543
9. Bhuyan, A.K., Nirmal, J.H.: Comparative study of voice conversion framework with line spectral frequency and Mel-frequency cepstral coefficients as features using artificial neural networks. In: 2015 International Conference on Computers, Communications, and Systems (ICCCS), Kanyakumari, pp. 230–235 (2015). https://doi.org/10.1109/CCOMS.2015.7562906
10. Luo, Z., Takiguchi, T., Ariki, Y.: Emotional voice conversion using deep neural networks with MCC and F0 features. In: 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), Okayama, pp. 1–5 (2016). https://doi.org/10.1109/ICIS.2016.7550889
11. Ayodeji, A.O., Oyetunji, S.A.: Voice conversion using coefficient mapping and neural network. In: International Conference for Students on Applied Engineering (ICSAE). Newcastle upon Tyne, vol. 2016, pp. 479–483 (2016). https://doi.org/10.1109/ICSAE.2016.7810239
12. Koolagudi, S.G., Vishwanath, B.K., Akshatha, M., Murthy, Y.V.S.: Performance analysis of LPC and MFCC features in voice conversion using artificial neural networks. In: Satapathy, S., Bhateja, V., Joshi, A. (eds.) Proceedings of the International Conference on Data Engineering and Communication Technology. Advances in Intelligent Systems and Computing, vol. 469. Springer, Singapore (2017)
13. Avarachan, A., Vekkot, S., Tripathi, S.: Glottal excitation and spectral mapping for voice conversion. In: International Conference on Signal and Image Processing (ICSIP 2017) (2017)
14. Yathigiri, A., Bathula, M., Kothapalli, S., Vekkot, S., Tripathi, S.: Voice transformation using pitch and spectral mapping. In: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, pp. 1540–1544 (2017). https://doi.org/10.1109/ICACCI.2017.8126060
15. Mitra, S.K.: Digital Signal Processing: A Computer-Based Approach. McGraw-Hill, New York (2013)
16. Rabiner, L.R., Schafer, R.W.: Theory and Applications of Digital Speech Processing. Pearson Publications (2010)
17. McLoughlin, I.: Applied Speech And Audio Processing With MATLAB Examples. Cambridge University Press, Cambridge (2009)

Haze Removal Using Generative Adversarial Network

Amrita Sanjay, J. Jyothisha Nair, and G. Gopakumar

Abstract The problem of haze removal has been addressed in many computer vision researches. Haze removal is the process of eliminating the degradation present in hazy images and getting the clearer counterpart. The presence of haze distorts the image, and as a result, it will be difficult to apply various image processing techniques on such images. The challenging aspect in haze removal arises due to the lack of depth information in images degraded by haze. The earlier methods for haze removal include various hand-designed priors, usage of the atmospheric scattering model or estimation of the transmission map of the image. The limitation with these models is that they are heavily dependent on the assumption of a good prior. In recent years, various models have been proposed which effectively remove the degradation caused by haze in images using various convolutional neural network architectures. This paper reviews a model which performs haze removal on a single image using generative adversarial network (GAN). The main advantage of this method is that it does not require the transmission map of the image to be explicitly calculated. The model was evaluated using NYU depth dataset and 0-Haze dataset. The model was able to significantly enhance the quality of the images by generating the corresponding haze-free counterpart. The model was evaluated using the peak signal-to-noise ratio and structural similarity index.

Keywords Generative adversarial network · Dehazing

1 Introduction

Haze is an atmospheric phenomenon in which particles in the atmosphere such as dust, smoke and fog obscure the clarity of the sky and cause dispersion of light in the atmosphere. As a result, when images are captured, the image quality drastically drops due to scattering of light randomly affecting all the pixels in the image. When

A. Sanjay · J. J. Nair (✉) · G. Gopakumar
Amrita Vishwa Vidyapeetham, Amritapuri, Kerala, India
e-mail: jyothishaj@am.amrita.edu

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_18

images are affected by haze, it will be difficult to apply computer vision applications such as feature detection, segmentation and object recognition on such images.

Haze removal is a very important step before applying computer vision applications. The advantages of removing haze from images are multifold. Haze removal increases the visibility of the scene making the image more visually pleasing. Most of the computer vision algorithms work under the assumption that application of radiometric calibrations on input images generates the scene radiance of the image. Such algorithms, if applied on an image degraded by haze, will suffer from the low-contrast scene radiance.

With the advances in the field of machine learning based on artificial neural networks, deep learning has been used in a wide area of research including text recognition and image processing. The deep learning model proposed in [1] addresses the problem of text recognition in cases where fonts are difficult to recognize due to complex backgrounds. The models proposed in [2, 3] deal with color image classification systems, which includes image conversion from color to gray scale, feature extraction and classification. [4] proposes a model to improve the contrast in images using pyramid histogram matching and thus enhancing the image. Among the different deep learning algorithms, convolutional neural network has been found to solve many real-world problems with great performance. In some of the recent works, priors for haze removal have been calculated using convolutional neural networks and advanced image filters. Also, generative adversarial networks (GANs), which were introduced by Goodfellow [5], have exhibited satisfactory performance for haze removal in images through image generation and manipulation. One of the most recent applications of GANs is proposed in [6], which uses GAN to generate images from instruction and modify the generated image using continuous instructions that follows.

In this model, a single neural network module, comprising a conditional GAN, is used to directly obtain a clear image from its hazy counterpart directly without any additional information such as explicitly estimating transmission map or haze-relevant features. The model makes use of a 54-layer Tiramisu [7] which makes use of dense blocks to enhance the information and gradient flow.

The rest of the paper has the following sections: Sect. 2 gives a brief summary on the related works in the field of haze removal using deep learning algorithms. Section 3 highlights the model used in this application and the working of GANs. Section 4 describes the data used, experiments conducted and results. The conclusions are given in Sect. 5.

2 Related Work

The problem of haze removal has been addressed in many research works. Haze removal in a single image is a challenging task and an under-constrained problem since sufficient information on the depth of an image is not obtained from a single image. Some of the earlier methods of haze removal involve generating transmission maps using prior-based methods. The assumption of a good prior plays a pivotal role in the performance of these algorithms.

Some of the earlier models concerning haze removal are model-based haze removal algorithms. One of the model-based algorithms [8] states that the color authenticity and contrast of resulting image are a direct result of the atmospheric light value. The model also proposes a transmission estimation algorithm to reduce the processing time. In [9], the weaknesses present in the atmospheric scattering model are analyzed and a model is proposed which maximizes the contrast in each scene. Once the contrast is maximized, haze is removed using an adaptive method for adjusting scene transmission. In [10], a model is proposed which performs haze removal process by estimating the thickness of the haze using a dark channel prior which works on the observation that haze-free outdoor images often contain local patches which has low intensity pixels in at least one of the color channels. The model proposed in [11] removes haze in a single image using an associative filter which computes the dark channel by transferring the structure of a reference image and gray levels of a coarse image to the filtering output.

The conventional approaches for haze removal mostly work by estimating the transmission map of the hazy scene. Estimation of the transmission map can decrease the effect of haze in an image, and thus the haze removal process can be done efficiently. Lately, with the advancements in deep learning, novel haze removal models based on deep learning have been proposed and it is shown to obtain acceptable results. A model which removes haze present in images by generating multi-channel depth maps is proposed in [12]. The generated dark channels are used to estimate the scene transmission map using multi-channel, multi-scale convolutional neural network. The scene transmission map thus obtained can be used to recover the original scene. The model proposed in [13], called All-in-One Dehazing Network (AOD-Net), uses a convolutional neural network (CNN). This model does not explicitly estimate the transmission matrix. It rather uses a modified atmospheric scattering model. In [14], Cai et.al. proposed an end-to-end trainable system, DehazeNet for generating haze-free images. DehazeNet takes a haze-induced image as input and recovers the corresponding haze-free image. This is done by estimation of the medium transmission map of the hazy image.

More recently, GANs [5] have been used for haze removal in images and it has shown improved results. GANs work by estimating generative models through an adversarial process. Two generative models are trained simultaneously: generator G and discriminator D. G tries to generate samples similar to the ground-truth samples, and D tries to estimate the probability that a sample came from the ground-truth samples rather than the ones generated by G. In [15], a GAN framework is proposed where the transmission map and final output of haze-free image are learned from the loss function using GAN. The model proposed in [16] uses a stacked GAN architecture for haze removal process on each color channel R, G and B independently. The stacking strategy in the GAN network enables to fasten the learning process.

3 Proposed Approach

In this work, a single, conditional generative adversarial network is used which is capable of removing the degradation caused by haze from hazy images without estimating the transmission map explicitly. Generative adversarial network architecture comprises two neural modules: generator (G) and discriminator (D). The modules compete with one another to generate new instances of data that can be passed as the real data. The components of the network are described in Sects. 3.1 to 3.3 (Fig. 1).

3.1 Generator

The generator (G) creates new synthetic data instances that can be passed for the real data. Conventionally, the U-Net is used as the generator which applies the classification task on every pixel so that input and output images have the same size. The U-Net has a symmetric architecture consisting of a contracting path, a convolution process which decreases the size of the image while increasing its depth and an expansive path which restores the image to its original size. This model makes use of a 54-layer Tiramisu similar to the network proposed in [17]. The dense layers present in the network enhance information and gradient flow. Replacing the conventional U-Net by a 54-layer Tiramisu yields a lesser number of trainable parameters for the generator. The 54-layer Tiramisu also contains dense blocks, transition down blocks, bottleneck blocks and transition up blocks. In each transition down layer, the operations taking place are BatchNorm followed by ReLU and a convolution operation and as a result, after passing through this layer, the spatial dimensions of the images

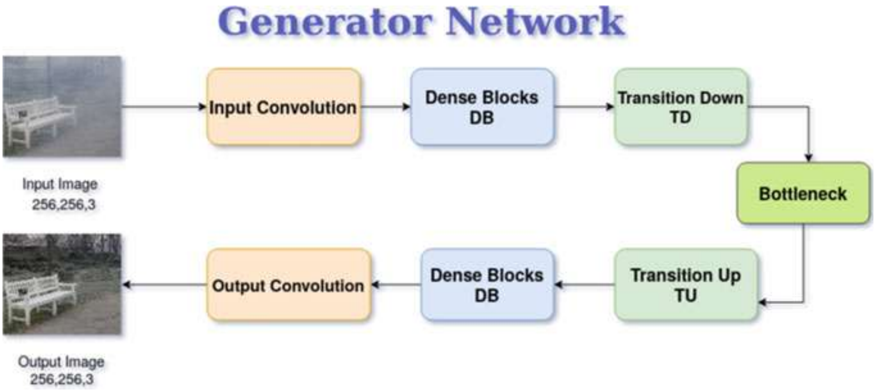


Fig. 1 Input: image degraded by haze having dimensions 256 × 256 × 3. Output: generated image with dimensions 256 × 256 × 3

are reduced by its half. In each transition up layer, the deconvolution operation takes place and as a result, after passing through this layer, the spatial dimensions of the images are doubled (Fig. 2).

The generator consists of an encoder side, decoder side and a bottleneck layer. Encoding the input data fed into the system as vectors is done by the encoder side. The decoder side reconstructs the input data based on its hidden representation. In this model, the encoder side and decoder side consist of 5 dense blocks each and the bottleneck layer consists of 1 dense block. Four layers are combined to form each of the dense blocks, and 15 layers are combined to form the dense block in the bottleneck.

3.2 Discriminator

The discriminator (D) evaluates the quality of the synthetic data instances generated by G; i.e., D determines if the image reviews belong to the original ground-truth set or not. The model uses the same discriminator network as stated in [18].

The discriminator performs the checking by passing the images through a convolutional neural network whose receptive field at the output corresponds to a patch of pixels in the original image. The image degraded by haze along with the ground-truth or generated image is fed as input to the discriminator. It outputs a 30×30 matrix, using which it can be determined if the image is fake or real.

3.3 Loss Functions

A multi-term loss function is defined for the model, similar to the one stated in [17], comprising adversarial loss (L_{Adv}), L1 loss (L_{L1}) and perceptual loss ($L_{\mathrm{Perceptual}}$).

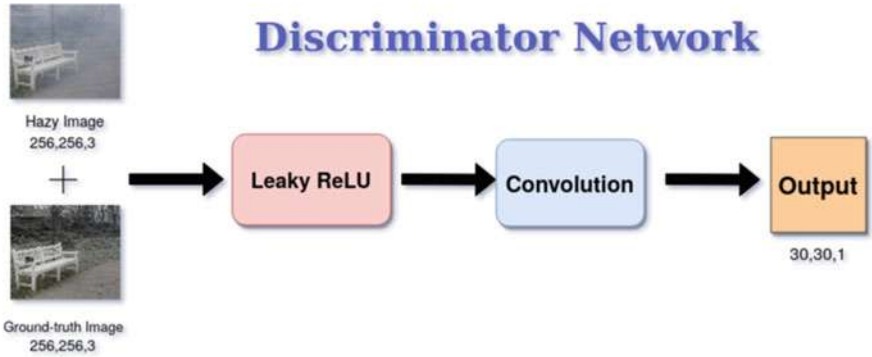


Fig. 2 Input: hazy image concatenated with ground-truth or generated image. Output: 30 × 30 matrix, used to determine if the image is generated or real

The loss components are weighted and combined to form the total loss (L), which is used to optimize the network.

$$L = W_{\text{gan}} * L_{\text{Adv}} + W_{L1} * L_{L1} + W_{\text{VGG}} * L_{\text{Perceptual}}$$

where W_{gan}, W_{L1} and W_{VGG} are scalar weights associated with the corresponding losses.

3.3.1 Adversarial Loss

According to [18], the conditional GAN loss is defined as:

$$L_{Adv} = E_{(x,y)}[\log{(D(x, y))}] + E_{(x,z)}[\log{(1 - D(x, G(x, z)))}]$$

where D is the discriminator, G is the generator, x is the input image fed into the system and the corresponding ground-truth image y is obtained. The model performs one generator update followed by one discriminator update per iteration. A scalar weight W_{gan} is multiplied with the resulting value L_{Adv}, after which it is added to the total loss L.

3.3.2 L1 Loss

As stated in [18], the L1 loss helps to reduce the distortions in the output image when it is added to the adversarial loss. The L1 loss between ground-truth image y and image generated by the model $G(x, z)$ is:

$$L_{L1} = E_{x,y,z}[|| \ y - G(x, z) \ ||] \tag{1}$$

A scalar weight W_{L1} is multiplied with the resulting value L_{L1}, after which it is added to the total loss L.

3.3.3 Perceptual Loss

As stated in [17], a feature reconstruction perceptual loss also contributes to the total loss L. The generated images and the ground-truth images are passed through a pre-trained VGG-19 network. The L2 loss of the outputs of the two images is calculated after passing through Pool-4 layer.

$$L_{\text{Perceptual}} = \frac{1}{\text{CWH}} \sum_{c=1}^{C} \sum_{w=1}^{W} \sum_{h=1}^{H} || \ V(G(x, z)^{c,w,h}) - V(y^{c,w,h}) \ ||$$

where W, H and C represent the output's dimensions (width, height and channels), respectively. The nonlinear transformation which is performed by the VGG-19 network is V. A scalar weight W_{VGG} is multiplied with the resulting value $L_{\mathrm{Perceptual}}$, after which it is added to the total loss L.

4 Experiments

The model architecture has been evaluated using synthetic haze-induced and haze-free images obtained from [1, 19]. For training the model, NYU Depth dataset [20] and O-Haze dataset [19] were used. The NYU Depth dataset contains 1449 indoor scenes recorded by RGB and depth cameras. The hazy counterparts of these images were synthesized by [17] in which homogeneous fog was added to the images by creating the corresponding transmission maps. The RGB-D images obtained from the NYU Depth dataset were used to create the corresponding transmission map of the images. The transmission map $t(x)$ obtained was scaled using a transmission coefficient chosen from the uniform distribution [0.2, 0.4] to obtain the transmission coefficient at every point in the image. At every pixel coordinate in the image, the sum of direct attenuation (product of hazy-free image pixel and corresponding transmission map) and airlight (product of atmospheric light which is set as 1 and $1 - t(x)$) produces homogeneous fog in the image. The O-Haze dataset contains 45 outdoor scenes with pairs of hazy and corresponding ground-truth images. The haze in the images was created using professional haze machines.

Figure 3 represents some hazy images from the NYU Depth dataset along with the corresponding ground-truth images. The details about the dataset generation of O-Haze can be found in [19]. All input images were resized to the dimension (256, 256, 3) as part of preprocessing.

The weights used for training the model were $W_{\mathrm{VGG}} = 10$, $W_{\mathrm{gan}} = 2$ and $W_{L1} = 100$. The learning rate value was fixed at 0.001. Gradient descent was performed to update the weights. The model was coded in Python with TensorFlow framework.

The entire dataset was split into two: training and validation sets with 0.15 of the entire dataset belonging to the validation set. In the case of NYU Depth dataset, 1232 images were in training dataset and 217 images were in validation dataset and for O-Haze, 38 images in training set and 7 images in validation set. The images for training and validation sets were chosen randomly. The model with NYU Depth dataset was trained for 18 epochs, and the model with O-Haze dataset was trained for 36 epochs. The images are shuffled in every epoch. In every iteration, the generator is updated once which is followed by one updation of the discriminator. Generator loss is when generator tries to maximize the discriminator's output for its fake instances. Discriminator loss is when discriminator tries to maximize the difference between discriminator's output on real instances and fit's output on fake instances.

During training, it was observed that the loss values generally tend to decrease over the epochs with a few fluctuations in between. Nevertheless, the generator loss and discriminator loss were decreased to negligible values on completion of the training

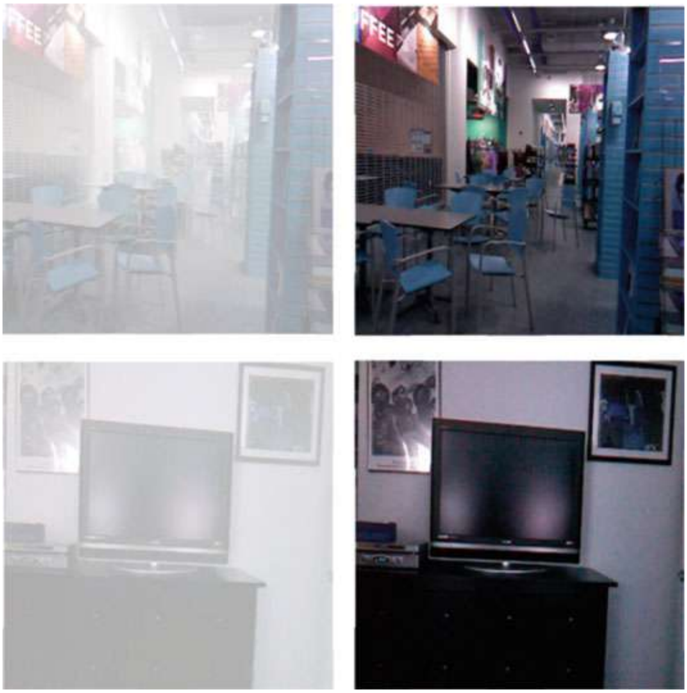


Fig. 3 Sample pairs of images from input dataset; left: hazy image ; right: ground-truth image

process. To visualize the output from the generator after validation run, sample images were provided. Apart from the loss values, for measuring the performance of the model, we consider some other factors as well.

For measuring the performance of the haze removal algorithm, two of the most commonly used performance metrics are considered: PSNR and SSIM. The quality measurement between the original image and the generated image can be measured using the metric peak signal-to-noise ratio (PSNR). In this model, the metric, PSNR measures how much the algorithm can remove haze or noise from an haze-inflicted image. Higher value of PSNR indicates better quality of reconstructed image. The metric, structural similarity index (SSIM) measures the similarity between two images. In the model, the similarity between the generated image and the ground-truth image is calculated using SSIM value. An SSIM value of 1 is obtained for two similar images.

In this model, a new metric is defined, called score, similar to the one defined in [17], which combines the properties of both PSNR and SSIM. The variable, score, is calculated as the sum of the two metrics, PSNR and SSIM after multiplying with their corresponding weights.

$$\text{Score} = W_{\text{PSNR}} * \text{PSNR} + W_{\text{SSIM}} * \text{SSIM}$$

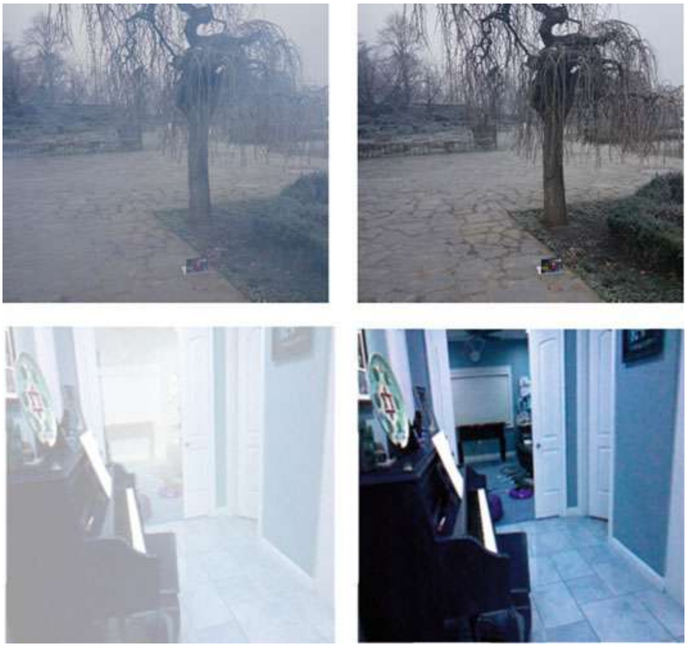


Fig. 4 Generated pairs of images from both datasets; left: hazy image ; right: generated image

The values for weights associated with PSNR and SSIM (W_{PSNR} and W_{SSIM}) are initialized as 0.05 and 1, respectively. The values for PSNR, SSIM and score are computed during the testing phase. The final PSNR and SSIM value obtained for the model with NYU Depth dataset was 19.967 and 0.692, and the model with O-Haze dataset was 20.403 and 0.734. The final score value for NYU Depth dataset was 1.69 and for O-Haze dataset was 1.75. The generated images along with the hazy counterparts are shown in Fig. 4.

From Fig. 4, it can be observed the generated image was able to extract almost all the details from the input image and it also has fairly good color contrast. The model was able to perform well even though the input images from NYU Depth dataset were heavily induced with synthetic haze. Since the number of images in O-Haze dataset was very low, the model was trained for 36 epochs. So, there are chances that the model got overfitted with the training sample, as can be seen from the visuals of the generated image.

5 Conclusion

In this paper, a model based on generative adversarial networks is reviewed for efficient haze removal in hazy images. Most of the previous works regarding haze removal in images have made use of transmission map to obtain the haze-free counterpart. When compared to those previous works, this model is able to perform well without explicitly calculating the transmission map of the image. The method evaluated in this paper yielded good results, provided sufficient training data was given to the model. From the observations recorded, it was found that the model could perform well in removing the degradation and obtaining the corresponding haze-free counterparts. The model was evaluated using a metric, which is the weighted sum of PSNR and SSIM. Other metrics could also be used to evaluate the performance of the model. From the visual outputs obtained, it was observed that the generated images did not contain the significant cloudiness which were present in the original image as a result of haze degradation. Such a model can be incorporated in various real-time applications where the presence of haze has to be removed for better performance such as automatic vehicle detection systems. One of the limitations of the model is that it was trained on only two datasets, NYU Depth dataset and O-Haze dataset, of which the latter contained less number of images. Since the model was trained on synthetically induced hazy images, further works could be done on how the model would perform on naturally hazy images. Future works could also include loss functions with more terms to improve the training process.

References

1. Shrivastava, A., Amudha, J., Gupta, D., Sharma, K.: Deep learning model for text recognition in images. In: 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, pp. 1–6 (2019). https://doi.org/10.1109/ICCCNT45670.2019.8944593
2. Sowmya, V., Govind, D., Soman, K.P.: Significance of contrast and structure features for an improved color image classification system. In: 2017 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), Kuching, pp. 210–215 (2017). https://doi.org/10.1109/ICSIPA.2017.8120608
3. Sowmya, V., Govind, D., Soman, K.P.: Significance of processing chrominance information for scene classification: a review. Artif. Intell. Rev. **53**(2), 811–42 (2020)
4. Sujee, R., Padmavathi, S.: Image enhancement through pyramid histogram matching. In: 2017 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, pp. 1–5 (2017). https://doi.org/10.1109/ICCCI.2017.8117748
5. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. :Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
6. Nadimpalli, V.V., Tekula, V.V.R., Javvaji, S., Pramod, S.S., Nair, J.J.: Iterative text-to-image conversion using recurrent generative adversarial models. IJAST **29**(4), 2953–2960 (2020)
7. Jégou, S., Drozdzal, M., Vazquez, D., Romero, A., Bengio, Y.: The one hundred layers tiramisu: fully convolutional DenseNets for semantic segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, pp. 1175–1183 (2017). https://doi.org/10.1109/CVPRW.2017.156
8. Wang, J.B., He, N., Zhang, L.L., Lu, K.: Single image dehazing with a physical model and dark channel prior. Neurocomputing. **149**, 718–28 (2015)
9. Ju, M., Zhang, D., Wang, X.: Single image dehazing via an improved atmospheric scattering model. Vis. Compu. **33**(12), 1613–1625 (2017)

10. He, K., Sun, J., Tang, X.: Single image Haze removal using dark channel prior. IEEE Trans. Pattern Anal. Mach. Intell. **33**(12), 2341–2353 (2011). https://doi.org/10.1109/TPAMI.2010.168
11. Li, B., Wang, S., Zheng, J., Zheng, L.: Single image haze removal using content-adaptive dark channel and post enhancement. IET Comput. Vis. **8**(2), 131–140 (2014). https://doi.org/10.1049/iet-cvi.2013.0011
12. Dudhane, A., Murala, S.: C2MSNet: a novel approach for single image haze removal. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, pp. 1397–1404 (2018). https://doi.org/10.1109/WACV.2018.00157
13. Li, B., Peng, X., Wang, Z., Xu, J., Feng, D.: An All-In-One Network for Dehazing and Beyond. arXiv preprint arXiv:1707.06543 (2017)
14. Cai, B., Xu, X., Jia, K., Qing, C., Tao, D.: DehazeNet: An End-to-End System for Single Image Haze Removal. IEEE Transactions on Image Processing **25**(11), 5187–5198 (2016). https://doi.org/10.1109/TIP.2016.2598681
15. Zhang, H., Sindagi, V., Patel, V.M.: Joint Transmission Map Estimation and Dehazing Using Deep Networks. IEEE Transactions on Circuits and Systems for Video Technology **30**(7), 1975–1986 (2020). https://doi.org/10.1109/TCSVT.2019.2912145
16. Suárez, P.L., Sappa, A.D., Vintimilla, B.X., Hammoud, R.I: Deep learning based single image dehazing. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, pp. 1250–12507 (2018). https://doi.org/10.1109/CVPRW.2018.00162
17. Bharath Raj, N., Venketeswaran, N.: Single image Haze removal using a generative adversarial network. In: 2020 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET), Chennai, India, pp. 37–42 (2020). https://doi.org/10.1109/WiSPNET48689.2020.9198400
18. Isola, P., Zhu, J.Y., Zhou, T. and Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1125-1134 (2017)
19. Ancuti, C.O., Ancuti, C., Timofte, R., De Vleeschouwer, C.: O-HAZE: a dehazing benchmark with real Hazy and Haze-free outdoor images. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, pp. 867–8678 (2018). https://doi.org/10.1109/CVPRW.2018.00119
20. Silberman, N., Hoiem, D., Kohli, P., Fergus, R., (2012) Indoor Segmentation and Support Inference from RGBD Images. In: Fitzgibbon A., Lazebnik S., Perona P., Sato Y., Schmid C. (eds) Computer Vision—ECCV, : ECCV 2012. Lecture Notes in Computer Science, vol. 7576. Springer, Berlin, Heidelberg (2012)

Natural Language Processing

Fake News Detection Using Passive-Aggressive Classifier and Other Machine Learning Algorithms

K. Nagashri and J. Sangeetha

Abstract Fake news means false facts generated for deceiving the readers. The generation of fake news has become very easy which can mislead people and cause panic. Therefore, fake news detection is gaining prominence in research field. As a solution, this paper aims at finding the best possible algorithms to detect fake news. In this paper, term frequency–inverse document frequency (TFIDF) as well as count vector techniques is used separately for text preprocessing. Six machine learning algorithms namely passive-aggressive classifier (PAC), naive Bayes (NB), random forest (RF), logistic regression (LR), support vector machine (SVM), and stochastic gradient descent (SGD) are compared using evaluation metrics such as accuracy, precision, recall, and $F1$ score, The results have shown that the TFIDF is a better text preprocessing technique. PAC and SVM algorithms show the best performance for the considered dataset.

Keywords Fake news · Supervised learning · Online learning · Passive-aggressive classifier · TFIDF · Count vector

1 Introduction

An enormous amount of information that is on social media which is actually true is unknown. Many people chat in the cloud via social network when any incident happens [1]. If this information were appropriate, the idea of going too far is incredibly simple, because publishing fake news publicly trigger confusion, using lies to manipulate anyone else's actions may have permanent repercussions. It is also possible for someone to share anything they wish. Fake news means wrong news

K. Nagashri (✉) · J. Sangeetha
Computer Science and Engineering, Ramaiah Institute of Technology, Bengaluru, India
e-mail: nagashri27@gmail.com

J. Sangeetha
e-mail: sangeethakirank@msrit.edu

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_19

contradicting the actual facts which may be generated by lack of proper understanding of the matter or interpreting a piece of information in an incorrect way or purposely spreading hoaxes in order to mislead and cause panic among the people. There is so much electronic information that cannot be deciphered what is right from false. It then adds to the fake news issue [2, 3].

Many works in the past have been done on the detection of fake news. Some of the works are listed here:

Granik and Mesyura [4] show the false identification of news using a naive Bayes (NB) classification approach [4]. Approximately 2000 articles have been listed in the dataset considered in this paper. This approach was used as a software application and a series of Facebook news releases were tested. The test sample obtained a classification accuracy of around 74%.

Abdullah-All-Tanvir et al. [5] provides a model for recognition of forged news from Twitter posts, in view of the computerization of falsified news recognition in Twitter datasets [5]. A comparison was made among five algorithms in machine learning, including support vector machine (SVM), the NB method, logistic regression (LR), and deep learning algorithms such as the recurrent neural network and long short-term memory. Their experimental results indicate that SVM and NB perform better than the other algorithms.

Gilda [6] examined the use of natural language processing techniques to detect counterfeit news [6]. The term frequency–inverse document frequency (TFIDF) is used for identifying fake news on a corpus of approx. 10,000 articles using a dataset from Signal Media plus a list of source material from OpenSources.com by using a probabilistic context-free grammar (PCFG). The stochastic gradient descent (SGD), SVM, gradient boosting, random forest (RF), and bounded decision trees algorithms are used to evaluate the dataset. It is shown that term frequency–inverse document frequency (TFIDF) for bigram fed on a model of SGD algorithm have an accuracy of 77.2% for identifying the fake news with PCFG showing minor effects on recall.

Reis et al. [7] identified by means of numerous classic and state-of-the-art classifiers, including RF, XGBoost (XGB), k-nearest neighbors, NB, and SVM are tested for the discriminative power [7]. No neural network model was used for comparing since handcrafted features were used, and it would just have associated weights to the features. XGB and RF classifiers with 0.85 (±0.007) and 0.86 (±0.006) values for AUC, respectively, were the best-performing algorithms. In order to rank features from every set, the relative power of the chosen attributes to discriminate every class from each other was assessed based on chi square.

The novelty of this paper is the use of an online supervised learning classifier namely the passive-aggressive classifier (PAC) for comparison which is scalable and efficient. PAC is very appropriate for the news articles which keep coming 24 × 7. Unlike the traditional approaches for batch learning which works under the assumption that the training data are already in the hand before the learning phase begins, in online learning, the predictive models are updated sequentially, and hence, this is most suitable for applications wherein training set usually arrive sequentially [8]. In this work by using TFIDF and count vector, which are the text preprocessing methods, PAC has been compared against five supervised machine learning algorithms such as NB, RF, LR, SVM, and SGD, using different performance measures like accuracy, precision, recall, and $F1$ score.

The organization of the rest of the paper is as follows. Section 2 presents the methodology wherein the description about dataset, text preprocessing techniques, algorithms, and performance metrics used are given. Section 3 shows the results obtained, and Sect. 4 concludes the work.

2 Methodology

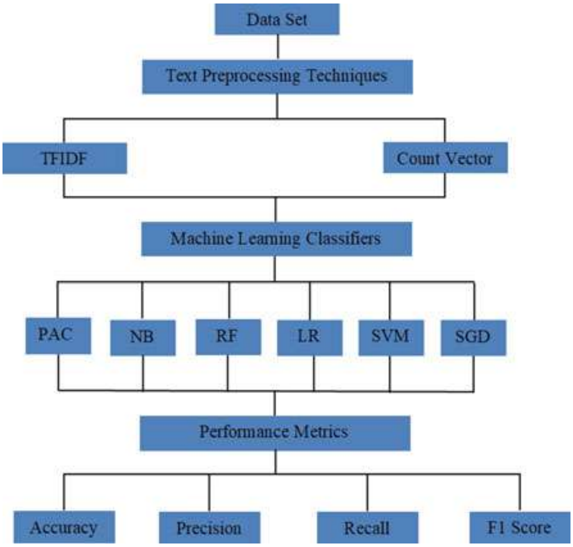
Figure 1 gives a complete overview of the methodology of our approach. The dataset is first passed through the text preprocessing techniques which is further passed on to the machine learning classifiers and is later on evaluated using the performance metrics.

2.1 Dataset

In this research, we have considered a US political dataset which is obtained from [9]. In this dataset, there are 7796 samples, three input features namely ID, title, text, and one target feature which is the label.

The features descriptions are as follows:

Fig. 1 Overview of methodology



- ID is a unique identifier to uniquely identify each instance.
- Title is the headlines of the news.
- Text is the content of the news.
- Label is the class assigned to the instance as real or fake.

Data cleaning is done first by removing the null values. Among the three input features, the ‘text’ input feature is the most relevant to our work since only from the content of the news, we can get to know whether the news is fake or real. Hence, only the ‘text’ input feature is used, and the other two features are not considered for further processing. It is a binary classification. The target feature has two classes which are real and fake, both having the same priority. Real label classifies the news as real, and fake label classifies the news as fake. 80% of the dataset is taken for training and 20% for testing purpose. It is a balanced dataset having an equal number of real and fake instances.

2.2 Text Preprocessing Techniques

The data must be first preprocessed since the textual data becomes difficult to handle as it is. For this, two text preprocessing techniques are used to convert text to numerical matrices. They are TFIDF and count vector.

Term Frequency—Inverse Document Frequency. TFIDF showcases the importance of a word to a document by assigning a weight. In text mining, this is one of the most preferred techniques. TFIDF is a product of TF and IDF. The higher the occurrence of a word, the lower is the TFIDF value. For example, “the”, a word which is most commonly used will be given a very low TFIDF value since it is not of much importance for text classification. Less frequently occurring words are given more importance and its TFIDF value will be high [4, 10]. TF, IDF, and TFIDF are calculated using Eqs. (1), (2), and (3), respectively.

$$\text{TF} = (\text{Frequency of a word in the document})/(\text{Count of words in the document}) \tag{1}$$

$$\text{IDF} = \log(\text{Count of documents}/\text{Count of documents possessing that word}) + 1 \tag{2}$$

$$\text{TFIDF} = \text{TF} * \text{IDF} \tag{3}$$

Count Vector. Count vector gives a matrix in which the highest frequency term is assigned the most weight, and it does not take into consideration the significance of a word. It is a matrix where in every row is a document, a column is the word occurring in the document, and its corresponding cell gives the count vector value of the term in that document. The disadvantage of this would be that the less important words occurring in many documents will be assigned with high weights which might cause problems with improper training of models affecting the performance metrics [4, 11].

2.3 Machine Learning Classifiers

An online supervised learning and five supervised learning algorithms are used which are described below:

Passive-Aggressive Classifier. A noteworthy classifier in the online learning algorithms is the PAC. The classification function is updated if there is a misclassification in a newly seen data, or if a predetermined margin is not exceeded by its classification score. PAC algorithm has proven to be a very effective and popular method for online learning to solve many problems in the real world [8]. PAC is an online learning classifier which is used in cases where there is a need to keep a check on the data 24 × 7 like news, social media, etc. The basic principle of this algorithm is that it sees a data, learns from the data, and discards it without the need of storing the data. For every misclassification, the algorithm reacts aggressively by updating the values, whereas for correct classification, it reacts lazily or passively. Hence, the name given is passive-aggressive classifier.

Naïve Bayes. It is a supervised learning classifier based on Bayes theorem. NB is a probabilistic algorithm where it assumes that the features are independent of one another. The probability of each word in the testing data will be calculated, and all the obtained probabilities are multiplied. For each class, a probability value is obtained, whichever is higher, the test data will be assigned to that particular class. Fake news articles usually use a similar set of words for suggesting if the particular article is actually fake. However, the article cannot be termed fake just on the grounds that there are few words that show up in it, but they have an effect on the probability. For the calculation of conditional probability, Eq. (4) is used [4, 12].

$$P(F|T) = P(T|F)P(F)/(P(T|F)P(F) + P(T|\mathrm{Tr})P(\mathrm{Tr})) \tag{4}$$

where

$P(F|T)$ conditional probability P, that news is fake given that term T occurs in it;
$P(T|F)$ conditional probability P of searching term T in the fake instances;
$P(F)$ overall probability P that a news article is fake;
$P(T|\mathrm{Tr})$ conditional probability P of searching term T in true instances;
$P(\mathrm{Tr})$ overall probability P that a news article is true.

Random Forest. RF is an ensemble supervised learning algorithm that makes use of decision trees to build a forest. It makes random splits at each level by considering different factors for each tree like the number of occurrences of the word, length of the word, and so on. Many decision trees are trained on a subset of randomly chosen features, then the prediction made are averaged. A RF will have n trees in parallel.

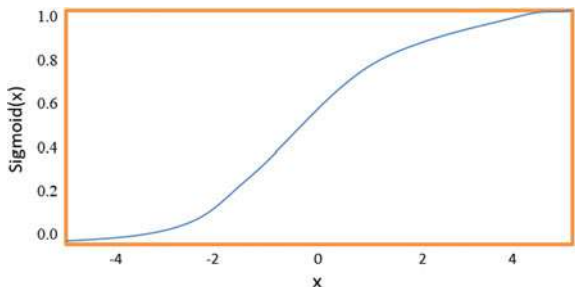


Fig. 2 Sigmoid function plot

After training all the trees, Tree 1 to Tree n for classifying a given sample X as real or fake, the class label is decided on the basis of majority of votes [13, 14].

Logistic Regression. Unlike linear regression which finds a straight line to fit the data, LR finds an S-shaped curve (Sigmoid function plot) as shown in Fig. 2. The likelihood of the data points along the curve is calculated by taking the product of the probabilities. The curve is shifted, and the process is repeated. Finally, the curve which gives the maximum likelihood is chosen. Sigmoid or logistic function ($\boldsymbol{\sigma}$) is like a step function given by Eq. (5).

$$\sigma(x) = 1/(1 + e^{-x}) \tag{5}$$

where **x** is the input data given to the sigmoid function [15, 16].

Support Vector Machine. SVM is one of the supervised learning methods having numerous desirable qualities which turn it into a powerful algorithm. This has a strong theoretical basis and performs better than many other models in various applications. Several studies have found that SVM may be one of the most accurate text classification methods. A linear SVM is made to use here for the binary classification. It finds the optimized hyper-plane to separate the two classes. The text will be transformed to a set of vectors (coordinates in space). Figure 3 shows a linear SVM. The area between the lines **A** and **C** is the maximal margin hyper-plane. The data points below line **C** belong to one class, and data points above line **A** fall in the other class [17, 18].

Stochastic Gradient Descent. It is faster than gradient descent (GD). In case of GD, the weights updating is done after the summation of all training samples. Unlike GD where the intercept and slopes are calculated for every data point, here only one data among a cluster will be randomly chosen, and the intercept and slope values are calculated incrementally. When a new data arrives, instead of starting from scratch, it takes the most recent estimate to calculate the intercept and slope. Iteration is done through every training sample, and the weight is updated. After weights are updated, the error function is calculated for every training sample according to equation. After each iteration, over the training samples, the weights are modified, and error function is calculated [19].

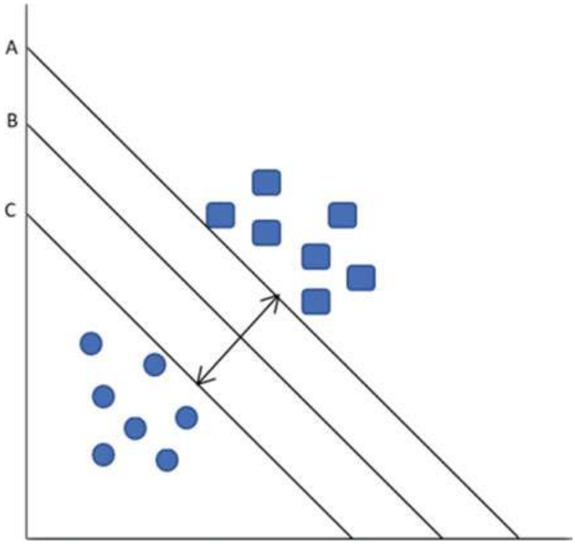


Fig. 3 Linear support vector machine

2.4 Performance Metrics

Four evaluation parameters are used to compare the algorithms. They are accuracy, precision, recall, and $F1$ score.

A confusion matrix is an $n \times n$ matrix, where n denotes the number of labels. The considered dataset has a binary classification. So, the size would be 2×2 as given in Table 1 [20].

where

TP This case arises when the actual label is real and the predicted label is also real.
FN This case arises when the actual label is real but the predicted label is fake.
FP This case arises when the actual label is fake but the predicted label is real.
TN This case arises when the actual label is fake and the predicted label is also fake.

Accuracy. Accuracy is a good performance metric when the number of samples of the classes (two classes in the current dataset) is almost equal. When the actual news is real or fake, how often that instance was indeed classified as real or fake, respectively, is given by accuracy [21]. Accuracy is calculated according to Eq. (6).

Table 1 Confusion matrix

	Predicted (real)	Predicted (fake)
Actual (real)	True positive (TP)	False negative (FN)
Actual (fake)	False positive (FP)	True negative (TN)

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FP} + \text{FN}) \tag{6}$$

Precision. If a case arose where the cost of the news which is fake but has been predicted as real (which is the FP in this case) is higher, precision would be a good measure. When the news is predicted as real, how often the prediction was correct is given by the precision metric [21]. Precision is calculated according to Eq. (7).

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}) \tag{7}$$

Recall. If a case arose where the cost of the news which is real but has been predicted as fake (which is the FN in this case) is higher, the recall would be a good measure. It is also called sensitivity. When the actual news is real, how often the prediction was real is given by recall [21]. Recall is calculated according to Eq. (8).

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}) \tag{8}$$

***F*1 Score**. It is a metric which considers both precision and recall values by giving the harmonic mean of the two metrics. This is a good performance metric when there is an imbalance in the count of the instances of the classes [21]. F1 score is calculated according to Eq. (9).

$$F1\ \text{Score} = (2 * \text{Precision} * \text{Recall})/(\text{Precision} + \text{Recall}) \tag{9}$$

Figure 4 gives the entire workflow of the implementation.

Text preprocessing methods are applied to the data which is the news content as shown in Fig. 4. The English stop words (like the articles, conjunctions, pronouns, etc.) which are very frequently used in the documents in general are removed. TFIDF and count vector are applied, and the transformed data are stored separately. This transformed data is transferred to the chosen machine learning classifiers. Test data is given to the classifier which generates the output label whether it is real or fake.

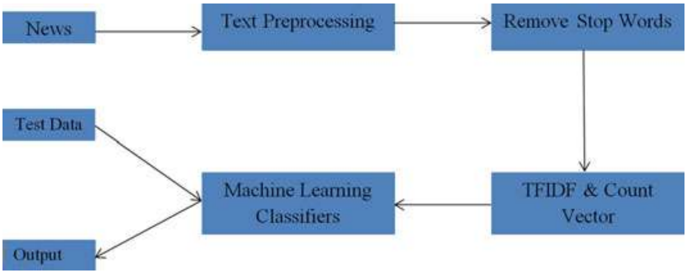


Fig. 4 Workflow of the implementation

3 Results and Discussions

The hardware and software specifications for the implementation of our work are given below.

Hardware: 8 GB RAM, Intel i3 processor, 64-bit operating system.

Software: The code is written in Jupyter notebook using Python programming language. The dataset was loaded from the csv file using pandas library. The required packages for the text preprocessing, algorithms, and performance metrics were imported from scikit learn [22]. Numpy and matplotlib libraries are used for the graphs.

In our work, two techniques are used for text preprocessing which are TFIDF and count vector. For both these techniques, we have set the threshold for document frequency as 0.6, i.e., the terms having document frequency value higher than this set threshold is ignored. Stop words are set to English, and unigrams are considered.

The comparison has been done using six supervised learning algorithms namely PAC, NB, RD, LR, SVM, and SGD. For PAC, maximum iterations over the training data are set to 1000. For NB, Laplace smoothing parameter is set to 1. For RF, 100 trees are chosen to be in the forest. For LR, maximum iterations for the algorithm to converge are set to 100. For SVM, the kernel chosen is linear. For SGD, maximum iterations over the training data are set to 1000, and an optimal learning rate is chosen. During training phase of each algorithm, we observed that SVM took the highest time, whereas NB took the least amount of time.

These algorithms have been evaluated using four performance metrics like accuracy, precision, recall, and F1 score. The results are shown through tables and graphs.

Results Obtained Using TFIDF Text Preprocessing Techniques

Table 2 shows the values obtained by all the algorithms for the different performance metrics using TFIDF.

In Table 2, it can be observed that PAC and SVM give the best accuracy when TFIDF was used and NB gives the worst accuracy. SVM gives the best precision, and PAC gives the second best, while NB gives the worst precision. For recall, NB fares the best followed by PAC, whereas RF shows the worst performance. PAC gives

Table 2 Algorithms-metrics values for TFIDF

Algorithms	Accuracy (%)	Precision (%)	Recall (%)	$F1$ score (%)
PAC	94.2	94.6	93.3	94.0
NB	82.5	74.6	97.1	84.3
RF	84.7	88.1	79.2	83.4
LR	91.5	94.4	87.7	90.9
SVM	94.2	95.0	92.9	93.9
SGD	93.8	94.4	92.9	93.6

Table 3 Algorithms-metrics values for count vector

Algorithms	Accuracy (%)	Precision (%)	Recall (%)	$F1$ score (%)
PAC	90.4	91.2	88.8	84.9
NB	88.5	85.4	92.0	88.6
RF	85.8	87.8	82.1	84.9
LR	91.6	93.1	89.4	91.2
SVM	87.8	89.4	84.9	87.1
SGD	92.2	90.6	93.7	92.1

the best, and SVM gives the second-best $F1$ score, whereas RF gives the worst $F1$ Score. The rank of accuracy values using TFIDF is PAC and SVM at the same level followed by SGD, LR, RF, and NB.

Results Obtained Using Count Vector Text Preprocessing Techniques

Table 3 shows the values obtained by all the algorithms for the different performance metrics using count vector.

In Table 3, it can be observed that SGD gives the best and LR gives the second-best accuracy when the count vector was used and RF gives the worst accuracy. LR gives the best precision, and PAC gives the second best, while NB gives the worst precision. For recall, SGD stands the best followed by NB, whereas RF shows the worst performance. SGD gives the best, and LR gives the second-best $F1$ score, whereas RF and PAC give the worst $F1$ score. The rank of accuracy values for the algorithms using count vector is SGD followed by LR, PAC, NB, SVM, and RF.

Remarks of TFIDF and Count Vector Values Obtained

A few more observations drawn from Tables 2 and 3 are:

The accuracy of PAC, SVM, and SGD algorithms using TFIDF is more than the accuracy obtained using the count vector for those algorithms, whereas for NB, RF, and LR, the accuracy obtained using the count vector is higher than the accuracy obtained using TFIDF.

The precision of the five algorithms PAC, RF, LR, SVM, and SGD using TFIDF is more than the precision obtained using the count vector, whereas only for NB, the precision obtained using the count vector is higher than the precision obtained using TFIDF.

The recall of PAC, NB, and SVM algorithms using TFIDF is more than the recall obtained using the count vector, whereas for RF, LR, and SGD, the recall obtained using the count vector is higher than the recall obtained using TFIDF.

The $F1$ score of PAC, SVM, and SGD algorithms using TFIDF is more than the $F1$ score obtained using the count vector, whereas for NB, RF, and LR, the $F1$ score obtained using the count vector is higher than the $F1$ Score obtained using TFIDF. This observation is similar to the observation made for accuracy values.

Figure 5 shows a bar plot of all the evaluation parameters which were taken into

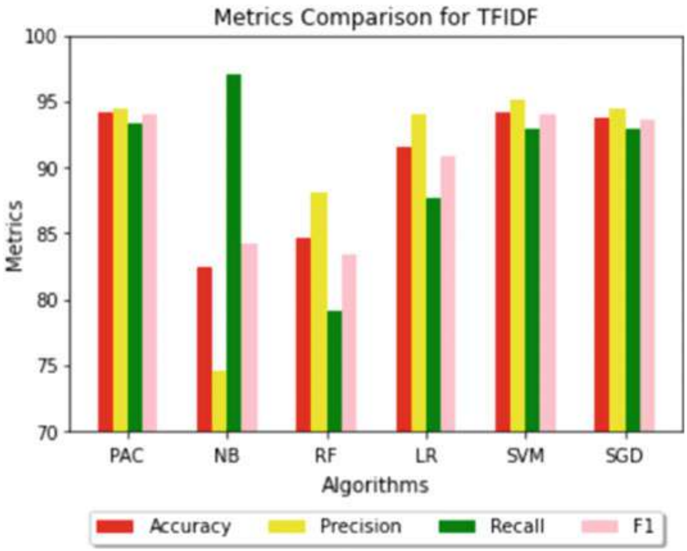


Fig. 5 Metrics versus algorithms bar plot using term frequency–inverse document frequency

consideration for the six algorithms obtained using the TFIDF text preprocessing technique.

Figure 6 shows a bar plot of all the evaluation parameters which were taken into

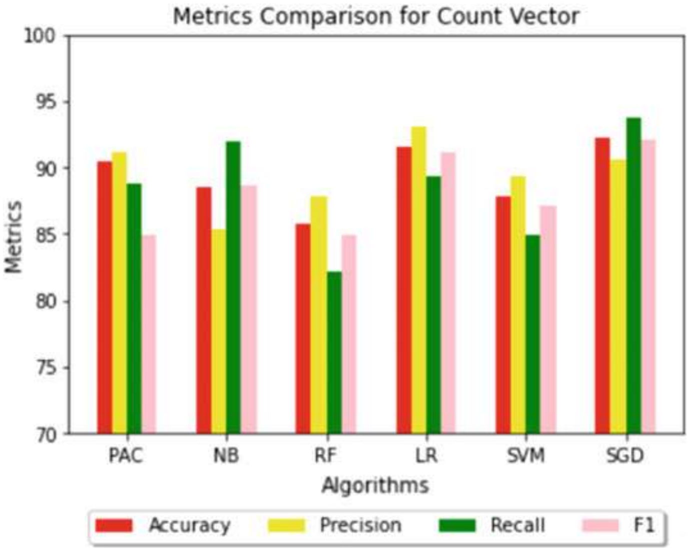


Fig. 6 Metrics versus algorithms bar plot using count vector

consideration for the six algorithms which are obtained by using the count vector text preprocessing techniques.

From Figs. 5 and 6, it is interesting to note that all the four performance metrics for PAC and SVM were higher when TFIDF was used than the metrics values obtained when count vector was used. NB fared better using count vector for all the metrics except recall. RF and LR performed better using the count vector for all the metrics except precision. SGD performed better using TFIDF for all the metrics except recall.

4 Conclusion

Fake news is the spread of information that is untrue to its source. They can be misleading and may result in serious consequences in some cases. For this reason, detecting fake news is of utmost significance. In this paper, an attempt to find the best algorithms to detect fake news was made with the help of a US political dataset by comparing different machine learning algorithms like PAC, NB, RF, LR, SVM, and SGD. TFIDF and count vector text preprocessing techniques were used to convert text to a processed numerical matrix form. They were compared using various performance measures such as accuracy, precision, recall, and $F1$ score.

From the results obtained, it was found that the TFIDF is a better text preprocessing technique than the count vector because it is based on the importance of words, whereas the count of the words is the basis for count vector. Though four performance metrics were used, the most suitable performance metric for the considered dataset is the accuracy. As discussed earlier, accuracy is a good performance metric when there is a balance in the number of instances belonging to real and fake classes and both these classes have the same level of importance. The best algorithms for this dataset among the considered algorithms using TFIDF are PAC and SVM. The conclusion was made based on accuracy. Both these algorithms gave a higher accuracy when TFIDF was used and lesser accuracy when the count vector was used.

In PAC, only if a fresh sample has been misclassified or if the classification score does not go beyond a predetermined margin, it reacts aggressively by updating the classification function [8]. For SVM, text data are perfectly suitable due to the sparse characteristic of text where in some features are not relevant, but seem to be correlated with each other and can be usually divided into categories which are separable linearly [23]. Due to these reasons, PAC and SVM fared better than other classifiers using TFIDF on this dataset giving an accuracy of 94.2%.

References

1. Aphiwongsophon, S., Chongstitvatana, P.: Detecting fake news with machine learning method. In: 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Chiang Rai, Thailand, pp. 528–531

(2018). https://doi.org/10.1109/ecticon.2018.8620051

2. Shu, K., Sliva, A., Wang, S., Tang, J., Liu, H.: Fake news detection on social media. ACM SIGKDD Explorations Newsl. **19**, 22–36 (2017)
3. Ozbay, F., Alatas, B.: Fake news detection within online social media using supervised artificial intelligence algorithms. Phys. A **540**, 123174 (2020)
4. Granik, M., Mesyura, V.: Fake news detection using Naive Bayes classifier. In: 1st Ukraine Conference on Electrical and Computer Engineering (UKRCON), IEEE, Kiev, Ukraine, pp. 900–903, 2017. https://doi.org/10.1109/ukrcon.2017.8100379
5. Abdullah-All-Tanvir, M.E.M., Akhter, S., Huq, M.R.: Detecting fake news using Machine learning and deep learning algorithms. In: 7th International Conference on Smart Computing and Communications (ICSCC), Sarawak, Malaysia, pp. 1–5, 2019. https://doi.org/10.1109/icscc.2019.8843612
6. Gilda, S.: Evaluating machine learning algorithms for fake news detection. In: 15th Student Conference on Research and Development (SCOReD), IEEE, Putrajaya, Malaysia, pp. 110–115, 2017. https://doi.org/10.1109/SCORED.2017.8305411
7. Reis, J.C.S., Correia, A., Murai, F., Veloso, A., Benevenuto, F.: Supervised learning for fake news detection. IEEE Intell. Syst. **34**, 76–81 (2019)
8. Lu, J., Zhao, P., Hoi, S.C.H.: Online passive-aggressive active learning. Machine Learn. **103**, 141–183 (2016)
9. Fake news dataset. https://data-flair.training/blogs/advanced-python-project-detecting-fake-news/. Accessed 10 Apr 2020
10. Ahmed, H., Traore, I., Saad, S.: Detection of online fake news using n-gram analysis and machine learning techniques. Lecture Notes in Computer Science, pp. 127–138, 2017
11. Kaliyar, R.K., Goswami, A., Narang, P.: DeepFakE: improving fake news detection using tensor decomposition-based deep neural network. J. Supercomput. (2020)
12. Poddar, K., Amali, D.G.B., Umadevi, K.S.: Comparison of various machine learning models for accurate detection of fake news. In: Innovations in Power and Advanced Computing Technologies (i-PACT), Vellore, India (2019). https://doi.org/10.1109/i-pact44901.2019.8960044
13. Kowsari, K., Meimandi, K.J., Heidarysafa, M., Mendu, S., Barnes, L.E., Brown, D.E.: Text classification algorithms: a survey. Information **10** (2019)
14. Saravanan, R., Sujatha, P.: A State of art techniques on machine learning algorithms: a perspective of supervised learning approaches in data classification. In: 2nd International Conference on Intelligent Computing and Control Systems (ICICCS), IEEE, Madurai, India, pp. 945–949, 2018. https://doi.org/10.1109/iccons.2018.8663155
15. Mitchell, T.M.: Machine Learning. McGraw-Hill, USA (1997)
16. Geron, A.: Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 1st edn. O'Reilly Media, USA (2017)
17. Moraes, R., Valiati, J.F., Neto, W.P.G.: Document-level sentiment classification: an empirical comparison between SVM and ANN. Expert Syst. Appl. **40**, 621–633 (2013)
18. Raju, M.P., Laxmi, A.J.: IOT based online load forecasting using machine learning algorithms. Proc. Comput. Sci. **171**, 551–560 (2020)
19. Harrington, P.: Machine Learning in Action. Manning Publications Co., USA (2012)
20. Gereme, F.B., Zhu, W.: Early detection of fake news "Before it flies high". In: 2nd International Conference on Big Data Technologies (ICBDT), ACM, New York, USA, pp. 142–148, 2019. https://doi.org/10.1145/3358528.3358567
21. Gupta, P., Garg, S.: Breast cancer prediction using varying parameters of machine learning models. Proc. Comput. Sci. **171**, 593–601 (2020)
22. Python Machine Learning Library. https://scikit-learn.org/stable/. Accessed 10 Apr 2020
23. Medhat, W., Hassan, A., Korashy, H.: Sentiment analysis algorithms and applications: a survey. Ain Shams Eng. J. **5**, 1093–1113 (2014)

Generative Adversarial Network-Based Language Identification for Closely Related Same Language Family

Ashish Kar, P. G. Sunitha Hiremath, and Shankar Gangisetty

Abstract The discrimination between similar languages is one of the main challenges in automatic language identification. In this paper, we address this problem by proposing a generative adversarial network-based language identification method for identifying the sentences from closely related languages of same language family. The proposed method works on dual-reward feedback learning comprising of generator to generate nearly close language sentences, discriminator for determining how similar the generated sentences are to that of the training data and classifier for optimal prediction of the correct label. We evaluate the proposed model for pairs of languages and overall testing data comparison on Indo-Aryan languages dataset [12]. The effectiveness of our method is demonstrated in comparison to other existing state-of-the-art methods.

Keywords Language identification · GAN · Indo-Aryan languages · Semi-supervised · RNN encoder · Closely related languages · Same language family · Text classification

1 Introduction

Languages that share common script are challenging to be identified by language identifiers [10]. Today, automatic language identification is an important task in many natural language processing applications. We look to address the existing issues in language identification (LI) domain and improve the language identification challenge where the languages are closely related and come from same language family.

A. Kar (✉) · P. G. Sunitha Hiremath · S. Gangisetty
KLE Technological University, Hubballi, India
e-mail: 01fe16bcs047@kletech.ac.in

P. G. Sunitha Hiremath
e-mail: pgshiremath@kletech.ac.in

S. Gangisetty
e-mail: shankar@kletech.ac.in

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_20

A language family is a group of languages related through descent from a common ancestral language or parental language. We propose a generative adversarial network (GAN)-based LI model to classify closely related languages, given the languages come from same origin family. Closely-related languages, language varieties, and dialects have also been the focus of a number of shared tasks in recent years [10]. One of the major problem in LI is identifying language pairs from the same language family which also share a common script and the same encoding. For instance, English and Dutch share a common ancestor Proto-Germanic where sentences by script are challenging to identify. Consider for example,

washing machine in workplace in English
wasmachine op de werkplek in Dutch

Similar languages like English and Dutch present significant lexical and structural overlap, making it challenging for systems to discriminate between them. Under resourced regional languages, in addition, lack part of speech taggers and language model for accurate language identification like Afrikaan languages [3] and various other language family like Indo-Aryan languages [8]. These are regional languages but have prominent influence over other languages making it interesting to study.

LI finds its applications in machine translation, transliteration, sentence generation and dialogue-based systems [10]. Inaccurate results from wrong LI can hamper the quality of the system and user experience. Recently, in Facebook, a technical error caused vulgar translation of Chinese leader's name from Burmese where source language was detected incorrectly [1]. For such challenges, the research is still open in LI, and we develop a novel GAN-based LI method for closely related languages.

The proposed GAN-based LI consists of generator, discriminator, and classifier. The generator produces similar fake sentences to that of the training data. The discriminator differentiates between real and fake language sentences for improving prior to overall feedback. The classifier determines weather the generator language sentences belong to correct languages. The dual-reward feedback from discriminator and classifier helps the generator to learn language sequences better. The dual feedbacks are the overall feedback at the end of the sequence and the leaked feedback. Thus, the adversary produces correct language sequences for prediction.

The main contributions of this paper are:

- We propose GAN-based LI method to solve the closely related similar languages identification problem.
- The GAN-based LI method consists of generator, discriminator and classifier built on recurrent neural network (RNN) encoder and decoder to improve the sequence learning of language sentences.
- We propose a dual-reward feedback obtained from discriminator and classifier for the generator to reduce the closeness of similar languages identification while sentence generation.
- We perform exhaustive analysis of the GAN-based LI model with existing state-of-the-art methods on Indo-Aryan languages dataset [12].

2 Related Works

An extensive survey on features, methods, and open issues on automatic language identification for text processing is discussed in [10]. The LI methods can be classified into hand-crafted and deep learning-based methods.

2.1 Hand-Crafted LI Methods

In [11], authors focused on the task of classifying short segments of the text in closely related languages for discriminating similar languages. They used nave Bayes classifier with characters and word n-gram features. Experiments were performed on groups of similar languages. However, the method lacked on domain specific classification and suffered from domain overfitting. In [2], authors proposed support vector machine (SVM) ensemble technique involving characters and word features to discriminate between five similar languages of the Indo-Aryan family [12]. However, the method lacked in performance due to mis-classification of languages. In [8], authors proposed unsupervised iterative LI adaptation model on Indo-Aryan dataset. In [6], authors presented ensemble method using naïve Bayes classifier with word n-gram as features, SVMs with both character and syntactic features, and neural networks with pre-trained character/word embeddings to distinguish Mainland and Taiwan Chinese languages. However, the sparse syntactic features affected the performance of the model. In [3], authors presented short text LI for under resourced languages. Under resourced languages lack LI tools like parser and POS-tagger; hence, it becomes difficult for such languages to be identified. This method uses hierarchical naïve Bayesian and lexicon-based classifier.

2.2 Deep Learning-Based Methods

In [7], authors focused on HeLI method for dialects of Swiss-German LI. Helsinki LI is a word-based transfer learning method. The languages considered are Bern, Basel, Lucern and Zurich. In [4], authors used six different deep learning models for Arabic LI data. The Arabic LI dataset consisted of Egyptian, Gulf, Iraqi, Levantine and Maghrebi languages. Attention-based bidirectional RNN method performed best among all the other deep learning methods. It required a large dialect specific word embeddings model for optimal performance. In [13], authors used hybrid architecture that combines character-based convolutional neural network (CNN) features with weighted bag-of-n-grams features for LI. The dataset consisted of German, Arabic and other languages. However, the performance was affected by named-entities during classification. Recently, in [9], authors proposed HeLI 2.0 considering unsupervised language model adaptation. The language model adaptation for language and dialect identification of text where used to enhance the performance of LI. The languages considered were Swiss-German and Indo-Aryan. However, this approach involves language model adaptation which was computationally expensive.

The existing hand-crafted and deep-learning methods do not focus on generation of language sequences based on paired similar languages for better classification on closely related languages of same language family. Some of the methods need ensemble and adaptive learning techniques for better performance which is computationally expensive. Other non-discriminative LI methods suffer penalty to particular language classification if vocabulary is not present. To address these issues, we propose GAN-based LI model which deals with GANs for sequence learning for given vocabulary of similar languages. GANs perform better with similar languages as they learn to generate sentences in those languages along with classification based on vocabulary and order of sequence. Thus, our method works efficiently with limited training data and is computationally less expensive. We adopt sequence generation learning in order to generate similar languages training data and hence enhancing the performance of closely related languages.

3 GAN-Based LI Architecture

We propose a GAN-based LI method shown in Fig. 1 to classify closely-related languages, given the languages are from the same origin family. The proposed GAN-based LI method consists of three major components, namely discriminator D, generator G and classifier C. Initially, we pre-process and clean the sentences from the languages and generate 10,000 words vocabulary dictionary for each language. We then pass the restricted language sentences adopting the parameter setting as specified in Algorithm 1. The generator learns from the training sentences and produces similar sentences (fake). The discriminator differentiates the similarity between sentences from the generator and real training dataset. The classifier judges whether the fake sentences generated belongs to the correct languages. The dual-reward feedback from the discriminator and classifier is fed to the generator for better learning the language sentence sequence generation.

Considering a sequence generation as a sequence decision-making process [15], RNN performs better in discrete sequence generation like text sentences than CNN and can generate relatively longer sentences. In Fig. 1, the generator takes two feedbacks to learn the sequences ($s_{1:N}$, where s_1 to s_N are the words in the sequence), one is the overall feedback, and the other is leaked feedback obtained from discriminator to generator. In the generator, the extractor uses RNN encoder to set goals from the leaked signals. The leaked feedback (D_f) is used for improving the results prior to overall feedback to the generator. We use RNN decoder in the discriminator to extract features of a sequence. The discriminator feedback (D_F) in addition to the leaked feedback helps to address the mode collapse problem. In the classifier, the extractor uses RNN decoder and provides classifier feedback (C_F) to generator

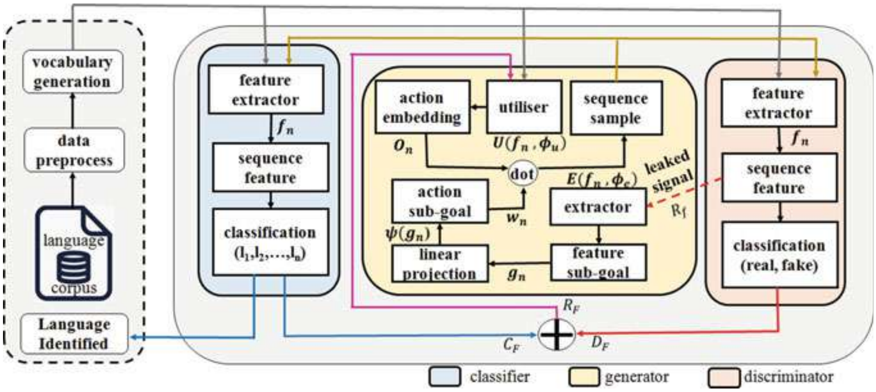


Fig. 1 Proposed semi-supervised GAN-based language identification architecture for closely related languages

for producing sentences with correct labels. Thus, the adversary between all three components helps generator to understand and produce language sequences better and with correct label for prediction of the testing data.

3.1 Generator

The generator (G) consists of two components, namely extractor (E) and utiliser (U). The extractor and utiliser starts from an all-zero hidden state. The training data is fed to the utiliser. Training data consists of language sentences with their respective class labels. The utiliser uses RNN encoders and serves as a component to perform conditional distribution on the sentences, known as goals. The utiliser generates similar sentences in given languages as of that in training data. The utiliser computes conditional probability matrix based on words in vocabulary for each language. This is followed by action embedding to arrange vocabulary for building sentences.

The words in the vocabulary are sampled to an ordered conditional sequence, output matrix O_n given as,

$$O_n = U\left(s_{1:N}, \phi_u\right) \tag{1}$$

where ϕ_u is utiliser network parameter. Equation 1 needs goals to improve the sentences formed which is obtained from the extractor. The extractor uses RNN encoders and serves as a component to compute goals from leaked signal, as well set goals for utiliser. The leaked signals are feedbacks revealing internal state of the discriminator during sequence learning by generator. During this computation, the extractor accepts leaked signals from D as D_f. The extractor finds a parameterised conditional distribution for the generated fake language sentences based on noise vector (q), context vector (f) and the network parameter (ϕ_g) for the language sequence as $G\left(s_{1:N}|q, f, \phi_g\right)$ [14].

The context vectors are based on leaked signals given to extractor at every timestep, and this also ensures retaining of labels and features of the sequence during the sequence generation. The reward function for this step is given as,

$$R_f = G(q, f, \phi_g | s_{1:N}) \tag{2}$$

This feedback sets feature sub-goals which are advantageous directions for utiliser to follow. The feature sub-goals are stored in goal matrix (g_n). The extractor outputs goal vector g_n based on discriminator feature extractor f_n (see Sect. 3.2 Eq. 8), given as,

$$\hat{g}_n,\ s_n = E(f_n\ ,\ s_{n-1};\ \phi_e) \tag{3}$$

$$g_n = \frac{\hat{g}_n}{||\hat{g}_n||} \tag{4}$$

where ϕ_e is the network parameter for extractor.

Linear projection is done on g_n to obtain goal embedding vector w_n. For utilising the goals produced by the extractor, a linear transformation ψ is used to obtain weight matrix W_ψ, which is performed as a summation over recent x goals to produce a k-dimensional goal embedding vector w_n [5], given as,

$$w_n = \psi\left(\sum_{i=1}^{x} g_{n-i}\right) = W_\psi\left(\sum_{i=1}^{x} g_{n-i}\right) \tag{5}$$

The w_n matrix from the extractor is combined with O_n matrix from the utiliser and given as a feedback to the generator G at timestep n. The dot product between O_n and w_n is performed to obtain probability for each vocabulary in language sequences. The scalar product of both O_n and w_n is given to generate language samples in G that are similar to training data. These samples are sent to D and C. In return, D and C provides feedback to G for the given generated sentences. The G also receives leaked signals from D during sequence learning. Both the feedback and leaked signals give reward functions during sequence learning. The G utilises two reward functions, namely R_f and R_F from leaked signal and combined feedback. Thus, we compute R_F as,

$$R_F = 2.\frac{D_F(s_{1:n-1}, s_n).C_F(s_{1:n-1}, s_n, c)}{D_F(s_{1:n-1}, s_n) + C_F(s_{1:n-1}, s_n, c)} \tag{6}$$

where there is a reduce cross entropy loss on the following token s_n over s_{n-1}. The overall reward function for G is given as,

$$R(s_{1:n}, c) = R_f + R_F \tag{7}$$

The utiliser keeps updating sequences based on feedbacks and extractor keeps setting directions for it to follow in order to improve every cycle of training. During adversarial training, gradient ascent is performed to update the G using the gradient as specified in SpamGAN [14].

3.2 Discriminator

The discriminator gets the language sequences from training data and generator. The discriminator determines how similar the generated languages are to that of the training data. The discriminator uses RNN decoder to predict if a language sentence belongs to training data or generator. If discriminator learns fake sentences as real, that's where generator is doing good and able to fool the discriminator. The discriminator extracts features from language sequences and compares them to samples in training data for determining sentences as real or fake. The discriminator tries to differentiate between real and fake language sequences based on conditional probability score $D(s_{1:N}|\phi_d)$ where ϕ_d is network parameter for discriminator. It is computed at every timestep n as approximated score, $D_V(s_{1:n-1})$. This score is averaged at the end of the sequence to produce the overall score $D_F(s_{1:n-1}, s_n)$ [14]. Unlike spamGAN [14], we make the score to be revealed to the generator as leaked feature f_n for better sequence generation, given as,

$$f_n = D_F(s_{1:n-1}, s_n) \tag{8}$$

After the sentences are determined as fake or real, the feedback is sent to the utiliser of generator. We also use f_n as leaked signal and send it to the extractor of generator. Both the feedbacks are used by generator for sequence learning as discussed in Sect. 3.1.

While training, we maximise $D(s_{1:N}|\phi_d)$ for real language sentences from training data and minimise for fake language sentences before utilising the f_n and sending leaked signal to extractor. We maximise the score by minimising discriminator loss before giving feedback D_F at the end of the sequence. The error minimisation and policy gradient calculation are adopted from spamGAN [14].

3.3 Classifier

The architecture of classifier is similar to the discriminator, but does not require adversarial training with generator, and the classifier is mutually bootstrapping. To learn correct label for sentences, classifier computes probability score similar to discriminator as $C_V(s_{1:n-1}, c)$ at timestep n and performs an average to produce the overall score $C_F(s_{1:n-1}, s_n, c)$. After sentence classes are determined, the feedback is sent to the utiliser of generator. Since the generator will have better understanding of the sequence by two feedbacks, the classification at inference stage tends to improve along with the usual classifier feedback. The error minimisation and policy gradient calculation are adopted from spamGAN [14]. While classifier is competing with generator to determine correct label during sequence learning, it helps classifier to learn labels of language sequences better. Thus, the classifier classifies the language sentences into real or fake.

3.4 Algorithm

The algorithmic procedure for GAN-based LI method is provided in Algorithm 1. The input to the algorithm is labelled language dataset D_{L_n}. The weights are assigned for parameters during the pre-training phase. In the pre-training phase, each component is trained individually and not adversarially. The *adv-steps* are used to sample and generate sentences. The *D-steps* compute discriminator and leaked feedback and the *C-steps* computes (Eq. 7) classifier (Eq. 4) feedback [14].

Algorithm 1: GAN-based LI Algorithm

```
Input: Training labeled dataset $D_{L_n}$
Parameters: $\phi_g$ , $C_{crit}$ , $D_{crit}$ , $\phi_u$ , $\phi_e$ , $f_n$ , $w_n$ , $O_n$ , $g_n$
Output: Predicted class label $P$
Perform pre-training to update network parameters.
1.Initialize generator with random weights $\phi_g, \phi_u, \phi_e$
2. while adv-steps do
    3. Generate fake sequence $s_{1:N} = (s_1, .., s_N)c$
    4. while $n \epsilon 1 : N$ do
        5. Store leaked information $f_n$ from $D$
        6. Compute $R(s_{1:N}, c)$
        7. Get the goal direction $g_n$ from extractor
        8. Update utiliser parameters $\phi_u$
        9. Update extractor parameters $\phi_e$
    10. Update $\phi_g$ using policy gradient [14]
11. while D-steps do
    12. Sample batch of real sentences from $D_L$
    13. Sample batch of fake sentences from $G$
    14. Extractor computes goal embedding matrix $w_n$
    15. Compute $D_F(s_{1:n-1}, s_n)$, $D_V(s_{1:n-1})$ for fake sentences
    16. Update $D_{crit}$ using policy gradients [14]
17. while C-steps do
    18. Sample batch of real sentences-class pairs from $D_L$
    19. Sample batch of fake sentence-class pairs from $G$
    20. Update classifier using policy gradients
    21. Compute $C_F(s_{1:n-1}, s_n, c)$, $C_V(s_{1:n-1}, c)$ on fake sentences
    22. Update $C_{crit}$ using policy gradients [14]
    23. Output $P$
```

4 Experiments and Results

4.1 Data Pre-processing

In pre-processing (Eq. 5) stage, we obtain vocabulary for each language. We remove the punctuation marks, URL links, hashtags, foreign language from sentences and generate vocabulary dictionary for each language upto 10,000 words. We restrict sentence length to 125 words based on median of sentences and pad each sentence with $\langle start \rangle$ for start of sentence and $\langle end \rangle$ for end of sentence. We use $\langle pad \rangle$ if length of the sentence is smaller than 125 words. For fake sentence generation, we restrict length of sentence to 10 words.

4.2 Implementation Details

The proposed GAN-based LI architecture is implemented on Nvidia m4000 GPU processor with 1502 MHz memory clock and 8 GB RAM. We have set the parameters same as spamGAN [14] for generator, discriminator and classifier. For language sequence generation, length is set to 20, the learning rate for utiliser and extractor is set to 0.001. The goal dimension size k is set to 16. The embedding size of the LSTM-RNNs is set to 32.

4.3 Language Dataset

For experimental analysis, we use Indo-Aryan languages dataset [12] from VarDial challenge organised at COLING 2018. The dataset consists of five languages namely Hindi (HIN), Braj Bhasha (BRA), Awadhi (AWA), Bhojpuri (BHO) and Magahi (MAG) split into training, validation, and testing datasets, respectively. In the Indo-Aryan languages dataset [12], the number of sentences in training data for HIN, MAG, BRA, BHO and AWA language are 15,228, 14,921, 14,686, 145,630 and 9055. Similarly, the validation data for each of the languages are 1792, 1815, 1848, 1639, and 1192, and testing data for each of the languages are 1708, 2064, 2002, 1866 and 1392. We compare our LI method with HeLI 2.0 [9] for all pairs of language using F_1 score, and SVM ensembler [2] and iterative language model (LM) [8] for overall testing data using confusion matrix.

4.4 LI for Pair of Languages

The pairwise F_1 score comparison for all pairs of Indo-Aryan languages compared to HeLI 2.0 approach [9] are shown in Table 1. We observe that our LI method performs better w.r.t. F_1 scores for all the pairs of language due to dual reward feedback by GAN architecture. The highest F_1 score is for (BRA-BHO) pair with 0.975, and the lowest F_1 score is for (HIN-AWA) pair with 0.8 because of syntactically most similar and closer making it tougher to predict. The syntactic features depend on the vocabulary and order of words in the sentences of each language. The similarity between languages serves as an additional information from our LI method; however, this is subjective to various syntactic features other than the ones evaluated. Our LI method effectively performs better for LI at lower computational cost and limited training data unlike HeLI 2.0.

4.5 LI for Overall Testing Data

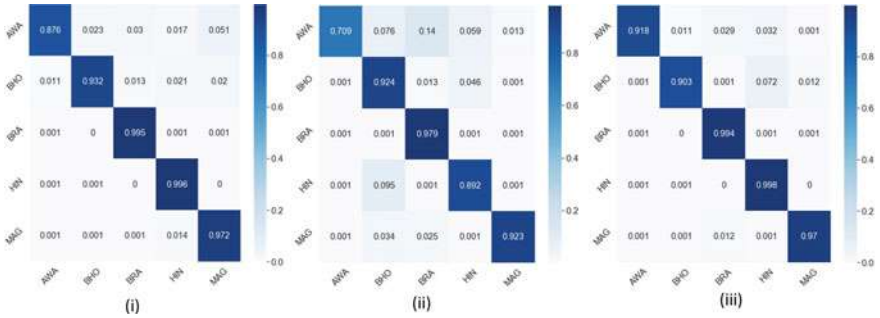
The confusion matrix for LI of our method compared to SVM ensembler [2] and iterative LM [8] on Indo-Aryan languages dataset are shown in Fig. 2. The proposed LI model performs better on Indo-Aryan dataset compared to SVM ensembler and iterative LM. The mis-classifications reported by our method is comparatively low which can be observed in the confusion matrix shown in Fig. 2i. As an example for mis-classification of languages due to close similarity in the LI of Indo-Aryan languages family, we show Bhojpuri (BHO) and Hindi (HIN) in Fig. 3. We observe that our LI method predicts the BHO test sentences correctly unlike SVM ensembler and iterative LM which mis-classifies the sentences as HIN. This performance of our method is due to the dual-reward feedback learning that we employed during the generator for fake sequence generation to enhance the performance of LI in closely related languages.

4.6 Perplexity and Fake Sentence Generation

The perplexity of the sentences generated by our LI method is shown in Fig. 4. The proposed LI method improves the perplexity with combination of labelled and unlabelled data for training dataset. It achieves the lowest perplexity of 98.2 on 100% labelled data and increases gradually with unlabelled data from Indo-Aryan languages dataset. There is no overfit observed in our analysis. This shows that using unlabelled data improves the generator performance in producing realistic language sentences for preparing semi-supervised training data for LI method. Fake sequences generated by our method were used for data augmentation for enhancing performance in LI. The length of sentences for fake generated sentences was 10 words. In Fig. 5, we have shown some of the intermediate fake sentences generated by the generator of our LI model for different Indo-Aryan languages.

Table 1 F_1 score comparison of our GAN-based LI method with HeLI 2.0 [9] (zoom-in to see the scores)

Method	HIN-MAG	HIN-BRA	HIN-BHO	HIN-AWA	MAG-BRA	MAG-BHO	MAG-AWA	BRA-BHO	BRA-AWA	BHO-AWA
Ours	**0.891**	**0.973**	**0.896**	0.800	**0.967**	**0.912**	**0.925**	**0.975**	**0.885**	**0.864**
HeLI 2.0	0.887	0.965	0.880	**0.826**	0.958	0.896	0.889	0.963	0.816	0.801



(i)

	AWA	BHO	BRA	HIN	MAG
AWA	0.876	0.023	0.03	0.017	0.051
BHO	0.011	0.932	0.013	0.021	0.02
BRA	0.001	0	0.995	0.001	0.001
HIN	0.001	0.001	0	0.996	0
MAG	0.001	0.001	0.001	0.014	0.972

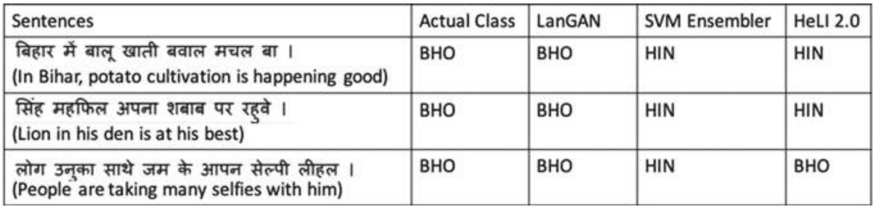
(ii)

	AWA	BHO	BRA	HIN	MAG
AWA	0.709	0.076	0.14	0.059	0.013
BHO	0.001	0.924	0.013	0.046	0.001
BRA	0.001	0.001	0.979	0.001	0.001
HIN	0.001	0.095	0.001	0.892	0.001
MAG	0.001	0.034	0.025	0.001	0.923

(iii)

	AWA	BHO	BRA	HIN	MAG
AWA	0.918	0.011	0.029	0.032	0.001
BHO	0.001	0.903	0.001	0.072	0.012
BRA	0.001	0	0.994	0.001	0.001
HIN	0.001	0.001	0	0.998	0
MAG	0.001	0.001	0.012	0.001	0.97

Fig. 2 Confusion matrix comparison of (i) our LI method with (ii) SVM ensembler [2], and (iii) iterative language model [8] (zoom-in to see the cell values)



Sentences	Actual Class	LanGAN	SVM Ensembler	HeLI 2.0
बिहार में बालू खाती बवाल मचल बा । (In Bihar, potato cultivation is happening good)	BHO	BHO	HIN	HIN
सिंह महफिल अपना शबाब पर रहुवे । (Lion in his den is at his best)	BHO	BHO	HIN	HIN
लोग उन्का साथे जम के आपन सेल्पी लीहल । (People are taking many selfies with him)	BHO	BHO	HIN	BHO

Fig. 3 Mis-classification examples of Bhojpuri (BHO) sentence from the testing data of Indo-Aryan languages dataset [12]

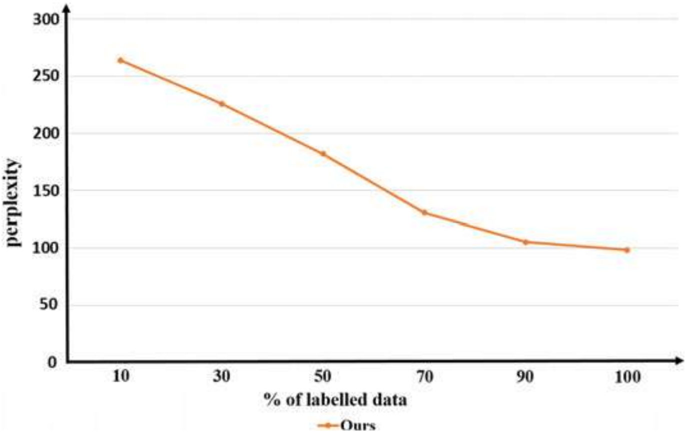
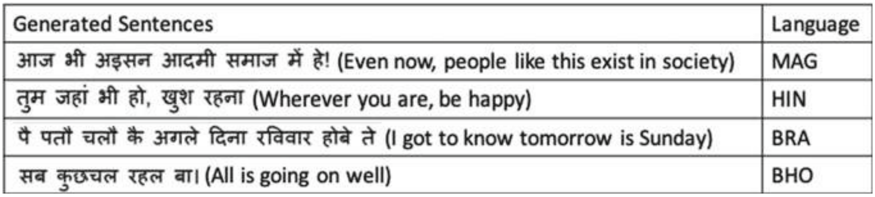


Fig. 4 Determination of unlabelled data on perplexity



Generated Sentences	Language
आज भी अइसन आदमी समाज में हे! (Even now, people like this exist in society)	MAG
तुम जहां भी हो, खुश रहना (Wherever you are, be happy)	HIN
पै पतौ चलौ कै अगले दिना रविवार होबे ते (I got to know tomorrow is Sunday)	BRA
सब कुछचल रहल बा। (All is going on well)	BHO

Fig. 5 Sample fake sentences of various Indo-Aryan languages generated by our LI model

5 Conclusions and Future Work

In this paper, we proposed GAN-based LI architecture to solve the LI problem on closely related similar languages based on sequence learning with exclusive vocabulary for each language. The proposed LI method includes generator for fake language sentence learning, discriminator for real language identification and classifier for correct prediction of the generated sentence. The dual-reward feedback from discriminator and classifier is used by the generator for optimised sequence learning and generate language sentences closely related to training languages data. Our LI method performed well compared to state-of-the-art methods on Indo-Aryan family of languages. The best F_1 score was for (BRA-BHO) pair with 0.975 of syntactically most similar languages. We also provided result analysis in terms of mis-classification issues and fake sentence generation which is handled by our method more effectively. The limitations of our model is that performance might reduces if foreign language vocabulary is used and code-mixing languages to be identified since our LI model relies on vocabulary. As a future work, we plan to advance our approach to solve the limitations and offensive language detection problems.

References

1. Facebook says technical error caused vulgar translation of Chinese leader's name. reuters.com/article/us-myanmar-facebook/facebook-apologizes-after-vulgar-translation-of-chinese-leaders-name-idUSKBN1ZH0IB (2020)
2. Ciobanu, A.M., Zampieri, M., Malmasi, S., Pal, S., Dinu, L.P.: Discriminating between Indo-Aryan languages using SVM ensembles. In: VarDial@COLING 2018, pp. 178–184. Association for Computational Linguistics (2018)
3. Duvenhage, B.: Short text language identification for under resourced languages. CoRR abs/1911.07555 (2019)
4. Elaraby, M., Abdul-Mageed, M.: Deep models for Arabic dialect identification on benchmarked data. In: VarDial@COLING 2018, pp. 263–274. Association for Computational Linguistics (2018)
5. Guo, J., Lu, S., Cai, H., Zhang, W., Yu, Y., Wang, J.: Long text generation via adversarial training with leaked information. In: AAAI, pp. 5141–5148. AAAI Press (2018)

6. Hu, H., Li, W., Zhou, H., Tian, Z., Zhang, Y., Zou, L.: Ensemble methods to distinguish mainland and Taiwan Chinese. In: Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects, pp. 165–171. Association for Computational Linguistics (2019)
7. Jauhiainen, T., Jauhiainen, H., Lindén, K.: Heli-based experiments in Swiss German dialect identification. In: VarDial@COLING 2018, pp. 254–262. Association for Computational Linguistics (2018)
8. Jauhiainen, T., Jauhiainen, H., Lindén, K.: Iterative language model adaptation for Indo-Aryan language identification. In: VarDial@COLING 2018, pp. 66–75. Association for Computational Linguistics (2018)
9. Jauhiainen, T., Lindén, K., Jauhiainen, H.: Language model adaptation for language and dialect identification of text. Nat. Lang. Eng. **25**(5), 561–583 (2019)
10. Jauhiainen, T., Lui, M., Zampieri, M., Baldwin, T., Lindén, K.: Automatic language identification in texts: a survey. J. Artif. Intell. Res. **65**, 675–782 (2019)
11. King, B., Radev, D.R., Abney, S.P.: Experiments in sentence language identification with groups of similar languages. In: VarDial@COLING, pp. 146–154. Association for Computational Linguistics and Dublin City University (2014)
12. Kumar, R., Lahiri, B., Alok, D., Ojha, A.K., Jain, M., Basit, A., Dawer, Y.: Automatic identification of closely-related Indian languages: resources and experiments. CoRR abs/1803.09405 (2018)
13. Martinc, M., Pollak, S.: Combining n-grams and deep convolutional features for language variety classification. Nat. Lang. Eng. **25**(5), 607–632 (2019)
14. Stanton, G., Irissappane, A.A.: GANs for semi-supervised opinion spam detection. In: IJCAI, pp. 5204–5210 (2019)
15. Yu, L., Zhang, W., Wang, J., Yu, Y.: SeqGAN: sequence generative adversarial nets with policy gradient. In: AAAI, pp. 2852–2858. AAAI Press (2017)

Statistical and Neural Machine Translation for Manipuri-English on Intelligence Domain

Laishram Rahul, Loitongbam Sanayai Meetei, and H. S. Jayanna

Abstract This paper describes the development and results of Manipuri-English machine translation system built on an intelligence domain. Manipuri is an under-resourced Tibeto-Burman language that is spoken mainly in the North-Eastern states of India. A total of 56,678 Manipuri-English parallel corpora from intelligence domain based on the open-source intelligence (OSINT) keywords and phrases are collected for the experiment. An evaluation of statistical machine translation (SMT) and neural machine translation (NMT) is carried out in terms of BLEU score. A BLEU score of 23.91 is achieved with the SMT-based approach which is outperformed by the NMT-based system with a BLEU score of 40.67. Further, a language-specific morphological analysis based on the suffixes is investigated. The findings on the incorporation of morphological analysis report a BLEU score of 25.03 with the SMT and a BLEU score of 44 with NMT, both of which are a significant improvement.

Keywords Low resource · Manipuri · Morphological analysis · OSINT · Machine translation · SMT · NMT

1 Introduction

Machine translation (MT) system translates a human language to another human language without the intervention of a human translator. MT systems have evolved over the years and different approaches have been used to develop it. Rule-based approaches [1] were employed in the earlier development of the MT systems. Design-

L. Rahul (✉) · H. S. Jayanna
SIT, Tumkur, India
e-mail: laishramrahulib@gmail.com

H. S. Jayanna
e-mail: jayannahs@gmail.com

L. S. Meetei
NIT Silchar, Silchar, India
e-mail: loisanayai@gmail.com

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_21

ing such a system is an exhaustive task since it requires linguistic knowledge and translation rules are needed to be predefined. The paradigm than sifted towards the corpus-based machine translation systems. Example-based machine translation (EBMT) [2–4], SMT [5, 6] and NMT [7, 8] are widely used corpus-based MT system. There has been a significant improvement with the introduction of neural machine translation with respect to the traditional SMT system. Recent trends also focused on training an MT system in specific domains such as medical [9, 10].

Despite the advancement of MT systems, for a low resource language like Manipuri, it is still far from reaching a standard, comparable to the MT systems for major languages such as Spanish, French and Japanese. Manipuri, a Tibeto-Burman language, is the lingua franca of Manipur, a North-Eastern state of India. It is one of the 22 scheduled languages of India and it is spoken by around 1.7 million people according to the language census of India 2011. It is also spoken in the state of Assam, Tripura and by some population in the neighbouring countries of Bangladesh and Myanmar.

Lack of training dataset, either monolingual or bilingual text in electronic format, set back the development of MT systems for low resource languages. The challenges of developing an accurate MT system increase if the source language and target languages are widely dissimilar. Aside from having a different word order (SOV) with that of English word order (SVO), Manipuri is an agglutinative language where words have extensive suffixation with limited prefixation. Manipuri is written using the Bangla script or the Meitei/Meetei Mayek script. At present, Bangla script is widely used for most documents and so the reason for using Bangla script for this research. In this work, translation from Manipuri to English on an intelligence domain is carried out. A comparison of SMT and NMT-based approaches are investigated in this experiment. Further, the usage of language-specific morphological analysis (MA) is studied and implemented in the MT systems. MT system trained on such intelligence domain dataset can help tackle certain tasks related to national security and law enforcement.

The rest of the paper is structured as follows: Sect. 2 discusses the previous related works; Sect. 3 describes the dataset preparation and MT models used in the experiment; an analysis of findings and results are reported in Sect. 4. Section 5 provides the conclusion of the work.

2 Related Works

SMT solves computational problem of translating text from a source language to a target language by utilizing aligned parallel text corpus. Koehn et al. [5] developed a phrase-based statistical machine translation (PBSMT) achieving a better result than traditional word-based methods. The author reported that phrases of the length of up to 3 words show the best performance. With the emergence of deep learning, different variants of NMT with encoder-decoder architecture [7, 11, 12] are available. By mainly concentrating on segments of source sentence during translation, Luong et al. [12] investigate two types of attention mechanisms, namely global and local. The global approach attends to all source words at all times while the local looks at a subset of source words at a time. The author reported that the models show better results when translating names and long sentences.

Various methodologies of MT system are investigated in some of the existing works on MT for low resource Indian languages: Manipuri-English MT system [4, 13], Gujarati-English MT system [14], English-Mizo MT system [16] and English-Hindi MT system [15]. Due to the lack of parallel corpus for these low resource language pairs, the authors trained their MT systems on a small dataset. In WMT19[1] shared task, the author [14] uses a monolingual corpus along with parallel dataset to train a NMT system achieving a BLEU score of 12.8 on test set. Using a small dataset of around 32,000 parallel training dataset, the author [15] explored the use of multiple modalities as an input to NMT system and reported a BLEU score of 28.45 on a test set. MT work on Mizo, a Sino-Tibetan language, lingua franca of Mizoram was reported by Pathak et al. [16] using PBSMT and NMT systems. Trained on a corpus of around 33,000 English-Mizo parallel sentences, the PBSMT achieved a BLEU score of 18.36 and 22.37 on two different test sets of 100 sentences each while the NMT system scored 21.48 and 22.60. The author reported the PBSMT system to achieve better performance than the NMT system on sentences of length greater than 5, while the NMT system performed better on sentences of length below 5.

Morphological analysis plays an important role in the MT of agglutinative languages [13, 17, 18]. With a corpus of 11,450 parallel sentences from news domain, Singh and Bandyopadhyay [13] developed a bidirectional SMT system for Manipuri-English. The author identified case markers and part-of-speech tags in Manipuri and dependency relations in English as an important factor for the Manipuri-English MT system. The MT system is trained on a parallel corpus of 10,350 sentences and tested with 500 sentences. The author reported a significant improvement in the translation quality with the factored SMT model achieving a BLEU score of 17.573. Pimpale et al. [17] proposed a factored SMT system for Marathi-Hindi by utilizing the morphological richness of Marathi through the splitting of compound words. The model is reported to improve accuracy with respect to the baseline [17]. Using the PBSMT system, Bisazza and Federico [18] studied the morphology of Turkish by utilizing various morphological word segmentation methods. The author reported that splitting the suffixes selectively enhances the performance of the Turkish-English MT system.

3 System Description

In order to train MT systems, both SMT and NMT architectures are employed separately in this experiment. Further, the use of language-specific morphological analysis is evaluated on the SMT and NMT architecture.

[1] http://www.statmt.org/wmt19/.

Table 1 Statistics of the dataset

Language	Sentences	Avg tokens per sentence	Unique tokens
Manipuri	56,678	9	55,440
English	56,678	10	22,296

3.1 Data Preparation

A total of 56,678 Manipuri-English parallel sentences are used in the experiment. Out of these, 2000 are used as development sets and 1000 are used as test sets, and the rest are utilized for training the system. The parallel corpus is gathered based on OSINT keywords and phrases. The OSINT dataset is collected from publicly available domains such as newspapers, magazines, TV, and Internet to be used in an intelligence context. A summarized statistics of the collected dataset is shown in Table 1. In Table 1, it could be observed that there is a significant difference in the number of unique tokens between Manipuri and English. The prime reason behind this is that Manipuri is a phonetically typed language and multiple graphemes in the Bangla script could be used to represent a single phoneme in Manipuri. For example, the word ‘rebels’ is written in different ways as লালহৌভসিং, লালহৌবশিং, লালহৌবসিং.

The corpus is pre-processed through tokenization and normalization. Moses tokenizer [19] is used to tokenized the English text. As for the Manipuri text, the tokenization is carried out by splitting words in white space. However, before tokenization, it is validated that all the punctuation is also spaced by white space except for dots and commas between numerals.

The tokenized English text is then normalized by converting the upper case to lowercase. In the case of Manipuri language, there is no upper or lower case, however, some of the acronyms are written in Roman characters, such text is lowercased, e.g., ‘GPS’ to ‘gps’.

3.2 Phrase-Based Statistical Machine Translation

In phrase-based SMT, a sequence of words with different lengths is translated. Unlike word-based SMT, phrase-based SMT allows the translation from several to several words and not only from one to several. To translate a foreign language f into English $\boldsymbol{e}_{best}$, a translation probability (Eq. 1) based on Bayes rule is used.

$$\boldsymbol{e}_{\text{best}} = \text{argmax}_e p(e|f) = \text{argmax}_e p(f|e) p_{\text{LM}}(e) \tag{1}$$

where $p(e|f)$ is the translation model and $p_{\text{LM}}(e)$ is the language model.

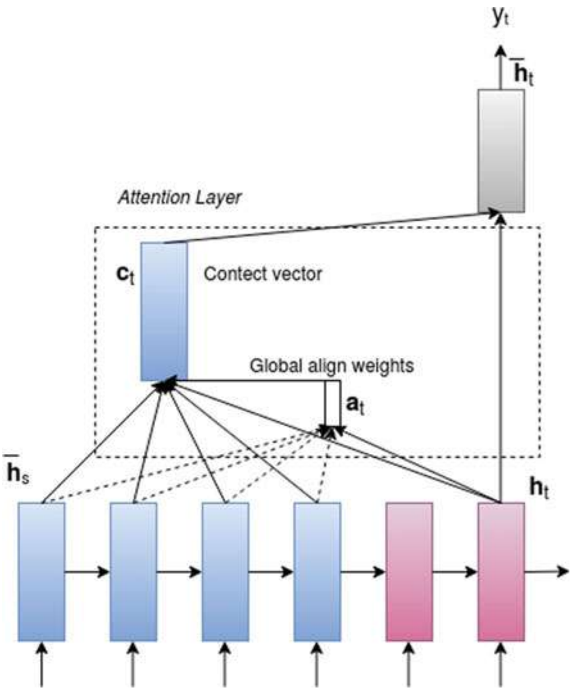


Fig. 1 NMT with global attention model

The PBSMT is implemented using the Thot [20] toolkit. HMM-based alignment models are used to obtain the word alignment matrices required for phrase model estimation. A 3-gram language model is then incorporated with the system.

3.3 NMT System

NMT follows an encoder-decoder model where the encoder maps the source text into a vector of inputs and the decoder decodes the input vector to the output target text. NMT global attention mechanism [6] where attention is focused on all source positions (Fig. 1) is used in the experiment. The attention mechanism considers all hidden states of the encoder to derive context vector (c_t). To calculate c_t a variable-length alignment vector, a_t is computed from the similarity measure between the source hidden state $\overline{h}_s$ and target hidden state h_t as follows:

$$a_t(s) = \text{align}(h_t, \overline{h}_s) = \frac{\exp(\text{score}(h_t, \overline{h}_s))}{\sum_{s'} \exp(\text{score}(h_t, \overline{h}_{s'}))} \tag{2}$$

where score is a content based function.

Table 2 Statistics of common suffixes

দগী	গী	দা	না	কী	নি	শিং	কি	দী	দি	সু	গনী	নী
893	6247	7278	4920	1324	1546	1016	350	121	383	269	111	111

In this experiment, two-layer LSTM with 500 nodes in each layer is used for both the encoder and decoder network. The model is optimized using stochastic gradient descent (SGD) with a learning rate of 1. The NMT system is train up to 150,000 iterations before the model converges as per the dataset availed for the purpose.

3.4 Morphological Analysis

Manipuri is an agglutinative language with extensive suffixation. A set of common suffixes in Manipuri text identified in this experiment is shown in Table 2. The occurrence of the mentioned set of suffixes is then identified from the Manipuri dataset along with their frequency. Words with length less than or equal to 3 characters, e.g., the words "সনা", "ঊনা", etc., are omitted from counting. The suffixes which have the frequency of occurrence close to or greater than 1000 are taken into consideration and the words which have such suffixes are separated into two tokens. Further, the words with suffixes which has occurrence of less than 400 as shown in Table 2 are split into two tokens. However, tokenization of words with suffixes which has less occurrence as mentioned, and training the MT system leads to a decrease in accuracy; therefore, such tokenization is not considered.

The morphological analysis leads to the tokenization of the Manipuri corpus into subword tokens resulting in the increase of average tokens per sentence from original 9 to 12. However, the tokenization process based on the morphological analysis significantly reduces the number of unique tokens from 55,440 to 43,616 in the overall corpus.

4 Result and Analysis

Experiments are carried out using both SMT and NMT systems in two scenarios, i.e., with and without incorporating the language-specific morphological analysis. The MT systems are trained with Manipuri as the source language and English as the target language. The collected dataset is partitioned into training, development and test dataset as shown in Table 3.

Employing the architecture described in Sect. 3, a SMT system (SMT$_b$) and a NMT system (NMT$_b$) are trained without the inclusion of morphological analysis as the baseline system. Later, by incorporating the language-specific morphological

Table 3 Dataset break-up

Training	Development	Test	Total
53,678	2000	1000	56,678

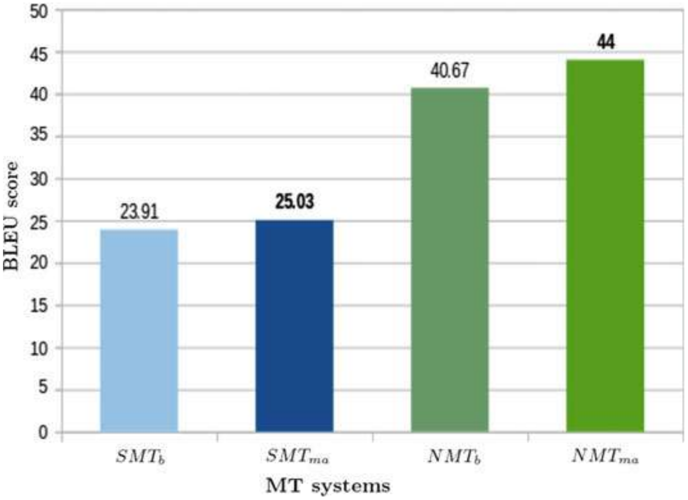


Fig. 2 Evaluation of MT systems using BLEU score

analysis, a SMT system (SMT$_{\text{ma}}$) and a NMT system (NMT_{ma}) are trained. The final result is evaluated using BLEU [21] and Fig. 2 shows the experimental results obtained. The chart in Fig. 2 summarizes the performance of the different MT systems obtained with the x-axis representing the MT systems grouped by the evaluation methods and the y-axis representing their BLEU scores. The result shows that in terms of BLEU score, the NMT system (NMT$_b$ = 40.67, NMT$_{\text{ma}}$ = 44) outperform the SMT system (SMT$_b$ = 23.91, SMT$_{\text{ma}}$ = 25.03) by a large margin in all cases. The result shows that the NMT system achieves a BLEU score of 40.67 outperforming the SMT system (BLEU score = 23.91) by a large margin. The result also highlights that the morphological analysis applied in the experiment improves both the SMT and the NMT systems significantly. With the morphological analysis, the SMT system achieves a BLEU score of 25.03 (SMT$_{\text{ma}}$) while the NMT system achieves a BLEU score of 44 (NMT$_{\text{ma}}$). Sample input and output of the MT systems used in the experiments are shown in Table 4.

Table 4 Sample input–output

Source	রেঞ্জরশিংগী গারীদা গ্রেনেদ এতেক তৌরকপবু গবর্নরনা কন্না য়ানিংদবা ফোংদোকখ্রে।
Reference	Governor slams grenade attack on rangers vehicle
SMT$_b$	Rangers vehicle in grenade attack on governor says stating that khan was attacked because he had been actively
SMT$_{\text{ma}}$	Rangers vehicle in grenade attack governor
NMT$_b$	Rangers thwarts grenade attack on rangers coach
NMT$_{\text{ma}}$	Governor condemns grenade attack on rangers vehicle

5 Conclusion

Manipuri is a low resource language with limited tools for NLP. The agglutinative of the language where words have extensive suffixation pose a challenge to the development of MT system for Manipuri. In this paper, the evaluation on morphological analysis of Manipuri to train Manipuri-English MT systems is carried out. Unlike most of the MT systems which are trained on an independent domain dataset, a domain-dependent dataset related to intelligence domain is trained and discussed in this paper. The analysis of this experiment shows that NMT system achieves better performance than SMT system by a large margin. Tokenization being a key aspect while working with the text data, the subword tokenization of Manipuri using the language-specific morphological analysis is carried out. The morphological analysis of Manipuri is used in this experiment results in a significant improvement in the Manipuri-English MT output for both the SMT (up to a +1.11 BLEU points) and NMT (up to a +3.33 BLEU points) systems. The performance report observed in this paper reflects that the morphological analysis of Manipuri improves the Manipuri-English MT system.

Acknowledgements This work was supported by Tata Power Strategic Engineering Division. The authors appreciate the anonymous reviewers for their valuable comments.

References

1. Bennett, W.S., Slocum, J.: The LRC machine translation system. Comput. Linguist. **11**(2–3), 111–121 (1985)
2. Nagao, M.: A framework of a mechanical translation between Japanese and English by analogy principle. Artif. Hum. Intell. 351–354 (1984)
3. Zhou, M., Huang, J.X., Huang, C.N.T., Wang, W.: Example based machine translation system. U.S. Patent 7,353,165 (2008)

4. Singh, T.D., Bandyopadhyay, S.: Manipuri–English example based machine translation system. Int. J. Comput. Linguist. Appl. (IJCLA) (2010). ISSN: 0976-0962
5. Koehn, P., Och, F., Marcu, D.: Statistical phrase-based translation. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, vol. 1, p. 4854. Association for Computational Linguistics (2003)
6. Singh, T.D.: Taste of two different flavours: which Manipuri script works better for English–Manipuri language pair SMT systems? In: Proceedings of the Seventh Workshop on Syntax, Semantics and Structure in Statistical Translation, pp. 11–18 (2013)
7. Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: encoder–decoder approaches. arXiv preprint arXiv:1409.1259 (2014)
8. Karakanta, A., Dehdari, J., van Genabith, J.: Neural machine translation for low-resource languages without parallel corpora. Mach. Transl. **32**(1–2), 167–189 (2018)
9. Eck, M., Vogel, S., Waibel, A.: Improving statistical machine translation in the medical domain using the Unified Medical Language System. In: COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics, pp. 792–798 (2004)
10. Wołk, K., Marasek, K.: Neural-based machine translation for medical text domain. Based on European medicines agency leaflet texts. Procedia Comput. Sci. **64**, 2–9 (2015)
11. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
12. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025 (2015)
13. Singh, T.D., Bandyopadhyay, S.: Manipuri–English bidirectional statistical machine translation systems using morphology and dependency relations. In: Proceedings of the 4th Workshop on Syntax and Structure in Statistical Translation, pp. 83–91 (2010)
14. Mondal, R., Nayek, S.R., Chowdhury, A., Pal, S., Naskar, S.K., van Genabith, J.: JU-Saarland submission to the WMT2019 English–Gujarati translation shared task. In: Proceedings of the Fourth Conference on Machine Translation, Shared Task Papers, Day 1, vol. 2, pp. 308–313 (2019)
15. Meetei, L.S., Singh, T.D., Bandyopadhyay, S.: WAT2019: English–Hindi translation on Hindi visual genome dataset. In: Proceedings of the 6th Workshop on Asian Translation, pp. 181–188 (2019)
16. Pathak, A., Pakray, P., Bentham, J.: English-Mizo machine translation using neural and statistical approaches. Neural Comput. Appl. **31**(11), 7615–7631 (2019)
17. Pimpale, P.B., Patel, R.N., Sasikumar, M.: SMT from agglutinative languages: use of suffix separation and word splitting. In: Proceedings of the 11th International Conference on Natural Language Processing, pp. 2–10 (2014)
18. Bisazza, A., Federico, M.: Morphological pre-processing for Turkish to English statistical machine translation. In: NNNN (2009)
19. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C.: Moses: open source toolkit for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, pp. 177–180. Association for Computational Linguistics (2007)
20. Ortiz-Martínez, D., Casacuberta, F.: The new Thot toolkit for fully-automatic and interactive statistical machine translation. In: Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics, pp. 45–48 (2014)
21. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318 (2002)

Fake Review Detection Using Hybrid Ensemble Learning

Sindhu Hegde, Raghu Raj Rai, P. G. Sunitha Hiremath, and Shankar Gangisetty

Abstract Opinion spam on online restaurant review sites are a major problem as the reviews influence the users' choice to visit or not to a restaurant. In this paper, we address the problem of detecting genuine and fake reviews in restaurant online reviews. We propose a fake review detection technique comprising data preprocessing, detection and ensemble learning that learns the reviews and their features to filter out the fake reviews. Initially, we preprocess to obtain the refined reviews and employ two independent classifiers using deep machine learning and feature-based machine learning techniques for detection. These classifiers tackle the problem in two aspects, i.e., the deep machine learning model learns the word distributions and the feature-based machine learning model extracts the relevant features from the reviews. Finally, a hybrid ensemble model from the two classifiers are built to detect the genuine and fake reviews. The experimental analysis of the proposed approach on Yelp datasets outperforms the existing state-of-the-art methods.

Keywords Opinion spam · Fake restaurant reviews · Yelp dataset · Hybrid ensemble learning · Text classification · Deep machine learning · Feature-based machine learning

Work done during Undergraduate study at KLE Technological University.

S. Hegde (✉)
IIIT Hyderabad, Hyderabad, India
e-mail: sindhu.hegde@research.iiit.ac.in

R. Raj Rai · P. G. Sunitha Hiremath · S. Gangisetty
KLE Technological University, Hubballi, India
e-mail: 01fe16bcs157@kletech.ac.in

P. G. Sunitha Hiremath
e-mail: pgshiremath@kletech.ac.in

S. Gangisetty
e-mail: shankar@kletech.ac.in

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_22

1 Introduction

Due to the enormous amount of restaurants that are coming up globally, there is always a competition among the restaurants to attract the customers for economic viability. The restaurants' popularity is influenced by the location, restaurant services and facilities such as ambience, discounts, offers, Wi-Fi, parking, delivery, which are promoted via online media like advertisements and social networking sites [10, 11]. Review sites such as TripAdvisor.com, Yelp.com, Zomato.com and OpenTable.com provide customers a platform to share their opinions and experiences about the restaurants. With most of the users relying on the review sites for deciding on the restaurants, the impact of the reviews on customers' turn up is immense. The review sites offer user-generated content in the form of electronic word-of-mouth (WoM) services, which are often considered more reliable and trustworthy. Also, online customer reviews have become significant in swaying customer decisions. Majority of the customers depend on reviews due to numerous reasons like risk reduction, search-time reduction, buyer's remorse and group influence as specified in [17]. However, the fake reviews have become one of the biggest threats in today's restaurant world as many review sites are plagued by fake reviews, e.g., TripAdvisor [20], Yelp, Zomato [12] and OpenTable. Since customer reviews are the real hunt for the restaurants to stay in the race, the restaurant hires spammers for generation of fake or deceptive reviews (crowdturfing [23]) to boost their reputation or to damage that of the competitor.

With the tremendous growth of customer reviews, it has become highly challenging to inspect all the reviews and detect their genuineness. Today, with AI becoming popular that can replace human spammers to generate fake reviews which are indistinguishable from genuine reviews written by human users, the detection of fake reviews is in demand. Thus, in this paper, we focus on reviews unlike other metadata, which is crucial in detecting fake reviews as there is a need for efficient fake review detection systems to tackle the menace of abundant fake over genuine reviews. We use Yelp in our work because it has become the dominant source of consumer reviews in the restaurant industry. Users tend to trust the reviews on Yelp and engage in the community aspects of the platform [17]. Studies on Yelp found that a one star rating increase for restaurants can lead to a 5–9% boost in revenue [13].

In our work, we propose a fake review detection technique which comprises three modules, namely data preprocessing, detection and ensemble learning. First, we preprocess the restaurant reviews from the Yelp datasets [18] using techniques such as stop word removal, part-of-speech (PoS) tagging and lemmatization to obtain the refined reviews. The refined data is represented using term frequency-inverse document frequency (tf-idf). Next, we build two independent classifiers using machine learning and deep learning techniques which classify the reviews into genuine and fake categories. In feature-based machine learning classifier, the tf-idf represented data along with other features obtained from the review text is given to the Naive Bayes classifier. On the same lines, to build a deep machine learning classifier, the tf-idf representation is fed to the multilayer perceptron. The results of both the classifiers are then ensembled to build a hybrid ensemble model which boosts the performance of detection of genuine and fake reviews.

The major contributions of our work are:

- We propose to build a novel hybrid ensemble learning architecture using feature-based machine learning and deep machine learning classifiers to detect the genuine and fake reviews.
- We explore and extract different features from the review text and meta-data which are in-turn given as inputs to the feature-based Naive Bayes and deep machine learning-based multilayer perceptron classifiers.
- We provide an evaluation of the proposed technique on the Yelp datasets [18]. The experimental results demonstrate the effectiveness of the proposed technique over the existing SOTA methods.

In Sect. 2, different categories of fake review detection approaches are discussed. In Sect. 3, we discuss the proposed fake review detection technique. In Sect. 4, we present the experimental results. In Sect. 5, we provide the concluding remarks.

2 Related Work

A comprehensive comparative study of review spam detection using various machine learning techniques is provided in literature [3]. For detection of fake reviews, majority of the works can be categorized into three approaches, namely linguistic [6, 15, 16], behavioral [14, 22] and graph-based [1, 5, 18, 21] approaches.

2.1 Linguistic Based Approaches

In [16], authors introduced automated approach to deceptive opinion spam detection based on computational linguistics and psychology. The n-gram-based machine learning classifiers were considered to classify the reviews into truthful and deceptive categories. In [15], authors presented generative model of deception which jointly models the classifier's uncertainty as well as the ground-truth deceptiveness of each review. In [6], authors discussed syntactic stylometry for deception detection. The features from context-free grammar parse trees are considered over lexico-syntactic features. And the results are evaluated on hotel review data of [16].

2.2 Behavioral-Based Approaches

In [14], authors do an in-depth investigation of the fake reviews on Yelp and provide an accuracy of 86% considering behavioral features in real-life fake reviews. The authors also counter argue indicating that Yelp's filter might be using similar behavioral approach. In [4], authors present fake review detection from hype restaurant reviews. The hype reviews considered are based on taste, environment, service and overall attitude. In [2], authors use review-centric and user-centric features for classifying fake reviews in the consumer electronics domain. The behavior from user-centric features is subgrouped into personal, reviewing, social and trust. In [9], authors proposed neural network model-based fake review detection using emotions hidden in the reviews with semantic meaning of reviews. Two deep feed-forward neural network and convolution neural network models learn document-level representation from n-grams, word embeddings (bag-of-words) and various lexicon-based emotion indicators for spam detection.

2.3 Graph-Based Approaches

In [18], authors use a graph-based approach which exploits both relational and metadata collectively in order to detect fishy users and reviews. The features considered for fake review detection are linguistic features and behavioral features. The authors provide large-scale quantitative evaluation results on opinion spam detection for Yelp reviews. In [21], authors presented a review graph by providing a relationships between reviewers, reviews and stores reviewed by the reviewers. By making use of this graph, an iterative model was built to identify untrusting reviewers in Yelp. In [8], authors proposed multi-task learning method based on logistic regression by sharing the knowledge present in training data exploiting the relatedness task. Unlike learning multiple task simultaneously, in this approach, every domain level detection is considered different like the spam restaurant reviews from restaurant domain is a different task. Thus, to improve the deceptive detection w.r.t. unlabelled data, a semi-supervised multi-task learning model via graph-based Laplacian regularized logistic regression is devised.

Our work is a hybrid method which is a culmination of linguistic and behavioral approaches.

3 Hybrid Ensemble Learning Architecture

We propose a hybrid ensemble learning architecture for detection of fake reviews is shown in Fig. 1. The proposed ensemble model comprises of three modules, namely data preprocessing (Sect. 3.1), detection (Sect. 3.2) and ensemble learning (Sect. 3.3). In data preprocessing module, the input reviews are cleansed and lemmatized to obtain refined reviews. These refined reviews are word embedded and representation learning is adopted in detection stage. The detection module consists of deep machine learning and feature-based machine learning classification techniques for building the classifiers using representation learning. Finally, these techniques are ensembled in the ensemble learning module to build a hybrid model for detection of genuine and fake reviews.

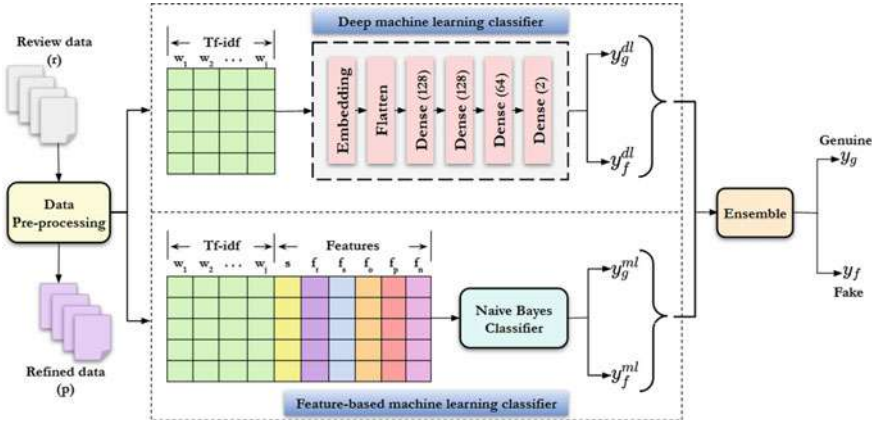


Fig. 1 Proposed hybrid ensemble learning architecture for fake restaurant review detection

3.1 Data Preprocessing

We employ stop word removal and lemmatization techniques to preprocess n text reviews, $r = \{r_1, r_2, \ldots, r_n\}$ and obtain the refined reviews, $p = \{p_1, p_2, \ldots, p_n\}$. Unlike stemming which is commonly adopted in most of the existing works, we lemmatize the refined reviews. Lemmatization is a text normalization technique which generates the root word from the inflected words by considering the morphological analysis. We use lemmatization over-stemming as the semantic information is lost in stemming. Moreover, stemming will produce less accurate results due to under-stemming and over-stemming. To further enhance the text normalization, parts of speech (PoS) tagging is used in order to provide the context to lemmatization and generate the appropriate root words. Table 1 shows how the lemmatization with POS tag modifies individual words based on their POS for a sample text review from Yelp datasets [18]. At the end of data preprocessing, we obtain the preprocessed text reviews p.

Table 1 Sample text review before and after preprocessing from Yelp datasets [18]

Before preprocessing	After preprocessing
Drinks were bad, the hot chocolate was watered down and the latte had a burnt taste to it. The food was also poor quality, but the service was the worst part, their cashier was very rude	Drink be bad the hot chocolate be water down and the latte have a burnt taste to it the food be also poor quality but the service be the bad part their cashier be very rude

3.2 Detection

To detect genuine and fake restaurant reviews, we build deep machine learning and feature-based machine learning classifiers.

3.2.1 Deep Machine Learning Classifier

We use a multilayer perception network to do the classification task. The input to the network would be the preprocessed review data p. Before we feed the preprocessed data to the network, we vectorize it using the count vectorization, keeping the size of our vocabulary to the most frequent words. The count vectorizer provides a sparse matrix where each number represents the frequency of the word f in a review. We normalize this sparse matrix to avoid the classifier to get biased toward some commonly occurring words in a review. In order to do this, we use the tf-idf method, which depicts the importance of words that appears in only few documents.

The term frequency f for each of the word in reviews and their corresponding weights are computed as,

$$\text{tf-idf}(w_j, p_i, C) = f(w_j, p_i) \times \log \frac{n}{|\{p_i \in C : w_j \in p_i\}|} \tag{1}$$

where w_j is the word that occurs in processed review sentence p_i and C is the corpus.

The word embeddings give the numerical encoding obtained from tf-idf represented data from Eq. 1 which is fed to the multilayer perceptron network as shown in Fig. 1. Initially, we embed the input sequence encoding into a 128-dimensional vector. We then use three dense layers, each with ReLU activation function to avoid vanishing gradients problem. The neurons in the fully connected layer might develop co-dependency among each other during training which restrains their power leading to overfitting of the data. To avoid this, we use a dropout layer (with dropout rate of 0.2) after every dense layer, which acts as a regularizer and controls overfitting. Finally, the last layer in the network is the dense layer with sigmoid activation function which classifies the reviews into genuine (y_g^{dl}) and fake (y_f^{dl}) categories. The loss function of the network used is binary cross entropy and the optimizer used is root mean square propagation (RMSprop).

3.2.2 Feature-Based Machine Learning Classifier

To improve the detection of fake reviews, extracting the relevant features which emphasize the contents of the reviews plays a vital role. In our work, for feature-based machine learning, we consider features such as star rating, review length, sentiment deviation, sentiment polarity and positive and negative word count from the refined reviews p [10, 11] described as,

- Star rating s given by the user for each review p_i is directly considered from the data. This idea of predicting by star rating is discussed in [7].
- Review length f_r is the number of words in the review.
- Sentiment deviation f_s represents number of times the review switched from positive word to negative word sentiments and vice-versa.

  The sentiment deviation is given as,

  $$f_s = \sum_{i=1}^{n-1} || \frac{\text{sent}(w_i) - \text{sent}(w_{i+1})}{2} || \tag{2}$$

  where

  $$\text{sent}(w) = \begin{cases} 1 & \text{if } w \text{ is positive} \\ -1 & \text{otherwise} \end{cases}$$

  where w_j represents the individual word in a text review and w_{j+1} is the successive word in the text review.

  We do this using a corpus of all positive sentiment dictionary of words and a negative sentiment dictionary of words.
- Sentiment polarity f_o computes a polarity of a review, which is a sentimental value of a review. We obtain the opinion of each review p_i using the TextBlob, which gives a numerical value in the range of $[-1, 1]$. The TextBlob handles cases like negation, modifier words and computes the polarity based on the average polarity of each individual word.
- Positive word count f_p and negative word count f_n are the number of positive and negative word counts in the review p_i which we find by cross-referencing our dictionary of positive and negative words.

These features are normalized using *min–max* normalization technique. The tf-idf represented data (Eq. 1) and the features $\{s, f_r, f_s, f_o, f_p, f_n\}$ are then fed to the Naive Bayes classifier, which classifies the reviews into genuine (y_g^{ml}) and fake (y_f^{ml}) categories. The Naive Bayes classifier is a popular spam classifier because of the conditional probability model of Bayes theorem. Unlike other classification techniques, Naive Bayes classifier performs well on small dataset, for binary classes with less computational resources.

3.3 Ensemble Learning

Inspired from the works of [19], we propose to use ensemble learning in comparison with other works where the focus was on linguistic, behavioral and graph-based approaches. We build an ensemble model which combines the results of feature-based machine learning and deep machine learning classifiers to obtain an hybrid ensembled model for detection of genuine (y_g) and fake (y_f) reviews. The ensemble learning is carried out to generate a more robust classifier by combining multiple instances of base classifiers. We consider the weighted sum of the base classifiers to build an ensembled model using Eqs. 3 and 4.

$$y_g = 0.5 \times y_g^{\mathrm{dl}} + 0.5 \times y_g^{\mathrm{ml}} \tag{3}$$

where y_g^{dl} is the result of genuine reviews for deep machine learning classifier and y_g^{ml} is the result of genuine reviews for feature-based machine learning classifier.

$$y_f = 0.5 \times y_f^{\mathrm{dl}} + 0.5 \times y_f^{\mathrm{ml}} \tag{4}$$

where y_f^{dl} is the result of fake reviews for deep machine learning classifier and y_f^{ml} is the result of fake reviews for feature-based machine learning classifier.

4 Experiments and Results

We evaluate the proposed fake review detection technique on the Yelp datasets [18]. The proposed technique is implemented on Intel® Xeon(R) processor at 3.5 GHz and 128 GB RAM. The evaluation metrics used are average precision (AP) and area under the curve (AUC) to analyze the performance of the proposed technique. We provide the description of the datasets used in Sect. 4.1. We present results and discussions in Sect. 4.2.

4.1 Datasets

In our work, we consider three datasets from Yelp [18], namely YelpChi, YelpNYC and YelpZip, the summary statistics of these datasets are given in Table 2. The YelpChi dataset consists of reviews from 85 hotels and 130 restaurants in the Chicago area as collected in [14]. The YelpChi contains 13.23% filtered reviews by 20.33% spammers. The YelpNYC contains reviews from 160, 225 reviewers of 923 restaurants. In this dataset, there exists 10.27% filtered reviews by 17.79% spammers. The YelpZip is a larger dataset comprising reviews collected from the regions of U.S. including from NY, NJ, VT, CT, and PA . In this dataset, there exist 13.22% filtered reviews by 23.91% spammers.

Table 2 Total number of genuine and fake reviews from three Yelp review datasets [18]

Datasets	Genuine reviews	Fake reviews	Total
YelpChi	58,479	8916	67,395
YelpNYC	322,177	36,875	359,052
YelpZip	528,141	80,457	608,598

Table 3 Precision, recall, f_1-score and accuracy of the proposed technique on all three yelp datasets

Datasets	Precision	Recall	f_1-score	Accuracy
YelpChi	0.94	0.89	0.90	0.88
YelpNYC	0.96	0.96	0.95	0.95
YelpZip	0.91	0.91	0.91	0.91

Table 4 AP and AUC performance of the proposed technique with the existing methods on all three datasets

Method	AP			AUC		
	YelpChi	YelpNYC	YelpZip	YelpChi	YelpNYC	YelpZip
FraudEagle [1]	0.1067	0.1122	0.1524	0.3735	0.5063	0.5326
Wang et al. [21]	0.1518	0.1255	0.1803	0.5062	0.5415	0.5982
Prior [18]	0.2241	0.1789	0.2352	0.6707	0.6705	0.6838
SpEagle [18]	0.3236	0.2460	0.3319	0.7887	0.7695	**0.7942**
Proposed	**0.5370**	**0.6357**	**0.4544**	**0.9343**	**0.8096**	0.7465

4.2 Results and Discussion

The metrics such as precision, recall, f_1-score and accuracy of the proposed technique on YelpChi, YelpNYC and YelpZip datasets are shown in Table 3. The AP and AUC results of the proposed technique for all three datasets are shown in Table 4. The proposed technique outperforms other existing methods. However, for YelpZip dataset, the AUC is slightly lower compared to that of SpEagle since the false positive rate might be high resulting into underfit. The robustness of the proposed technique signifies that the review text is a crucial feature for fake review detection. Thus, features based on review text perform better than the behavioral features.

5 Conclusions

In this paper, we addressed the problem of fake review detection by proposing a fake review detection technique based on restaurant review text. In the proposed technique, we ensembled feature-based machine learning and deep machine learning classifiers to build a hybrid ensemble model. We evaluated the performance of the proposed technique on three Yelp datasets, namely YelpChi, YelpNYC and YelpZip and obtained accuracies of 88%, 95% and 91%, respectively. The experimental results demonstrated the effectiveness of the proposed technique over the existing approaches. As a future work, the proposed technique can be extended for detecting fake news, movie reviews and products.

References

1. Akoglu, L., Chandy, R., Faloutsos, C.: Opinion fraud detection in online reviews by network effects. In: ICWSM (2013)
2. Barbado, R., Araque, O., Iglesias, C.A.: A framework for fake review detection in online consumer electronics retailers. Inf. Process. Manag. **56**(4), 1234–1244 (2019)
3. Crawford, M., Khoshgoftaar, T., Prusa, J.D., Richter, A.N., Al-Najada, H.: Survey of review spam detection using machine learning techniques. J. Big Data 2 (2015)
4. Deng, X., Chen, R.: Sentiment analysis based online restaurants fake reviews hype detection. In: APWeb Workshops. Lecture Notes in Computer Science, vol. 8710, pp. 1–10. Springer (2014)
5. Fei, G., Mukherjee, A., Liu, B., Hsu, M., Castellanos, M., Ghosh, R.: Exploiting burstiness in reviews for review spammer detection. In: ICWSM (2013)
6. Feng, S., Banerjee, R., Choi, Y.: Syntactic stylometry for deception detection. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers, ACL '12, vol. 2, pp. 171–175 (2012)
7. Ganu, G., Elhadad, N., Marian, A.: Beyond the stars: improving rating predictions using review text content. In: WebDB (2009)
8. Hai, Z., Zhao, P., Cheng, P., Yang, P., Li, X., Li, G.: Deceptive review spam detection via exploiting task relatedness and unlabeled data. In: EMNLP, pp. 1817–1826. The Association for Computational Linguistics (2016)
9. Hájek, P., Barushka, A., Munk, M.: Fake consumer review detection using deep neural networks integrating word embeddings and emotion mining. Neural Comput. Appl. 1–16 (2020)
10. Hegde, S.B., Satyappanavar, S., Setty, S.: Restaurant setup business analysis using yelp dataset. In: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 2342–2348 (2017)
11. Hegde, S.B., Satyappanavar, S., Setty, S.: Sentiment based food classification for restaurant business. In: 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1455–1462 (2018)
12. Inc42: Peoject fairplay zomato fake reviews (2018)
13. Luca, M., Zervas, G.: Fake it till you make it: reputation, competition, and yelp review fraud. Manag. Sci. **62**, 3412–3427 (2016)
14. Mukherjee, A., Venkataraman, V., Liu, B., Glance, N.S.: What yelp fake review filter might be doing? (2013)
15. Ott, M., Cardie, C., Hancock, J.: Estimating the prevalence of deception in online review communities. In: Proceedings of the 21st International Conference on World Wide Web, WWW '12, pp. 201–210 (2012)
16. Ott, M., Choi, Y., Cardie, C., Hancock, J.T.: Finding deceptive opinion spam by any stretch of the imagination. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, HLT '11, vol. 1, pp. 309–319 (2011)
17. Parikh, A., Behnke, C., Vorvoreanu, M., Almanza, B., Nelson, D.: Motives for reading and articulating user-generated restaurant reviews on yelp.com. J. Hosp. Tour. Technol. **5** (2014)

18. Rayana, S., Akoglu, L.: Collective opinion spam detection: bridging review networks and metadata. In: Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15, pp. 985–994 (2015)
19. Roy, A., Basak, K., Ekbal, A., Bhattacharyya, P.: A deep ensemble framework for fake news detection and classification. CoRR abs/1811.04670 (2018)
20. Times, T.: 'A third of TripAdvisor reviews are fake' as cheats buy five stars (2018)
21. Wang, G., Xie, S., Liu, B., Yu, P.S.: Review graph based online store review spammer detection. In: IEEE 11th International Conference on Data Mining, pp. 1242–1247 (2011)
22. Wu, X., Dong, Y., Tao, J., Huang, C., Chawla, N.V.: Reliable fake review detection via modeling temporal and behavioral patterns. In: IEEE International Conference on Big Data (Big Data), pp. 494–499 (2017)
23. Yao, Y., Viswanath, B., Cryan, J., Zheng, H., Zhao, B.Y.: Automated crowdturfing attacks and defenses in online review systems. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS '17, pp. 1143–1158 (2017)

Utilizing Corpus Statistics for Assamese Word Sense Disambiguation

Nomi Baruah, Arjun Gogoi, Shikhar Kr. Sarma, and Randeep Borah

Abstract Classification or categorization of a word based on its meaning in respect to a context is one of the major problems in Natural Language Processing (NLP). Such a problem is termed as Word Sense Disambiguation (WSD), and the mentioned problem is seen to be prevalent in all languages across the globe. However, in Indian languages, WSD poses greater challenges due to limitation of digital resources and lack of UNICODE. In this paper, we have made an attempt to highlight the efforts put by researchers to overcome WSD. It is also to be mentioned that for the said purpose, two WSD algorithms for Assamese language WSD are contrasted while asserting the corpus statistics in the approach. Of the two aforementioned WSD algorithms, the first is applied using the Lesk algorithm simpler, while the second is exercised to determine the probability of words and phrases on grounds of condition that co-occur with every meaning of an ambiguous word in disambiguation. Both the algorithms delivered affirmative results for a trained set of corpus. However, compared to the second, the Lesk algorithm yielded better results in terms of overall efficiency of the system developed in comparison to words and phrases co-occurrence.

Keywords Assamese · Corpus · Wordnet · Words sense disambiguation

N. Baruah (✉) · A. Gogoi · R. Borah
Dibrugarh University, Dibrugarh 786004, India
e-mail: baruahnomi@gmail.com

A. Gogoi
e-mail: gogoiarjun27@gmail.com

R. Borah
e-mail: randeep89borah@gmail.com

S. Kr. Sarma
Gauhati University, Guwahati 781014, India
e-mail: sks001@gmail.com

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_23

1 Introduction

Word Sense Disambiguation (WSD) is defined as the inference of meaning of a word made available by a computer in reference to the targeted context. Almost in every language across the globe, one would find hundreds of words that are in usage and carry more than one meaning. Because of such ambiguity of word meaning, the computers find it difficult to identify the correct meaning in a given context or input content. For instance, the word "way" used in a sentence "It is only the way to reach there" indicates a method, style or manner of doing something, whereas in another sentence, the word "way" is used as "I am on my way to home". Here the word "way" means a direction or progress on a work. The ambiguities are understood by humans due to semantics of language and syntax. But in case of computers, acquisition of knowledge about language is essential or vital to carry out WSD task and because of such reason, WSD faces an artificial intelligence problem [1]. WSD is classified into two variations—lexical and all-word. The lexical variant is designed to remove unpredictable meanings of a single word targeted in a sentence, whereas the all-word variant disambiguates every word present in the targeted text. However, the all-word variant suffers from data sparseness problem as the classifiers are unable to cover the whole text. But compared to all-word, the performance of lexical is better as it targets limited ambiguous words.

Researchers had done researches mainly on supervised, unsupervised and knowledge-based methods to evaluate the performance of WSD. Among all the proposed methods, the supervised method is considered to be better. In supervised method, data is set and tagged manually so as to instruct the classifier to evaluate the context of the targeted ambiguous word and identify its meaning. But this method is language-specific and involves much time [2]. Supervised WSD in English language proved to be successful and is developed using different classified algorithms like Naïve Bayes (NB) [2], decision trees (DT) [3], Support Vector Machine (SVM) [4] and K-Nearest Neighbour (K-NN) [4]. But it has been observed that work on WSD in Indian language is still in an early stage.

In unsupervised method, annotated datasets are created based on the assumptions having similar senses that occur in a document. The task is done manually by mapping the clusters of words carrying meanings of sense inventory.

Knowledge base methods are applied by using an external lexical resource, such as dictionary, thesaurus, semantic networks ad tagged corpus, and are rivaling supervised methods because of its wide coverage [4].

The paper is described as follows: Sect. 2 describes the challenges in Indian languages WSD, Sect. 3 describes the related work, Sect. 4 describes the methodology, Sect. 5 discusses the dataset and experiments, Sect. 6 discusses the observations found while doing the work, and Sect. 7 discusses the conclusion.

2 Challenges in Indian Languages WSD

Implementation of WSD in Indian languages has made no significant development because of various challenges. For the last several years, researchers have developed WordNet for different languages and the same is considered as the sense inventory of choice as it allows to compare the results of various research groups and give the provision of a large conceptual network that can support the disambiguation process and manually tagged corpora. But as the catalogue of meaning involves a great attention to detail it makes disambiguation challenging.

ML community also proposed methods to solve WSD problem: Naive Bayes, Decision Lists, AdaBoost, Support Vector Machines (SVMs), etc. are a few to be cited in this reference. On the other hand, results derived based on comparison exhibit that even the highly sophisticated methods are unable to provide accurate solutions to the problems. Such an exercise is done usually by applying different words and different algorithms to achieve the best results. Many experts hold the opinion that optimization of ML methods and determining features per word might be helpful in solving the problem.

Certain feature sets are considered as model of WSD in Indian language, but the drawback is that sets used for this purpose are too limited. The reason is that conventional basic feature sets consisting of bigrams, trigrams, and "bags of words" are applied to develop the contexts of the words targeted. However, for better results, it is observed that ML methods may be considered based on the information from the texts. For this objective, the characteristics derived by analyzing the text such as morphological, syntactic, semantic, domain, etc. could be used. Moreover, the combination of different types of features may also be considered.

For high-tech systems considering the existing hand-tagged corpora are not only insufficient but also expensive and hard to obtain. However, the prevalent hand-tagged corpora does not support fully for updated systems. The outcome of the tagging is also not optimistic and also methods that are considered to derive data automatically are lagging behind in terms of quality compared to hand-tagged data. Apart from this, the results accounted generally in the literature are for the words having training examples, but it is recommended that a WSD tool includes all the words present in the vocabulary (Table 1).

3 Related Work

Sarmah and Sarma proposed "Word Sense Disambiguation for Assamese" [5]. In this paper, Naive Bayes (ND) classifier was used for the purpose. The classifier applied used lexical features. In this research, it is seen that the baseline method enhanced the classifier by providing results with accuracy of 78%. Moreover, a sense inventory is obtained from two significant NLP resources—Assamese WordNet and Corpus that aid annotators in the work of tagging sense. A total of 160 ambiguous words are

Table 1 Comparison of Word Sense Disambiguation in Indian languages

Language	Language family	Approaches used	Comments
Assamese [5]	Indo-European Indo-Aryan Indo-Iranian Eastern zone Kamarupic Bengali–Assamese Eastern Kamarupic	The supervised technique	The accuracy is calculated using precision and recall and F-measure
Assamese [6]	Indo-European Indo-Aryan Indo-Iranian Eastern zone Kamarupic Bengali–Assamese Eastern Kamarupic	Supervised machine learning approach—decision tree	The accuracy is calculated based on using average F-measure of 0.611 when tenfold cross-validation evaluation was performed on ten Assamese ambiguous words
Kannada [7]	Dravidian, Southern Tamil–Kannada Kannada–Badaga Kannada	A suite of Perl programs used to disambiguate the polysemous Kannada words	All major grammatical categories of words have been covered. Tenfold cross-validation has been performed in all test cases
Hindi [8]	Indo-European Indo-Iranian Indo-Aryan Central zone Western Hindi Hindi Hindustani	(1) Training phase approach— (a) Training documents (b) Hyperspace analog to language. (c) Dimension reduction of the matrix. (d) Normalization (e) Fuzzy C-means clustering (2) Testing phase	Tested across various domains of the Hindi words. The probabilistic latent semantic analysis for unsupervised WSD—74.12%
Telugu [9]	Dravidian South-central Telugu kui Proto–Telugu	Unsupervised learning using decision list	The accuracy is calculated using precision and recall
Telugu [10]	Dravidian South central Telugu kui Proto–Telugu	N-Gram-based technique	The accuracy calculated using precision, recall, F-measure
Hindi [11]	Indo-European Indo-Iranian Indo-Aryan Central zone Western Hindi Hindustani Hindi	The supervised approach, unsupervised approach and the domain-specific sense with the help of knowledge-based methods	Result accuracy WSD system is compared with the gold standards to calculate the accuracy of WSD

(continued)

Table 1 (continued)

Language	Language family	Approaches used	Comments
Bengali [12]	Indo-European Indo-Iranian Indo-Aryan Magadhi Prakrit Eastern zone	The supervised approached Naive Bayes classifier	Accuracy calculated using lemmatized and bootstrapping
Bengali [13]	Indo-European Indo-Iranian Indo-Aryan Magadhi Prakrit Eastern zone	The unsupervised approach clustering (maximum entropy method)	Accuracy calculated using PCA and context expansion
Punjabi [14]	Indo-European Indo-Iranian Indo-Aryan Northwestern Indo-Aryan	The neural network technique LSTM with accuracy of 84%	Accuracy metric Acc = C_i $\{i \in (1, 2, \ldots)/n\}$
Gujarati [15]	Indo-European Indo-Iranian Indo-Aryan Western zone	(1) Knowledgebase approach (2) Deep learning approach (3) Supervised technique: (a) Naive Bayes classifier (b) Decision list (4) Unsupervised techniques: (a) Context clustering (b) Word clustering algorithm used LSTM model	Accuracy is calculated and Hyperlex 73% has the highest accuracy
Gujarati [16]	Indo-European Indo-Iranian Indo-Aryan Western zone	Knowledge-based approach	It can also be made generalized for Indo–Aryan languages where the Indo–Aryan WordNet for different languages can be used to disambiguate the ambiguous word

included in sense inventory, and the system used in this process is applied into two phases. In the primary phase, 2.7 k sense annotated training data and 800 test data were evaluated, and accuracy of 71% was derived in the subsequent result. In the second phase, the results of the primary phase are validated manually and to the prior training data the clean sense tagged data were augmented. An additional accuracy of 7% was achieved by using an iterative learning by the system.

Sarmah and Sarma proposed "Decision Tree based Supervised Word Sense Disambiguation for Assamese" [6]. Decision Tree (DT) refers to a decision model flowchart like tree structure where the internal node, each branch and each leaf denotes a test, results of a test and a sense label, respectively. In this research, as a part of training and test dataset, a few polysemous words having different real occurrences in Assamese text with manual sense annotation were collected. It was observed that DT algorithm generated average F-measure of 0.611 after evaluation of tenfold cross-validation was conducted on 10 Assamese ambiguous words.

Parameswarappa and Narayana put forwarded “Kannada Word Sense Disambiguation for Machine Translation” [7]. In this research, to work on WSD task an Integrated Kannada WSD system including a suite of providing better performance Natural Language Processing (NLP) modules was implemented in Perl. Using the web, the corpus builder module constructed the raw Kannada corpora. However, the method for conducting a test bed for disambiguation put to use certain sentences that were randomly selected from the corpora. Moreover, for this purpose, Dictionary builder module using corpora built the electronic machine readable dictionary. In the sentences, the polysemous verb is disambiguated using Verb Sense Disambiguation module. During the entire process, it was observed that the system efficiency is extendable and reliable.

Tayal, Ahuja and Chhabra proposed “Word Sense Disambiguation in Hindi language using Hyperspace Analogue to Language and Fuzzy C-Means Clustering” [8]. In order to train the system, they converted the Hindi language text into Hyperspace Analogue to Language (HAL) vectors and mapped every word into a high-dimensional space. To form clusters representing different context in which polysemous word occur the Fuzzy C-Means Clustering algorithm is used. The test data used is then categorized into the high-dimensional space which was created during the training phase.

Palanati and Kolikipogu proposed “Decision List Algorithm for Word Sense Disambiguation for Telugu Natural Language Processing” [9]. They conducted the research by assigning a specific sense for a word depending on the use of context. The probability of co-occurrence of word over the corpus is evaluated by using Decision list algorithm which provided accuracy about 90% for English text corpus. Then the sense is assigned and the same is considered to disambiguate sense of Telugu words. In the performance, recovery of Telugu information is evaluated both without and with Word Sense Disambiguation. And it was observed that the results yielded are more accurate with WSD when contrasted to word sense ambiguity.

Prasad, Sunitha and Rani proposed “Word N-Gram Based Approach for Word Sense Disambiguation in Telugu Natural Language Processing” [10]. The approach put forward provided an answer to the problem of WSD, and it was done considering the help of n-gram technique that has yielded better results in other languages. It identified the words that co-occurred to the target polysemous word termed as *n*-grams. Moreover, it was observed that a Telugu corpus conveyed as input for training phase so as to search n-gram joint probabilities. These joint probabilities take into account the intended polysemous word to allot a correct sense in testing phase. The entire method was assessed on certain polysemous Telugu verbs and nouns, and while testing on Telugu corpus, it was observed that it yielded F-measure of 0.94. The Telugu corpus was compiled from story books, newspapers, CIIL, etc.

Mall and Jaiswal proposed "Word sense disambiguation in Hindi applied to Hindi-English machine translation" [11]. The work is aimed to assess the use of parsing for machine translation from Hindi to English language. It has been observed that no earlier research has been carried in literature to assess the translation of Hindi into English translation for Indian language, and thus, this research is considered as pioneer. The work of Mall and Jaiswal confirmed that the parts-of-speech tagging algorithm obtained 92.09% accuracy result. In this research, the modified Lesk algorithms are used to overlap the finding between three pieces of words in a given context and to identify the correct word sense. The process is done by counting word overlaps between glosses of the words in the context. And all the glosses of the targeted keyword are then contrasted with the glosses of the other words. Based on the assumption, it is seen that the senses for which the most numbers of overlaps come about characterize the desired sense of the polysemy word. In this research, both English and Hindi WordNet are used in lexical knowledge. The use of modified Lesk algorithm has improved word sense disambiguation, and the result generated by the system is compared with Google Translator. The comparison also showed that Google Translator cannot handle sense disambiguation of words, but the proposed work resolves WSD.

Das and Sarkar proposed "Word Sense Disambiguation in Bengali applied to Bengali-Hindi Machine Translation" [12] to obtain accurate lexical choice in Bengali–Hindu translation. In this method, they applied a graph-based methodology to sort out sense clusters in Bengali language and the reason is that no annotated Bengali corpus or parallel corpus for Bengali–Hindu pair is available. Accordingly, a vector space-based approach was considered to locate the sense clusters to Hindi translation of the words targeted and also to predict translation of the target word. In this research, Das and Sarkar also considered the use of bilingual sense dictionary, Monolingual Bengali corpora, monolingual Hindi corpora, Bengali WordNet and Hindi WordNet.

Pal and Saha proposed "Word Sense Disambiguation in Bengali language using unsupervised methodology with modifications" [13]. The research was conducted in two phases. In the first phase, using maximum entropy method sentence clustering is carried out and then the clusters are categorized with innate senses. The process of labeling is done by manual intervention because sense-tagged clusters could be used as sense inventories for further experiment. In the next phase, cosine similarity measure is applied to search the closeness of the test disambiguated data with the earlier sense-tagged clusters. And the least distance of the test data from a specific sense-tagged cluster allocate similar sense to the test data alike the cluster it is assigned with.

Singh and Kumar proposed "Sense disambiguation for Punjabi language using supervised machine learning techniques" [14]. They analyzed an explicit WSD system for Punjabi language considering supervised techniques by manually creating a sense tagged corpus of 150 ambiguous Punjabi noun words and six supervised machine learning techniques, and then, they are put to work to investigate for the objective. For this purpose, classifier was used and each classifier made use of similar feature space compassing lexical (unigram, bigram, collocations and co-occurrence) and syntactic (parts-of-speech) count-based features. The semantic traits of Punjabi language from the unlabeled Punjabi Wikipedia text was also considered in this research, and it was done by considering word2vec bag of word in addition to skip gram shallow neural network models. For WSD, Singh and Kumar further applied two deep learning neural network classifiers multilayer perceptron and long short-term memory.

Vyas and Ganatra in their paper proposed "Gujarati Language Model: Word Sense Disambiguation using Supervised Technique" [15]. They put forwarded two advanced methodologies for WSD: knowledge-based and deep learning approach. The latter methodology is observed to perform better irrespective of its shortcoming and is important as colossal data sources which is used to select the implications of words in a specific setting.

Vaishnav and Sajja proposed "Knowledge-Based Approach for Word Sense Disambiguation Using Genetic Algorithm for Gujarati" [16]. For their objective, they used genetic algorithms (GAs). The problem of word sense disambiguation attempted to identify the meaning of a polysemous word so that it could be applied in a surrounding context to a provided text.

4 Methodology

4.1 Simplified Lesk Algorithm

The Lesk algorithm [17], a classification of overlap-based approaches, establishes co-relation between the context bag and sense bag. From the context of the source document the context bag is developed, whereas the sense bag is developed from the gloss using the ambiguous words accumulated in the WordNet. Similarly, the ambiguous words of the input document remain to be a prime source in context bag. In this research, the sense having the highest number of overlapping between context bag and sense bag is denoted as winner sense for the targeted word. The context of the targeted word is considered as a list of words that shows in $a \pm n$ window size, having the specific word amidst. For a window size of n, the context vector size is determined as $2n + 1$.

Let's assume W represents an ambiguous word and C denotes a collection of a set of context words in its adjoining window. Taking the help of the Assamese WordNet seek for the word W and try to find the senses of W. The result derived as such indicates a lot of senses S for W of words. Subsequently, Lesk algorithm is applied to determine the best sense of the word W. So, it is observed that the output derived will indicate the sense as the most probable one having the maximum number of overlapping. A list of stop words developed in NLP lab of Gauhati University is considered. A test is run indicating the targeted word. The authors had chosen Lesk algorithm because work on knowledge-based approach in Assamese language is very limited. Apart from them Lesk algorithm is simple which makes it easy and fast to apply in any language.

Algorithm

- User Input for Assamese Tagged sentences.
- Eliminating stop words from the user input.
- Developing context vector from the selected word (SW) and the same is done by deriving all the existing words in each side of the word in proximity of n words.
- Presume the number of senses of SW is *m*.
- For $x = 1$ to m do.

  Of the target word (T_t) sense definition vector is developed for sense *x*.

- $\text{Score}_x \leftarrow \text{Score}_x + \text{Score}_{sw}$
- Sense having the highest Score_x is selected.

4.2 *WSD Algorithm Using Word Co-occurrence*

WSD algorithm helps in getting rid of the ambiguity of a word while taking into consideration the phrases and co-occurrence words, wherein the co-occurrence statistics is gathered based on a sense tagged training corpus. In this experiment, we have put to use a preset size window that surrounds an ambiguous word to obtain phrases and co-occurrence words. For the said purpose, two words to the right and left of an ambiguous word W are considered. We have taken note of each word that appears in $a \pm 2$ window of the word W and also appearance of the word in terms of frequency with each senses of *w*.

<হিন্দু সকলৰ বাবে কাশী এক পৱিত্ৰ স্থান |>, {Hindu Hokolor Babe Kakhi Ek Pabitro Isthan}, {For the people of Hinduism community, Kaashi is a Holy place}.

In the word co-occurrence method, the above input sentence has been taken into token of word co-occurrence phrases as follows:

"হিন্দু সকলৰ","সকলৰ বাবে","বাবে কাশী","কাশী এক","হিন্দু সকলৰ বাবে কাশী এক".

5 Dataset and Experiments

We, to evaluate two algorithms, developed and applied a sense annotated Assamese corpus as shown in Table 2. The sense inventory used in this research has been derived from Assamese WordNet [18]. Senses that have very fine-grained sense distinctions

Table 2 Details of the corpus

Details	Values
Total number of words	50,001
Ambiguous words (nouns)	15,606
Total number of unique words	12,282
Total number of instances	12,282
Total number of word senses	50,001
Total number of instances per word	4.07
Total number of senses per word	1.0

are combined and instances of the senses with lower rate of use could not be found as a result they are being dropped by us.

Using the terms of precision, recall and F-measure metrics, the evaluated performance is measured as defined in [19].

$$\begin{aligned}\text{Precision} &= \text{Number of correct senses predicted}\\ &\quad /(\text{Number of words provided})\end{aligned}$$

$$\begin{aligned}\text{Recall} &= \text{Number of correct senses predicted}\\ &\quad /(\text{Total number of words in the test set})\end{aligned}$$

$$\begin{aligned}\text{F - Measure} &= 2 * \text{Precision} * \ \text{Recall}\\ &\quad /(\text{Precision} + \text{Recall})\end{aligned}$$

5.1 Experiment 1

It is performed applying simplified Lesk algorithm. The score of accuracy, precision, recall and F-measure is presented in Table 3.

Table 3 Evaluation scores of simplified Lesk algorithm

Evaluation metric	Values (%)
Avg accuracy	73
Avg precision	80
Avg recall	76
Avg F-measure	77.95

Table 4 Evaluation scores of word co-occurrence

Evaluation metric	Values (%)
Avg precision	79
Avg recall	74
Avg F-measure	76.42

5.2 *Experiment 2*

It is measured using word co-occurrence. The word co-occurrence is done for 200 words taking window size 2. The score of precision, recall and F-measure is presented in Table 4.

5.3 *Experiment 3*

In contrast to other Approaches (Fig. 1).

Both the methods, i.e., Lesk algorithm and word co-occurrence, fall under the same category, i.e., dictionary-based approach. Here in experiment 3, we had compared with the latest approaches that fall under dictionary-based approach. Observation regarding the performance of Lesk algorithm approach is made and found that it excels to the approach proposed by Borah et al. [20] by 2.45% and also algorithm with word co-occurrence enhances by 1.53%, respectively. It is also to be mentioned that word co-occurrence algorithm was considered for 135 words by Borah et al. [20], whereas we had done it for 200 words.

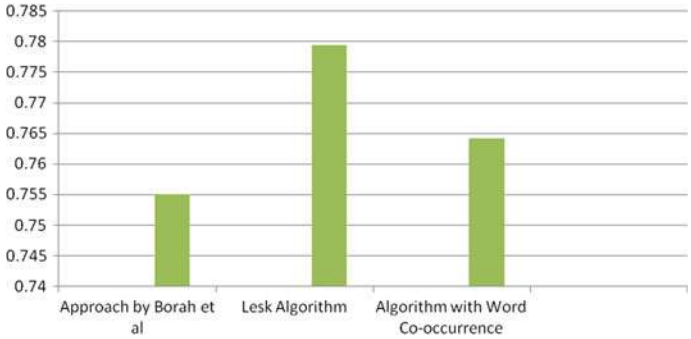


Fig. 1 Comparison of the approaches

6 A Few Close Observations

6.1 Very Short Sentence

As a good number of sentences are too short in length, thus the proposed system could not retrieve sufficient information from them.

6.2 Spelling Error

In a few cases, spelling errors in words impeded the performance of the system.

6.3 Scarcity of Information in Assamese WordNet

In this dictionary, synonymous sense definitions of the common Assamese ambiguous words are absent. However, a few sense definitions are detected in the WordNet, but the same are missing in the standard lexical dictionary.

7 Conclusion

In this research, a review related to the field of word disambiguation is presented in various Indian languages so that further researches can be done. The researches in Indian languages considered various datasets of limited size but due to the unavailability it makes difficult to consider the results. Moreover, the use of WordNet in various Indian languages has several drawbacks such as precision and limited coverage compared to English WordNet.

It is also observed that knowledge-based approaches provide higher precision, but the performance depends on dictionary definitions. It is also to be noted that unsupervised approaches do not require sense annotated corpora, but they lack in performance compared to knowledge-based approaches. Hybrid approaches are also applied with the objective to benefit from their component methods and yield better results.

WSD (i.e., Lesk algorithm and word co-occurrence algorithm) is proposed in Assamese language, and after analysis it is observed that Lesk algorithm outperforms the word co-occurrence algorithm. In the coming days, it may be assumed that scalability of the proposed approach may be enhanced by augmenting more ambiguous Assamese words. Moreover, the proposed research can also be integrated with other Assamese NLP applications to add to their respective performance. Further work on other knowledge-based approach is undergoing in this language such as Walker method, etc. and also in supervised methods of WSD.

References

1. Navigli, R.: Word sense disambiguation: a survey. ACM Comput. Surv. **41**(2):1–69 (2009)
2. Agirre, E., Edmonds, P.: Word sense disambiguation: algorithms and applications, vol. 33. Springer Science & Business Media, New York (2007)
3. Zhong, Z., Ng, H.T.: It makes sense: a wide-coverage word sense disambiguation system for free text. In: Proceedings of the 48th ACL, pp. 78–83 (2010)
4. Agirre, E., Lacalle, O.L., Soroa, A.: Random walks for knowledge-based word sense disambiguation. Comput. Linguist. **40**(1), 57–84 (2014)
5. Sarmah, J., Sarma, S.K.: Word sense disambiguation for assamese. In: Proceedings 6th International Conference on Advanced Computing, pp. 146–151 (2016)
6. Sarmah, J., Sarma, S.K.: Decision tree based supervised word sense disambiguation for assamese. Int. J. Comput. Appl. **141**(1), 42–48 (2016)
7. Parameswarappa, S., Narayana, V.N.: Kannada word sense disambiguation for machine translation. Int. J. Comput. Appl. **34**(10), 1–8 (2011)
8. Tayal, D.K., Leena, A., Chhabra, S.: Word sense disambiguation in Hindi language using hyperspace analogue to language and fuzzy C-means clustering. In: Proceedings of the 12th International Conference on Natural Language Processing, pp. 49–58 (2015)
9. Palanti, D.P., Kolikipogu, R.: Decision list algorithm for word sense disambiguation for Telugu natural language processing. Int. J. Electron. Commun. Comput. Eng. **4**(6), 176–178 (2013)
10. Prasad, P.D., Sunitha, K.V.N., Rani, B.P.: Word N-gram based approach for word sense disambiguation in Telugu natural language processing. Int. J. Rec. Technol. Eng. **7**(6S4), 686–690 (2019)
11. Mall, S., Jaiswal, U.C.: Word sense disambiguation in Hindi applied to Hindi-English machine translation. Comput. Model. New Technol. **21**(2), 58–68 (2017)
12. Das, A., Sarkar, S.: Word sense disambiguation in Bengali applied to Bengali-Hindi machine translation. In: International Conference on Natural Language Processing (2013)
13. Pal, A.R., Saha, S.: Word sense disambiguation in Bengali language using unsupervised methodology with modifications. Sadhana **44**, 168 (2019)
14. Singh, V.P., Kumar, P.: Sense disambiguation for Punjabi language using supervised machine learning techniques. Sadhana **44**, 226 (2019)
15. Vyas, T., Ganatra, A.: Gujarati language model: word sense disambiguation using supervised technique. Int. J. Rec. Technol. Eng. **8**(2), 3740–3744 (2019)
16. Vaishnav, Z.B., Sajja, P.S.: Knowledge-based approach for word sense disambiguation using genetic algorithm for Gujarati. In: Satapathy, S., Joshi, A. (eds.) Information and Communication Technology for Intelligent Systems. Smart Innovation, Systems and Technologies. Springer, Singapore (2019)
17. Pal, A.K., Kundu, A., Singh, A., Shekhar, R.: An approach to word sense disambiguation combining modified lesk and bag-of-words. Comput. Sci. Inform. Technol. **3**, 517–524 (2013)
18. Sarma, S.K., Medhi, R., Gogoi, M., Saikia, U.: Foundation and structure of developing an assamese WordNet. In: The 5th International Global WordNet Conference (GWC 10) (2010)
19. Singh, S., Siddiqui, T.: Utilizing corpus statistics for Hindi word sense disambiguation. Int. Arab J. Inform. Technol. **12**(6A), 755–763 (2015)
20. Borah, P.P., Talukdar, G., Baruah, A.: WSD for assamese language. In: Kalita, J., Balas, V., Borah, S., Pradhan, R. (eds.) Recent Developments in Machine Learning and Data Analytics. Advances in Intelligent Systems and Computing, vol. 740. Springer, Singapore (2019)

A Novel Approach to Text Summarisation Using Topic Modelling and Noun Phrase Extraction

Nikhil M. Lal, S. Krishnanunni, Vishnu Vijayakumar, N. Vaishnavi, S. Siji Rani, and K. Deepa Raj

Abstract Over the past few years, one of the remarkable developments that happened on the web is the rapid growth of textual data. This substantial increase, however, induces a complication in the retrieval of vital information from the digitized collection of data. The conventional technique used to tackle this problem is Automatic Text Summarisation. This technique extracts the essential words or sentences from the data and summarises it without affecting the semantics. Automatic text summarisation is classified into two, Extractive and Abstractive. The Extractive method summarises a document by selecting the important words or sentences from it, based on some attributes while the Abstractive method attempts to generate its summary from the semantics of the data. In this paper, we propose a novel approach in Extractive text summarisation by using a new sentence scoring parameter. The experimental results show that the proposed sentence scoring parameter improves the performance of the Extractive text summariser, when compared with other summarisation models. To validate our proposed model, we compared it with four commonly used summarisation models on grounds of ROUGE-1 score and F1 score.

Keywords Extractive text summarisation · Abstractive text summarisation · Topic modelling · Noun phrase extraction · Natural language processing

1 Introduction

In this world of digitized data, the volume of textual information available empowers the development of efficient computation models in the research community. But the extraction of essential information using information retrieval techniques from a vast collection of textual documents is constantly challenging [1]. Nowadays, it is

N. M. Lal (✉) · S. Krishnanunni · V. Vijayakumar · N. Vaishnavi · S. Siji Rani · K. Deepa Raj
Department of Computer Science and Engineering,
Amrita Vishwa Vidyapeetham, Amritapuri 690546, India
e-mail: nikhilmlal@gmail.com

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_24

common that for a particular field, there is an extensive number of articles present on the internet. In this scenario, it is essential to develop a new computational model that accelerates the process of information retrieval in a systematic manner.

Text summarisation is a method to create a clear-cut summary of multiple documents while focusing on the pre-defined semantics of the text in an efficient manner. The objective of summarisation tools is to transform single or multiple documents into shortened versions using computation models, as the manual implementation of this is a time consuming and expensive process [2]. Automatic text summarisation is a natural language processing technique, wherein computers will analyse and understand the overall meaning of the source text and forms a shorter version of it. It is also an application of machine learning and data mining [3].

Automatic text summarisation is classified into extraction-based and abstraction-based summarisation. Extractive methods work by selecting a subset of existing words, phrases, or sentences in the original text to form the summary [4]. In contrast, Abstractive approaches generate an internal semantic representation and then use deep learning networks to construct a summary that is similar to what a human might express [5]. Such a summary might include verbal innovations as well.

The Extractive summarisation models are easy to implement and do not require any training before execution. On the other hand, Abstractive models could only be implemented using deep learning techniques [5] and demand high computing power and training time to setup. Implementation of an optimized text summariser with computational simplicity is still being researched upon.

Natural Language Processing (NLP) [6] is a field that is home to many types of research. Most of them are made possible using techniques like topic modelling, word vectorisation etc. These methods have a lot of scope in NLP as they can serve for almost all the language evaluation techniques.

For a better understanding of the concepts expressed in a text, most Abstractive text summarisation algorithms use pre-trained word vectors. Word vectors (also known as word embeddings) are learned in-order to collect all of the conceptual, hierarchical, and relational bits of knowledge in a word. The most effective and commonly-used techniques for word vectorisation are Word2Vec and GloVe [7].

Topic modelling [8] is an unsupervised machine learning technique that's capable of scanning a large body of text to identify a single or a set of words that can best characterize the text. It has got many applications when combined with text summarisation. Company executives, consultants, paralegals, and university scholars need to comb through the enormous number of documents to keep ahead, so a good portion of their time is consumed only by finding out what document is relevant and what is not. It is, therefore, possible to quickly assess whether or not a document is worth reading by extracting its important sentences or by finding out the topics in which document talks about.

The Extractive text summarisers currently being used only evaluates the statistical features in a sentence. In researches like [9, 10] features like position, digit count, noun count etc. were used as additional attributes to improve the efficiency. This makes it clear that many factors can be further added to improve the overall performance of the Extractive summariser model. Thus, it is evident that Extractive summarisers still need to be optimised in terms of attributes in-order to extract sentences efficiently.

In this paper, we demonstrate an Extractive text summariser which uses an additional sentence scoring parameter with the help of pre-trained word vectors and a topic modeller. The model is compared with four other summarisation methods like Statistical scoring method, Graph-based method, Fuzzy logic method and Deep Learning method. The experimental results, when run on three datasets show that our proposed model can outrun the other summarisation techniques with the help of the new sentence scoring parameter.

The rest of the paper is organized as following: Sect. 2 describes the Literature review talking about the various researches done in the field of Text Summarisation, Sect. 3 describes the techniques used to model the Proposed method and Sect. 4 presents Experimental Results and Evaluation. We conclude the paper in Sect. 5.

The concept was formulated by Nikhil M. Lal under the leadership of Siji Rani S. and Deepa Raj K., with implementation primarily performed by Nikhil M. Lal and Krishnanunni S. Appropriate models were selected and tested by Vishnu Vijayakumar, as per testing scheme proposed by Vaishnavi N.

2 Literature Review

Over the last few years, the research area has raised many advanced models for summarising text documents. In [11], Patel and Tabrizi surveyed 50 research papers related to automatic text summarisation, which were published from the year 2012 to 2019. With this study, the authors proved how vastly the text summarisation algorithms were researched. From this survey, the need to update summarisation methodologies to tackle the ever-growing big data is understood.

It is evident from the analysis [11] that Extractive summarisation is the field that has been mainly studied in contrast with Abstractive summarisation, in recent years. Several studies were conducted to find the optimal set of parameters that can be used for sentence scoring in Extractive summarisation. In [9, 12], the authors implemented Extractive text summarisers using a set of many sentences scoring parameters and proved their performance benefits.

In [4], the authors used TF-IDF as the only parameter and proved how effective the summarisation model is when a lesser number of attributes are used for sentence scoring. Therefore, it is understood that using additional parameters for sentence scoring does not always enhance the model's performance unless the parameter is efficient in evaluating the significance of a sentence in the text.

Sentence scoring parameters are the factors with which a summariser identifies the importance of a sentence in the text. These parameters are of different types: statistical, semantical etc. and can be utilised in many ways. Statistical scoring parameters are the ones which are calculated by evaluating the sentences based on its statistical features. In [9], the authors implemented an Extractive summarisation model using a set of four statistical parameters, namely Title score, Length score, Position score and Keyword-frequency score. In contrast, in [4], an Extractive summariser was implemented by using only TF-IDF as the parameter. The statistical features extracted were directly used as a scoring parameter in both of the above studies [4, 9]. The related works thus prove how efficient the statistical parameters are in sentence scoring, but fail to compare the performance of the model with those using other scoring parameters.

Clustering is an unsupervised machine learning method to classify data based on some attributes. In Clustering, the attributes used for the summarisation technique are the statistical features extracted [13]. The sentences are clustered mainly into two categories, the primary and secondary. The primary cluster contains the relevant data of the document, which will be the part of the summary and secondary cluster contains the irrelevant data. In [3, 14–16] the authors used a Clustering method for sentence scoring and the summary was formed by combining the sentences in the primary cluster.

Similarly, several other studies were conducted to discover new ways of utilising the statistical parameters. In [12, 17], the authors used a Fuzzy logic system and a Graph-based system respectively to score sentences. A Fuzzy logic system is a method which calculates a score between 0 and 1, based on a set of Fuzzy rules. In [12], Mandal et al. used a set of eight attributes for the Fuzzy system to calculate the score.

A Graph method, on the other hand, is a text evaluation technique which transforms a document into a graph, where each sentence in the document is represented by the nodes of the graph. The summary is formed by selecting the sentences based on their similarity score. In [2], the author proposed a graph model based summarisation in near-linear time using the web-pages as documents. Thus, the researches on Clustering, Fuzzy logic and Graph-based models prove how efficient different scoring approaches are to evaluate sentences.

Similar to the Extractive methods, Abstractive summarisation methods were also studied vastly. In [5], Nallapati et al. built a text summarisation model using complex deep learning networks. They also added an attention layer to further enhance the model's performance. The authors were able to prove that the model can outperform the state-of-the-art summarisers available.

From the above literature survey, it is understandable that in the field of Extractive text summarisation, the most commonly used method is the statistical scoring method. It is also conceivable from the survey that the proper combination of sentence scoring parameters is still being researched upon.

3 Proposed Model

It is observable from the Literature survey that, Extractive text summarisation methods currently being used only evaluate the statistical features of a text for sentence scoring. Abstractive summarisation methods on the other hand are difficult to imple-

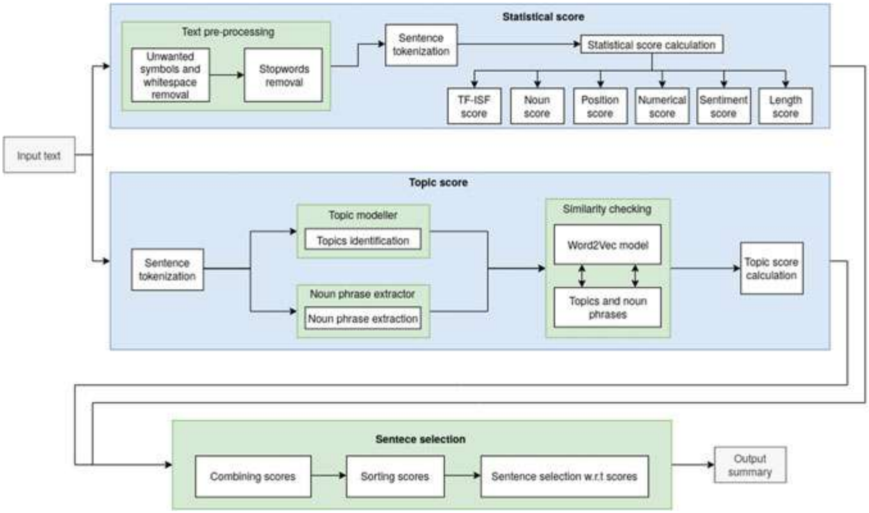


Fig. 1 Proposed model (block diagram)

ment and requires high computing power. In this paper, the proposed approach is developed based on Extractive text summarisation. The proposed model, as depicted in Fig. 1, consist of two additional elements apart from an Extractive summarisation system: a topic modelling system and a pre-trained word vector. These elements are introduced to make the model further efficient without making it computationally complex.

Topic modelling: It is an unsupervised machine learning technique that is capable of scanning a set of documents to detect word and phrase patterns within them, and to automatically cluster word groups and similar expressions that best characterizes a set of document. Topic modelling helps in analysing and identifying a large amount of text data, word clusters, document similarities and abstract topics.

Word Vectors: A deep learning model cannot evaluate text data in the raw form. This is where vectorising methods like Word2Vec comes into play. According to [18] word vectors made a significant leap forward in text evaluation, as they aid machines by providing maximum information about words. They are also the root reason why technologies like speech recognition and machine translation are possible.

3.1 Score Calculation Method

As the proposed model is an extraction-based summariser, the sentences to be included in the summary are selected solely based on the score it gets. In this model sentence scoring is done with two scores.

Score 1 (Statistical Score) This score is calculated by evaluating 6 statistical features of the sentence:

- Term Frequency-Inverse Sentence Frequency (TF_ISF) value: The TF_ISF value for a term 't' and sentence 's' is calculated using Eq. 1. It determines the importance of a sentence in the document by evaluating the frequency of each word in it.

$$\text{TF\_ISF}_{t,s} = \text{TF}_{t,s} * \log\left(\frac{\text{total sentences}}{\text{SF}_t}\right) \tag{1}$$

- Sentence position score: The position score of each sentence in a document is calculated based on Table 1. The position of a sentence 's' which occurs in nth sentence of the document is calculated by Eq. 2.

$$\text{Position}_{n,d} = \left(\frac{n}{\text{no. of sentences in } d}\right) * 100 \tag{2}$$

- Sentence numerical score: It is calculated using the count of digits in the sentence as defined in Eq. 3, to give weightage to sentences with numbers in it.

$$\text{Numerical score}_s = \text{count(Numbers in } s) \tag{3}$$

- Sentence noun score: It is calculated using the count of proper nouns in the sentence as explained in Eq. 4 to give weightage to sentences with names in it.

$$\text{Noun score}_s = \text{count(Nouns in } s) \tag{4}$$

- Sentence sentiment score: It is calculated using the sentiment polarity value in a sentence as explained in Eq. 5. It is found by using the in-built functions of TextBlob python library. By the addition of this attribute, sentences which portray a more positive or negative sentiment will obtain a greater score.

$$\text{Sentiment score}_s = \text{Sentiment polarity value of } s \tag{5}$$

- Sentence length score: It is calculated using the length of each sentence as explained in 6. By the addition of this attribute, sentences which are long will get a greater score.

$$\text{Length score}_s = \text{Length of } s \tag{6}$$

- Total statistical score: The total statistical score is calculated from Eqs. 1–6 and Table 1 as explained in Eq. 7.

$$\begin{aligned}\text{Statistical score}_s = {} & \text{TF-ISF score}_s + \text{Position score}_s \\ & + \text{Numerical score}_s + \text{Noun score}_s \\ & + \text{Sentiment score}_s + \text{Length score}_s\end{aligned} \tag{7}$$

Table 1 Position scores

Position (%)	Score
0–10	0.17
10–20	0.23
20–30	0.14
30–40	0.08
40–50	0.05
50–60	0.04
60–70	0.06
70–80	0.04
80–90	0.04
90–100	0.15

Score 2 (Topic Score) This score is used to further improve the efficiency of the summariser. The score is calculated for each sentence with the help of three elements.

- Topic modeller: A Topic modelling system which will return four of the most possible topics of the document is implemented utilising Latent Dirichlet Allocation technique [19].
- Noun Phrase extractor: The noun phrases in a sentence are identified by using the in-built functions of TextBlob Python library.
- Word Vectors: A Word2Vec model is implemented to find the similarity between words. It is a pre-trained word embedding model which contain millions of words represented as vectors. The word vectors are loaded and executed using Gensim [20] python library.

The score is calculated by finding the vector distance between the topics and noun phrases in each sentence using the Word2Vec model. The total 'Topic score' of a sentence 's' in a document with topics 't' is found using Eq. 8.

$$\text{Topic score}_{s,t} = \sum \text{Similarity between noun phrases in '}s\text{' and topics '}t\text{'} \tag{8}$$

3.2 Total Score

The total score of each sentence is calculated in Eq. 9 by summing the scores from Eqs. 7 and 8. The number of scores calculated will be equal to the number of sentences in the document.

$$\text{Total score}_{\text{sentence}} = \text{Statistical score}_{\text{sentence}} + \text{Topic score}_{\text{sentence}} \tag{9}$$

3.3 Algorithm

The pseudo-code for the proposed model is depicted below (Algorithm 1).

```
Data: Text Document
Result: Summarised Text Document
Function SIMILARITY($word_1$,$word_2$) is
    $word\_vector_1 \leftarrow vectorise(word_1)$
    $word\_vector_2 \leftarrow vectorise(word_2)$
    $distance \leftarrow Vector\ distance\ between$
     $word\_vector_1\ and\ word\_vector_2$
    $distance \leftarrow Normalise(distance)\ similarity = 1 - distance$
    return $similarity$
$pre\_processed\_document \leftarrow Remove\ unwanted$
 $whitespaces, symbols\ and\ stopwords\ topics \leftarrow Identify\ topics\ using$
 $Topic\ modelling$
foreach $sentence \in pre\_processed\_document$ do
    $noun\_phrases \leftarrow Noun\ phrases\ in\ the\ sentence$
    foreach $phrase \in noun\_phrases$ do
        $score \leftarrow$ SIMILARITY$(phrase, topics)$
    end
    $topic\ score = \sum score$
end
foreach $sentence \in pre - processeddocument$ do
    $score1 \leftarrow TF\_ISF\ score$
    $score2 \leftarrow Position\ score$
    $score3 \leftarrow Noun\ score$
    $score4 \leftarrow Numerical\ score$
    $score5 \leftarrow Length\ score$
    $score6 \leftarrow Sentiment\ score$
    $statistical\_score = score1 + score2 + score3 + score4 + score5 + score6$
end
foreach $sentence \in pre\_processed\_document$ do
    $total\_score = topic\_score + statistical\_score$
end
$sentence\_scores = sort\ 'total\_score'\ in\ descending\ order$
foreach $score \in sentence\_scores$ do
    Print the sentence corresponding to the score
end
```

Algorithm 1: Proposed algorithm

4 Experimental Results and Evaluation

The summarisation models are implemented in Python programming language (version 3.6) and the experiments are carried out in Google Colab [21] with 12 GB of RAM. The performance of our model is compared and analyzed with four different models, which is described in the following subsection.

4.1 Validating Models

The performance of the proposed model is compared to that of four dominant summarisation models.

Statistical Model Similar to the models in [4, 9, 10] a statistical summarisation model is built with additional parameters like position score, noun score, sentiment score, numerical score and length score. It is the same method used in the proposed model for the calculation of 'statistical score' (7). The total score is calculated according to Eq. 7.

Fuzzy Logic Model A fuzzy logic model is implemented to calculate the sentence scores from 4 statistical parameters, namely position score, numerical score, noun score and sentiment score. The model is formulated based on the Takagi-Sugeno-Kang (TSK) fuzzy rule [22] which is defined by Eq. 10.

$$\text{If } x \text{ is } A \text{ and } y \text{ is } B, \text{ then } z = f(x, y) \tag{10}$$

The model is implemented using the 'scikit-fuzzy' python library [23] and the fuzzy scores are calculated based on the 81 fuzzy rules defined using the combination of 4 parameters having 3 values each:

$$\begin{aligned}&\text{Position score, Numerical score, Noun score, Sentiment score}\\&\quad \epsilon \ \{\text{high, medium, low}\}\end{aligned}$$

Thus the membership function is defined by:

$$\text{Sentence score} = f(\text{Position score, Numerical score, Noun score, Sentiment score}) \tag{11}$$

The sentence score of each sentence is then used as the argument to select the most valuable sentences from the text.

Graph Model A graph model is a text evaluation technique where each sentence is represented as nodes in a graph. In [24], the authors built an Extractive text summariser which used a weighted undirected graph for scoring. Similarly, a graph-based summariser is built which considers the whole document as a graph and each sentence in the document as the nodes of the graph. The similarity between sentences is calculated using cosine-similarity and the pair of sentences which got a similarity score greater than some fixed threshold values are considered to have a connection between each other. Finally, the sentences with a greater count of connections are selected as the summary.

Deep Learning Model A deep learning model uses Recurrent Neural Networks (RNN) for summary calculation, it requires high computing power and time for summary generation. Due to the restrictions in computational resources, we used Resoomer [25], a pre-trained third-party summariser application to get the abstraction-based summaries.

The abstraction based summaries generated by deep learning models differ from those generated by extraction based models in many ways. A deep learning model generates shorter texts from the original ones using the semantics of the original text, whereas Extractive models form the summary by selecting relevant sentences from it. Upon comparing the two models using ROUGE-1 and F1-scores, we obtain quantitative insights about the preservation of native words used in the original text. The result verifies the effectiveness of the extraction based models in maintaining the native essence of the text when compared to abstraction based models.

4.2 Evaluation Metrics

The methods are evaluated based on ROUGE-1 score and F1 score. These are the most commonly used text evaluation techniques. ROUGE-1 score calculates 'Precision' and 'Recall' using Eqs. 12 and 13 respectively.

$$\text{Precision} = \frac{\text{no of overlapping words}}{\text{total no of words in reference summary}} \tag{12}$$

$$\text{Recall} = \frac{\text{no of overlapping words}}{\text{total no of words in program generated summary}} \tag{13}$$

F1 score is calculated using Eq. 14.

$$\text{F1 score} = 2 * \left(\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}\right) \tag{14}$$

4.3 Reference Summary: Sumy Models

The reference summary for the comparison is generated using the Luhn summariser function provided by the Sumy library in Python. It generates Extractive heuristic summaries [26] based on the frequency of words in the document.

4.4 Dataset Used

The models are evaluated using three datasets: two 'NEWS datasets' [27, 28], containing thousands of NEWS articles and a 'COVID19 dataset' [29] containing data about the non-medical aspects of COVID19. To evaluate the performance of the model in different input scenarios, the three datasets are combined and then split into two, based on their size: one containing articles with less than 20 sentences and other with articles containing more than 50 sentences. For the sake of simplicity in execution, only 5000 articles from each dataset are taken for the evaluation.

4.5 Results

The ROUGE-1 and F1 score obtained for each of the models are depicted in Tables 2, 3 and 4. A graphical representation of ROUGE-1 scores are illustrated in Fig. 2.

Table 2 F1-scores of each model

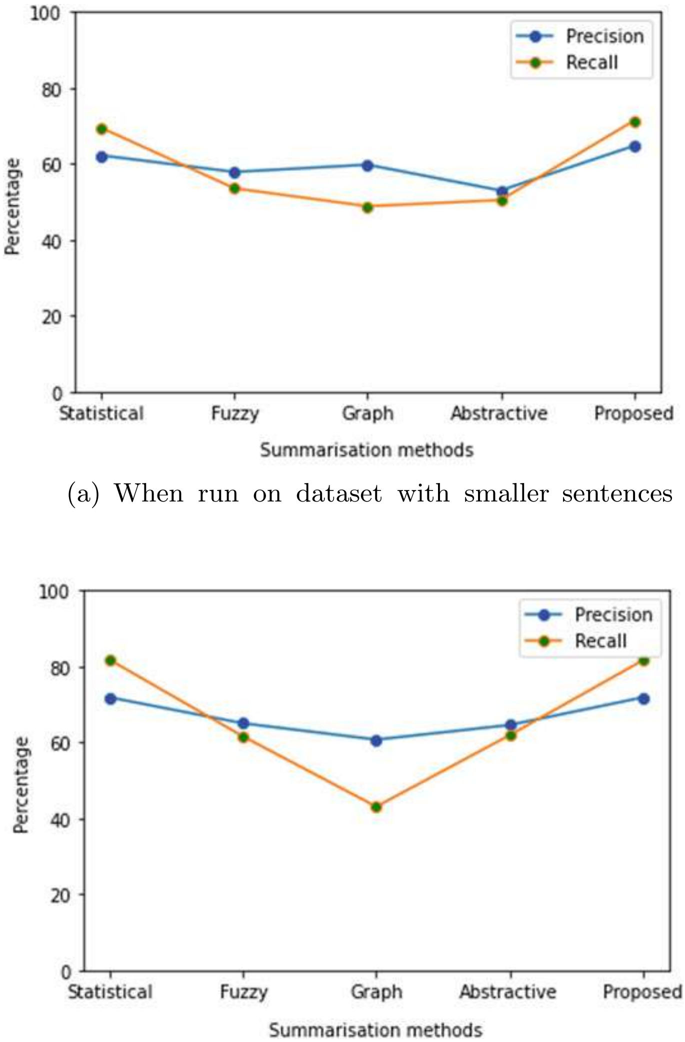
Method	Smaller data	Larger data
Statistical model	65.6342	76.4208
Fuzzy logic model	55.6121	63.2244
Graph model	53.7377	50.3903
Abstractive model	51.6848	63.1772
Proposed model	**67.8453**	**76.3701**

Table 3 ROUGE-1 values when run on dataset with smaller sentences

Method	Precision	Recall
Statistical method	62.2124	69.4545
uzzy logic method	57.8546	53.5371
Graph method	59.7585	48.8191
Abstractive method	52.9758	50.4554
Proposed method	**64.6983**	**71.3142**

Table 4 ROUGE-1 values when run on dataset with larger sentences

Method	Precision	Recall
Statistical method	71.8145	81.6586
Fuzzy logic method	65.0081	61.5360
Graph method	60.7167	43.0659
Abstractive method	64.5574	61.8549
Proposed method	**71.8157**	**81.5415**



(a) When run on dataset with smaller sentences

(b) When run on dataset with larger sentences

Fig. 2 ROUGE-1 values

5 Conclusion

In this paper, we presented a new method for Extractive text summarisation, to improve its efficiency in terms of summarisation accuracy. The proposed method computes the sentence score in two steps, namely by evaluating its statistical features and by checking the similarity each had with topics of the document. The performance of our model was compared with that of a few other popular models in Extractive text summarisation using the metrics ROUGE-1 and F1 score. The experimental results show that the proposed model outperforms other models when run on both long and short text data. Thus, the proposed model was able to achieve superior performance than the existing Extractive summarisers without compromising on implementational simplicity. As future work, we plan to implement the proposed model as a web application.

References

1. El-Kassas, W.S., Salama, C.R., Rafea, A.A., Mohamed, H.K.: Automatic text summarization: a comprehensive survey. Expert Syst. Appl. 113679 (2020)
2. Deepa Raj K., Sajeev, G.P.: A community based web summarization in near linear time. In: 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 962–968 (2018)
3. Naveen, G.K.R.: Query-based multi-document summarization by clustering of documents (2014)
4. Christian, H., Agus, M.P., Suhartono, D.: Single document automatic text summarization using TF-IDF. ComTech: Comput. Math. Eng. Appl. **7**(4), 285–294 (2016)
5. Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B., et al.: Abstractive text summarization using sequence-to-sequence RNNs and beyond. arXiv preprint arXiv:1602.06023 (2016)
6. Otter, D.W., Medina, J.R., Kalita, J.K.: A survey of the usages of deep learning for natural language processing. IEEE Trans. Neural Netw. Learn. Syst. (2020)
7. Bommasani, R., Davis, K., Cardie, C.: Distilling static embeddings from pretrained contextual representations, Bert wears gloves (2019)
8. Jacobi, C., Van Atteveldt, W., Welbers, K.: Quantitative analysis of large amounts of journalistic texts using topic modelling. Digit. J. **4**(1), 89–106 (2016)
9. Gunawan, D., Pasaribu, A., Rahmat, R.F., Budiarto, R.: Automatic text summarization for indonesian language using TextTeaser. Proc. Electr. Eng. Comput. Sci. Inform. **3**(1), 012048 (2016)
10. Patel, S.M., Dabhi, V.K., Prajapati, H.B.: Extractive based automatic text summarization. JCP **12**(6), 550–563 (2017)
11. Patel, V., Tabrizi, N.: An automatic text summarization: a systematic. Int. J. Inf. Eng. Electron. Bus. (2019)
12. Mandal, S., Singh, G.K., Pal, A.: A hybrid text summarization approach. J. Inform. Math. Sci. **9**(3), 547–555 (2017)
13. Khan, R., Qian, Y., Naeem, S.: Extractive based text summarization using k-means and TF-IDF. Int. J. Inf. Eng. Electron. Bus. **11**(3), 33 (2019)
14. El-Refaiy, A., Abas, A.R., Elhenawy, I.: Review of recent techniques for extractive text summarization. J. Theor. Appl. Inf. Technol. **96**, 7739–7759 (2018)
15. Kumar, K.M.S., Soumya, R.: Text summarization using clustering technique. Int. J. Appl. Eng. Res. **10**, 25511–25519 (2015)

16. Bewoor, M.S., Patil, S.H.: Empirical analysis of single and multi document summarization using clustering algorithms. Eng. Technol. Appl. Sci. Res. **8**(1), 2562–2567 (2018)
17. Mallick, C., Das, A.K., Dutta, M., Das, A.K., Sarkar, A.: Graph-based text summarization using modified TextRank. In: Soft Computing in Data Analytics, pp. 137–146. Springer (2019)
18. Dzone.com. Word vectors. https://bit.ly/3deytoQ (2018)
19. Wikipedia. Latent Dirichlet allocation. https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation
20. Gensim. Documentation. pypi.org/project/gensim (2020)
21. Google. Colaboratory. colab.research.google.com
22. Research Hubs. TS fuzzy model. http://researchhubs.com/post/engineering/fuzzy-system/takagi-sugeno-fuzzy-model.html
23. pypi.org. scikit-fuzzy. https://pypi.org/project/scikit-fuzzy/
24. Siji Rani, S., Sreejith, K., Sanker, A.: A hybrid approach for automatic document summarization. In: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 663–669 (2017)
25. Resoomer. Text summariser. resoomer.com/en
26. Leidner, J.L.: Summarization in the financial and regulatory domain. In: Trends and Applications of Text Summarization Techniques, pp. 187–215. IGI Global (2020)
27. Kaggle. Dataset1: All news dataset. kaggle.com/snapcrack/all-the-news
28. Kaggle. Dataset2: News summary. kaggle.com/sunnysai12345/news-summary
29. Kaggle. Dataset3: Covid19 dataset. kaggle.com/jannalipenkova/covid19-public-media-dataset (2020)

Part of Speech Tagging Using Bi-LSTM-CRF and Performance Evaluation Based on Tagging Accuracy

Shilpa Kamath, Chaitra Shivanagoudar, and K. G. Karibasappa

Abstract Part of speech tagging (POS) refers to the computational task of identifying related parts of speech for specific words in text documents. A research challenge is to use various techniques to identify and utilize these tags to improve several natural language processing applications. In this paper, a bidirectional long short-term with conditional random field (denoted as Bi-LSTM-CRF) model has been proposed for POS tagging. This novel model trained on the named entity resolution dataset is compared with other recurrent neural network models such as bidirectional long short-term model networks and long short-term model with conditional random field. Bi-LSTM-CRF model is applied to the annotated NER dataset and its tagging accuracy and F1-score are compared with the other pre-existing models. Experimental results show that Bi-LSTM-CRF provides better results for POS tagging. The Bi-LSTM-CRF model is competitive on the annotated NER dataset for English to produce greater accuracy and F1-score and outperform the rest of the models.

Keywords LSTM · CRF · Bidirectional LSTM · Natural language processing

1 Introduction

Neural network-based models are powerful models that perform significantly better on large datasets compared to other machine learning models. The researchers have been working on sequence tagging and this hot topic has drawn their attention for a few decades. Sequence tagging mainly consists of classic natural language processing (NLP) tasks such as named entity recognition (NER), part of speech tagging (POS),

S. Kamath (✉) · C. Shivanagoudar · K. G. Karibasappa
KLE Technological University, Hubballi, Karnataka, India
e-mail: shilpa.kamath@kletech.ac.in

C. Shivanagoudar
e-mail: chaitragoudar9119@gmail.com

K. G. Karibasappa
e-mail: karibasappa_kg@kletech.ac.in

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_25

and chunking. Named entity recognition is a process of classifying named entities related to a text. Chunking is a process of adding more structure to a sentence by identifying short phrases and POS tags in that particular sentence. It also helps to group phrases which are syntactically correlated. Recurrent Convolution Neural Network has the ability to capture the contextual information for these short phrases[1]. POS tagging refers to assigning a specific part of speech tags to the words in a sentence. These labeled texts find a wide range of applications in natural language understanding, building chatbots, text classification [2], speech recognition, sentiment analysis, automatic summarization, and market intelligence analysis. Named entity recognition has applications in clustering and classifying categories in a sentence such as a person's name and organization name. In this paper, the major focus is on POS tagging, with the help of which a word can be reduced to its root form. POS tags give a brief idea of how certain words are used in a sentence. With the help of these tags, the context of the sentence can be clearly understood.There have been many applications where POS taggers are used for tokenisation but, it would be better if the individual POS tagging is applied as a part of preprocessing for better results [3].

In this paper, a Bi-LSTM-CRF model has been proposed and examined for the POS tagging task. The pre-existing models such as bidirectional LSTM (Bi-LSTM) networks and LSTM with conditional random field (LSTM-CRF) have been used for comparison. These pre-existing models are initially trained using an annotated NER dataset along with the proposed model. Later, the comparison of models is done with the proposed model based on the obtained tagging accuracy and F1-score. The rest of the paper is organized as follows; Sect. 2 describes the background and related work. Section 3 describes the different LSTM's mentioned in the paper. Section 4 describes the dataset used for training the models. Section 5 describes the procedure used for training the model. Section 6 reports the experimental results and Sect. 7 concludes the paper.

2 Background and Related Work

Zhiheng Huang et al. [4] proposed RNN-based models, which had a special feature of using past and present features of input to estimate future results. These are long short-term memory (LSTM)-based models that were used for sequence tagging. For the sequence tagging, datasets such as computational natural language learning (CoNLL), NER, and POS a bidirectional LSTM with a CRF layer was applied. The experimental study proved that the Bi-LSTM-CRF model could produce close to accurate results for NLP datasets.

Xuezhe Ma [5] introduced a hybrid of bidirectional convolutional neural network (CNN) and CRF. This neural network benefited from both word and character level representation. This model was tested on two sequence labeling tasks. The first dataset Penn Treebank WSJ corpus for POS tagging obtained an accuracy of 97.55% and the other dataset CoNLL 2003 corpus for NER tasks obtained an F1-score of 91.21%.

Guillaume Lample et al. [6] introduced two novel neural architectures. The approach used was inspired by the shift-reduce parsers. The first network was a bidirectional LSTM network and the other network constructed and labeled segments using a transition-based approach. These models obtained good accuracy for NER, for four languages. The performance of these models was independent of language-specific resources and knowledge.

Yue Zhang and Liu [7] investigated an alternative LSTM structure for encoding text, which consisted of a parallel state for each word. The results on various sequence labeling benchmarks and classification depicted that the proposed model was capable of exhibiting highly competitive performances compared to stacked Bi-LSTM models with similar parameter numbers.

Derry Wijaya [8] presented a unique method for clustering. These clusters were the outputs of VerbKB: verbs with their multiple argument types. He used novel low-dimensional verb embeddings and their arguments to produce high-quality clusters. Here, the verb could be in different clusters and its position in clusters depending on the argument type. The experimental studies showed that the verb clusters formed did a better job than the hand-built clusters of predicting sentiment and sarcasm in tweets.

3 Models

In this section, pre-existing models such as Bi-LSTM, LSTM-CRF, and the proposed model Bi-LSTM-CRF model have been described.

3.1 LSTM Networks

The structure of the human brain inspires a neural network. Human thoughts have persistence. Neural networks especially the traditional ones fail to behave as humans do in this aspect and this becomes the major disadvantage. Recurrent neural networks emerge as a solution to these shortcomings of traditional networks [9]. These networks consist of loops in them, which seems to allow information to persist. Almost all the best results on the recurrent neural networks can be achieved with LSTM's [4] which is a special kind of recurrent neural network. LSTM's avoid long term dependency problem. LSTM networks have purpose-built-memory-cells in them which helps to outperform RNN [10]. This attribute helps LSTM's to identify long-range dependencies in the data. The RNN structure consists of an input layer *x*, hidden layer *h*, and output layer *y*. In the POS tagging context, input features are represented by *x*, and in output features, tags are denoted as *y*. In a POS tagging system, each word is tagged with respective POS tags. The sentence Corona is a deadly virus that can be tagged as proper noun (NP), verb be (VBZ), preposition (IN), and adjective (JJ).

3.2 Bidirectional LSTM Networks

When the two independent RNNs are put in parallel it results in a bidirectional recurrent neural network. The inputs will be run in two ways, one direction is from past to future and the other direction is from future to past. In a unidirectional way, the inputs are run only in a single direction, and information is preserved, but the two-way run approach in Bi-LSTM can preserve both the past and future information [11]. This quality of Bi-LSTM makes them perform better than regular LSTM.

3.3 LSTM-CRF Networks

An LSTM-CRF model is a combination of an LSTM network and a CRF network. Conditional random field (CRF) is a class of statistical modeling techniques applied in machine learning and neural network-based models to make structured predictions. The CRF network accompanied by LSTM can use both past and future tags to predict the current tag. This approach is similar to the use of past and future features in the bidirectional LSTM network the only up-gradation is the CRF layer which aims at giving sequential output.

3.4 Bi-LSTM-CRF Networks

A combination of the bidirectional long short-term memory with a conditional random field (Bi-LSTM-CRF) is one of the high performing recurrent neural networks [4]. The tokens of a sentence are processed in an LSTM which forms an intermediate representation. This representation acts as an input for the CRF. This provides the prediction of all the token labels. The LSTM successfully captures sequential relationships among tokens, and CRF aims at sequentially joining predictions of every label or tag in the sentence. This model can also use future input features along with sentence-level tagging information and input features from the past. For example, the input sentence Corona is a deadly virus; along with its respective POS tags: proper noun (NP), verb be (VBZ), preposition (IN), adjective (JJ) are processed through a Bi-LSTM network. The Bi-LSTM network learns each word of the sentence and the POS tag associated with it in non-reverse and reverses the order. This gives additional context to the network and the sentence is learned quickly and accurately. The CRF layer helps to join the predictions sequentially and output the sentence as Corona (NP) is (VBZ) an (IN) deadly (JJ) virus (NP). Our work is close to the work of (Zhiheng Huang et al.) as they employed Bi-LSTM-CRF for sequence tagging. The model proposed in this paper is trained on an annotated NER dataset for POS tagging. With the one-hot-encoding used for embedding and other parameter settings, the model yields better tagging accuracy and F1-score.

4 Dataset Description

The models LSTM-CRF, Bi-LSTM, and Bi-LSTM-CRF have been tested on annotated corpus for named entity recognition using Groningen Meaning Bank (GMB) corpus for POS tagging obtained from Kaggle. The dataset consists of 47,959 sentences containing 35,178 different words, 17 NER tags, and 42 different POS tags. The models are evaluated on three test cases. In the first test case, 38,846 sentences are taken for training, 4796 sentences are considered for testing, and 4317 sentences are considered used for validation. For the second test case, 34,503 sentences are taken for training, 9619 sentences are considered for testing, and 3837 sentences are considered for validation. For the third test case, 30,213 sentences are taken for training, 14,388 sentences are considered for testing, and 3358 sentences are considered for validation, and for the fourth test case, 26,230 sentences are used for training, 18,807 for training, and 2922 sentences for validation. Preprocessing has not been carried out for this dataset as it does not have any effect on the results. POS assigns each tag to the respective word in a sentence. For example, a word starting with a noun phrase can be tagged as B-NN.

5 Training Procedure

In this paper, the model proposed is inspired by [4] where the combination of Bi-LSTM-CRF is applied for language-independent NER datasets. Given a sentence, the proposed model predicts a label corresponding to each of the input tokens in the sentence. The labels are the POS tags for those particular words of the sentences. Firstly, through the embedding layer, the sentence is represented as a sequence of vectors $X = (X_1, \ldots, X_n)$ where the length of the sentence is denoted by n. One-hot encoding has been used for embedding where each word is represented by a vector. Next, the embeddings are given as an input to a Bi-LSTM layer. The LSTM-CRF network is fit with an embedding layer. In each epoch, the whole training data is divided into batches and one batch are processed at a time. The batch size is another hyperparameter which is chosen to be 32, which means the sentences whose total length not greater than 32 are included for training the model. This batch size helps to yield better accuracy for our Bi-LSTM-CRF model. Parameter optimization is performed with a learning rate of 0.1 and the dropout value is set to be 0.1 to attain better accuracy. The parameter setting is depicted in Table 1. The boundary indicates the LSTM layer and the tag indicate the CRF layer. The layers in the model such as

Table 1 Parameter setting

Word embedding	Drop out	Batch size	Optimizer
One-hot encoding	0.1	32	Rmsprop

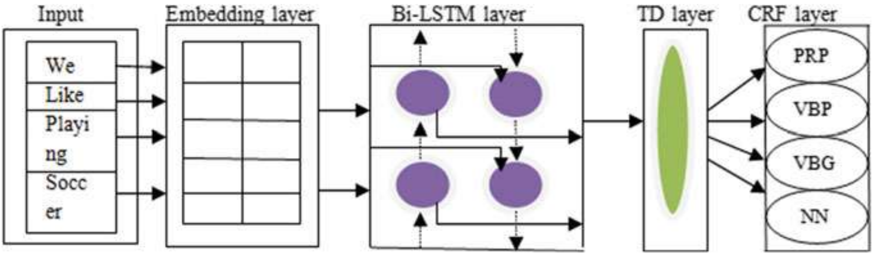


Fig. 1 Various layers of the proposed BI-LSTM-CRF model

the embedding layer, Bi-LSTM network, time distributed layer (TD), and CRF layer are depicted in Fig. 1.

The model operates as follows. A bidirectional LSTM-CRF model forward pass is run for each batch where the forward pass is for both the forward state and the backward state of LSTM. An output score at all positions for all the tags has been obtained. The sequence from left to right for each word t is computed by a representation Ht which is the hidden layer, and backward computation is represented as Ht for the same sequence in a reverse direction. Different parameters are used for these two distinct networks and the word representation is obtained by concatenating its left and right context representations. These two distinct networks use different parameters, and then the representation of a word is obtained by concatenating its left and right context representations. To compute gradients for network output and state transition edges, a CRF layer is run. Backpropagation takes place and the errors travel back from the output to input. Later, the network parameters are updated which include the state transition matrix P. A time distributed layer T is used to process the sequential output one by one. The architecture for the proposed Bi-LSTM-CRF is depicted in Fig. 2. The boundary indicates the LSTM layer and the tag indicate the CRF layer.

6 Experimental Results

The dataset obtained is split into a test, train, and validation. In the prescribed annotated NER dataset for the first test case, 38,846 sentences are considered for training, 4796 sentences are considered for testing, and 4317 sentences are considered for validation. For the second test case, 34,503 sentences are considered for training, 9619 sentences are considered for testing, and 3837 for validation. For the third test case, 30,213 sentences are considered for training, 14,388 sentences for testing, and 3358 sentences are taken for validation, and for the fourth test case, 26,230 sentences are used for training, 18,807 for training, and 2922 sentences for validation. Model accuracy and F1-score, which conveys the balance between precision and recall are used to evaluate the performance of the model.

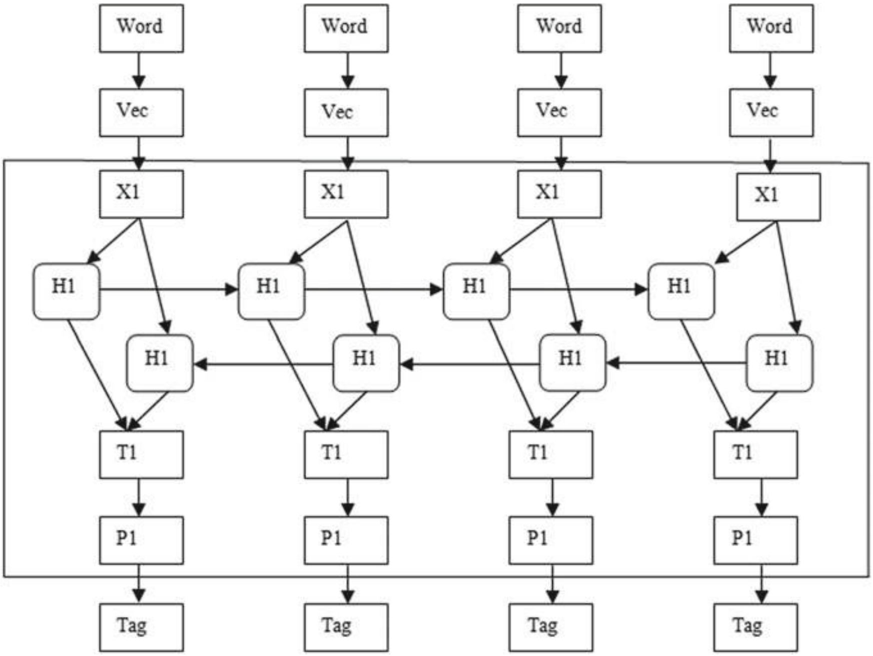


Fig. 2 Architectural diagram of the proposed model

The LSTM-CRF, Bi-LSTM, and Bi-LSTM-CRF models are trained on the annotated NER dataset. The models are trained on training data and performance is monitored on validation data. A learning rate of 0.1 is used to train the models. The hidden layer size will not affect the performance of the model. The training requires 5 epochs and it takes less than a few hours to converge. The model performance based on accuracy and F1-score for POS tagging on four varied training, testing, and validation set is given in Table 2. The model performance based on accuracy and F1-score for an increasing number of epochs is depicted in Table 3. The graphical representation of the tagging accuracy where the *x*-axis represents the number of test cases with test_size value and the *y*-axis represents tagging accuracy for the prescribed models is depicted in Fig. 3a. The graphical representation of F1-score where the *x*-axis represents the number of test cases with test_size value and the *y*-axis represents F1-score for the prescribed models is depicted in Fig. 3b. For the first test case, the test_size is set to be 0.1, for the second test case, the test_size value is taken as 0.2, 0.3 for the third test case, and 0.4 for the fourth test case. The experimental results show that for increasing test_size the performance accuracy and F1-score of all the three models get decreased. The experiment is stopped at test_size 0.4 as further split may lead to overfitting of the model. Hence, further processing is not carried out.

The best accuracy and F1-score for all the models are obtained at test_size 0.1. By comparison of performance accuracy and F1-score of all the models, it is evident that for the POS tagging task Bi-LSTM-CRF model results in the highest tagging

Table 2 Comparison of tagging accuracy and F1-score on different models for annotated NER dataset

Models	Test case	Training set	Test set	Validation set	Accuracy (%)	F1-score (%)
Bi-LSTM	Test Case 1	38,846	4796	4317	98.61	92.5
	Test Case 2	34,503	9619	3837	97.25	91.16
	Test Case 3	30,213	14,388	3358	96.15	89
	Test Case 4	26,230	18,807	2922	95.03	87.65
LSTM-CRF	Test Case 1	38,846	4796	4317	96	92.13
	Test Case 2	34,503	9619	3837	95.31	90.13
	Test Case 3	30,213	14,388	3358	95.01	88.2
	Test Case 4	26,230	18,807	2922	94.75	87.3
Bi-LSTM-CRF	Test Case 1	38,846	4796	4317	99.44	97.1
	Test Case 2	34,503	9619	3837	99.32	96.96
	Test Case 3	30,213	14,388	3358	99.17	96.43
	Test Case 4	38,846	4796	4317	98. 92	96.6

accuracy and F1-score which is significantly better than the performance metrics obtained by other models as shown in Fig. 3a, b. The graphical representation of tagging accuracy and F1-score versus the number of epochs depicted in Fig. 4a, b proves that increasing epochs improves the accuracy and F1-score of the model. The epochs ranged from 2 to 10. The accuracy and F1-score of Bi-LSTM-CRF attained maximum value at epoch 8 and further got decreased at epoch 10. Hence, the number of epochs is restricted to 10.

7 Conclusion

In this paper, the three LSTM models are trained on the annotated NER dataset for POS tagging. The test_size of the dataset was varied, and the models were trained. The models attained the highest accuracy and highest F1-score for test case 1 and the

Table 3 Comparison of tagging accuracy and F1-score on increasing epochs for different models

Models	Epochs	Accuracy	F1-score
Bi-LSTM	2	98.19	94.6
	4	98.47	95.8
	6	98.46	96
	8	98.51	96.65
	10	98.52	96.4
LSTM-CRF	2	96	93
	4	95.31	93.4
	6	95.01	93.54
	8	94.75	93.78
	10	97.51	93.83
Bi-LSTM-CRF	2	99.22	95.8
	4	99.43	96.8
	6	99.48	97
	8	99.51	97.3
	10	99.48	97.3

downfall of accuracy and F1-score was noticed for all the other test cases, due to the overfitting of the model. Hence, test case 1 seemed to provide good performance for the models. Compared to the other models of LSTM, Bi-LSTM-CRF model produced a tagging accuracy of 99.44% and F1-score of 97.1% for test case 1. Later, the model accuracy and F1-score of Bi-LSTM-CRF were noted by increasing the number of epochs ranging from 2 to 10. For epoch 8, the model produced the highest tagging accuracy of 99.51% and F1-score of 97.3%. This enhancement in tagging accuracy and F1-score is due to its unique three-layered architecture which is successful in predicting related POS tags for the sentences. These models can also be used to build a question answering system and find major applications in various NLP tasks. The future scope of the project would be to use different neural networks along with Bi-LSTM-CRF for various sequence tagging methods.

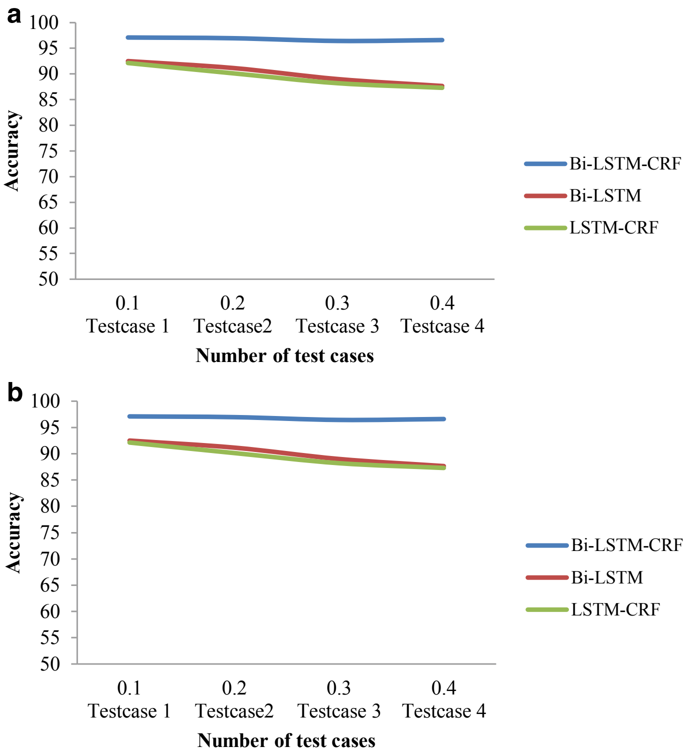


Fig. 3 **a** Accuracy obtained for the models as test-data size increases. **b** F1-score obtained for the models as test-data size increases

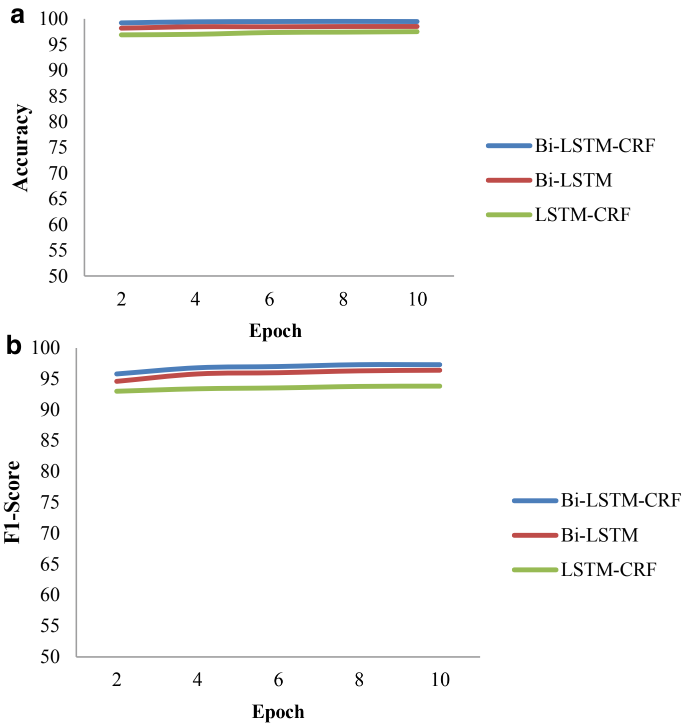


Fig. 4 **a** Accuracy obtained for the models with an increase in the number of epochs. **b** F1-score obtained for the models with an increase in the number of epochs

References

1. Lai, S., Xu, L., Zhao, J.: Recurrent convolutional neural networks for text classification. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence, Austin, USA, pp. 25–30 (2015)
2. Kim, Y.: Convolution neural networks for sentence classification. In Proceedings with the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, pp. 1746–1751 (2014)
3. Dos Santos, C., Gatti, M.: Deep convolutional neural network for sentimental analysis of short texts. In: Proceedings of the COLING 2014, the 25th International Conference on Computational Linguists, Dublin, Ireland, pp. 69–78 (2014)
4. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF for sequence Tagging. In: Proceedings of Advances in Deep Learning Research, Beijing, China, pp. 177-187 (2016)
5. Ma, X., Hovy, E.: End to end sequence labeling via Bi-directional LSTM-CNN's-CRF. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, pp. 1064–1074 (2016)
6. Lample, G., Ballesteros, M., Subramanian, S.: Neural architectures for named entity recognition. In: Proceedings of NAACL San Diego, California, USA (2016)
7. Zhang, Y., Liu, Q.: Sentence state LSTM for text representation. In: Proceeding of Advances in Neural Information Processing Systems, pp. 349–357 (2018)

8. Wijaya, D.T.: VerbKB: a knowledge base of verbs for natural language understanding. Ph.D. Thesis, Carnegie Mellon University. In: Proceedings of NAACL-HLT, pp. 818–827 (2016)
9. Jozefowicz, R., Zaremba, W., Sutskever, I.: An empirical exploration of recurrent network architectures. In: Proceedings of the 32nd International Conference on Machine Learning, pp. 2342–2350 (2015)
10. Tai, K.S., Socher, R., Manning, C.: Improved semantic representations from tree-structured long short-term memory networks. In: Proceedings of ACL, Beijing, China, pp. 1556–1566 (2015)
11. Schmidhuber, J.: Deep learning in neural networks: an overview. **61**, 85–117 (2015)

Clustering Research Papers: A Qualitative Study of Concatenated Power Means Sentence Embeddings over Centroid Sentence Embeddings

Devashish Gaikwad, Venkatesh Yelnoorkar, Atharva Jadhav, and Yashodhara Haribhakta

Abstract Mathematical average of word embeddings is a common baseline for sentence embedding techniques which typically fall short of the performance of more complex models such as BERT and InferSent. There has been significant improvement in the field of sentence embeddings and especially towards the development of universal sentence encoder that can be used for transfer learning in a wide variety of downstream tasks. Academic paper retrieval systems are widely used in academic institutions to store and categorise scientific papers and find connections between them using citation links, but these methods do not account for the content of the papers. For unsupervised clustering of these papers, a new approach of sentence embeddings is proposed using concatenated power means sentence embeddings and centroid sentence embeddings. The sentence embeddings so created are clustered using K-means clustering algorithm. The results show a clear increase of 47.94% in cosine distance of nearest papers using concatenated power means sentence embeddings with respect to baseline centroid embeddings for the highest performing GloVe models proving that the computationally inexpensive P-Means clustering sentence embeddings can be used for unsupervised clustering of scientific research papers using their abstracts.

Keywords Sentence embedding · Word embedding · Natural language processing · Clustering · Text classification

1 Introduction

Categorising text has been a topic of interest for a long time in the field of natural language processing, as it represents the most natural structural hierarchy for storing documents. While features such as keywords, tags and citation links help retrieve similar research papers from repositories, they do not account for the actual content

D. Gaikwad (✉) · V. Yelnoorkar · A. Jadhav · Y. Haribhakta
College of Engineering, Pune, Wellesley Road, Shivajinagar, Pune, Maharashtra 411005, India
e-mail: gaikwadds16.it@coep.ac.in

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_26

of the paper. Thus, missing out on semantic similarity and structure of related papers. Instead of using human generated tags and citations, sentence embeddings can aid the paper retrieval and storage systems by clustering similar papers semantically. These clusters created using sentence embeddings thus represent papers related to the true meaning of underlying text.

While sentence embeddings are able to express the meaning of the sentences, they fall short of representing the meaning of long document length texts. Sentence embeddings work best on paragraphs of text. The vector representations of sentences are also very large. In context of research papers, the paragraph containing the most information about the paper is the abstract of the paper. Thus, the authors postulate that the sentence embeddings of an abstract can represent the whole paper, and thus be used to cluster papers. The authors have used concatenated power means sentence embeddings [14] which give comparable and steady performance with minor trade-offs as compared to baseline centroid measure with less computational overhead as compared to state-of-the-art models.

The authors propose and evaluate unsupervised clustering of academic papers for exploratory data analysis by using the concatenated power means [14] (P-Means) and a simple centroid of their abstracts as the clustering keys. The models use word embeddings (GloVe [12], Word2Vec [9] and FastText [3]) of words in the abstract and create a single sentence embedding from them, which will be later used to cluster them using K-means clustering algorithm. The embeddings clustered using K-means are evaluated using the metrics of the cosine distance, Silhouette score and Davies–Bouldin score.

2 Related Work

2.1 Citation-Based Clustering

Clustering research papers using their citations [17] is one of the earliest proposed methods for clustering papers and retrieving information from them. Derek de Solla Price [17] proposed creating networks of scientific papers using their citation links to create groups of papers which would allow finding out papers which are on the edge of a particular science, papers which are timeless classics and papers which will be lost to time. In the modern times, three main methods of clustering using citations exist—bibliographic coupling put forth by Kessler [7] which clusters the documents which are cited by the same set of papers into same cluster, co-citation analysis proposed by Irina Marshakova-Shaikevich [8] which puts the documents which cite the same set of papers into same cluster and direct citations refined by Shibata et al. [15] which clusters the papers which cite and are cited by in the same cluster. Out of these bibliographic coupling is the best [4].

2.2 Tag-Based Clustering

Sjogarde et al. [16] propose a algorithmically constructed publication-level classification of research publications, by using already constructed citation graphs, clustering them, applying tags to each cluster and then classifying documents by clustering model trained on this algorithmically generated data. Adayman et al. [1] propose clustering economics papers by their Journal of Economic Literature (JEL) tags using a nonparametric novel Adaptive Weight Clustering Algorithm by Efimov et al. [6]. All of these methods do not take the actual content of the papers into account for clustering and categorising them.

2.3 Natural Language Processing-Based Clustering

With the latest advancements in natural language processing, several ways of clustering documents and texts are being explored. Boyak et al. [5] applied TF-IDF, latent semantic analysis, topic modelling and Poisson-based language models on MEDLINE biomedical publication dataset. Nikfarjam et al. [11] clustered social media posts for pharmacovigilance using word embeddings and conditional random fields. These consider the semantic meaning of underlying words but not the whole sentences.

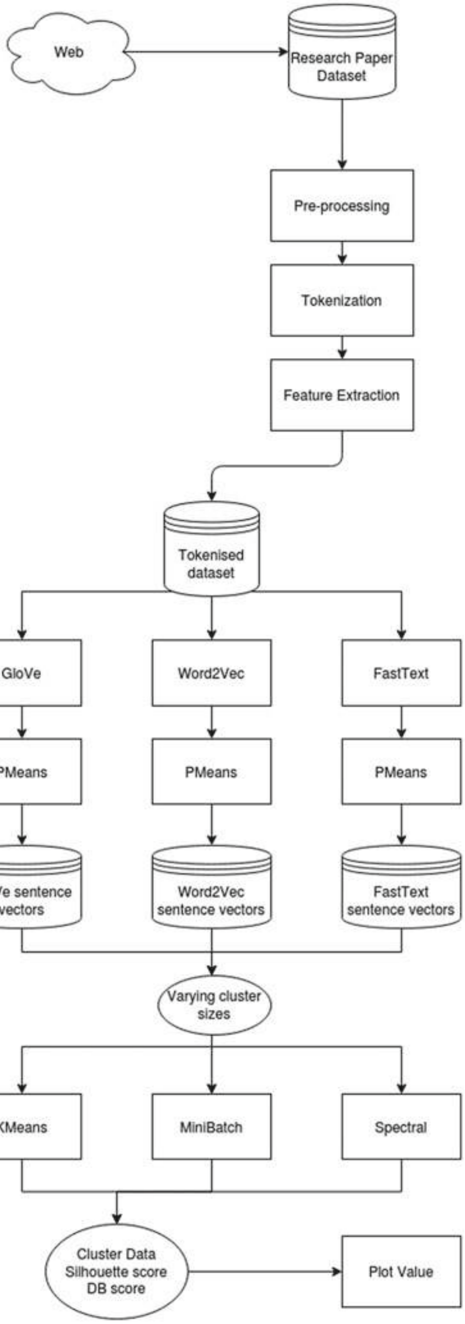
The authors propose to cluster scientific papers using the concatenated power means sentence embeddings and centroid sentence embeddings as representations of their abstracts for K-means unsupervised clustering algorithm and evaluate the performance of proposed methods.

3 Methodology

Authors cluster 30,000 and 15,000 research papers from CORE[1] dataset into 350 and 250 clusters, respectively, in accordance with Zipf's law [10], using K-means clustering algorithm on concatenated power means sentence embeddings and centroid sentence embeddings of their abstracts, made using GloVe, Word2Vec and FastText word embeddings and then compare the quality of clusters by calculating Silhouette scores and Davies–Bouldin scores using the machine learning pipeline given in Fig. 1.

[1] https://core.ac.uk/services/dataset/.

Fig. 1 Research methodology



3.1 Dataset

The dataset was collected from CORE[1], the service offers multiple datasets varying in the year of publication and number of papers. The metadata dataset published in 2013 containing over 275,000 records was chosen, as it suited the need of having just the abstract, as the body of the papers was not used for clustering.

3.2 Sentence Embeddings

The abstracts were lemmatised, stop words were removed and converted to arrays of word embeddings. For every word embedding type, GloVe 50d, Word2Vec 100d and FastText 300d word embeddings were used.

Concatenated power means sentence embeddings [14] and centroid sentence embeddings were created for every word embedding type. Power means method generalises the average word embeddings by retrieving many well-known means such as the arithmetic mean ($p = 1$), the geometric mean ($p = 0$) and the harmonic mean ($p = -1$). When ($p = \pm\infty$), the power mean specialises to the minimum ($p = -\infty$) and maximum ($p = +\infty$) of the sequence.

$$\epsilon = \left(\frac{x_1^p + x_2^p + x_3^p + \cdots + x_n^p}{n}\right)^{\frac{1}{p}} \quad p \in \mathbb{R} \cup \{\pm\infty\} \tag{1}$$

Here, x is word embedding and n is the number of words.

$p = 1, +\infty, -\infty, 2, 4$ were used for the model. Concatenated power means (P-Means) was chosen because of its established accuracy in multiple downstream tasks and low computational power requirement [13, 14].

Centroid method summarises a sequence of embeddings $x_1, x_2, \ldots, x_n \in \mathbb{R}$ by component-wise arithmetic averages. Centroid sentence embedding is just power means with $p = 1$.

$$\epsilon = \left(\frac{x_1 + x_2 + x_3 + \cdots + x_n}{n}\right) \tag{2}$$

P-Means was compared with the simplest sentence embedding algorithm, i.e. centroid. It was not compared with SIF-sentence embeddings [2] since the number of words repeated in paper abstracts is very less.

3.3 Optimal Number of Clusters

Selection of the number of clusters (K) was done using Zipf's law [10]. Silhouette score for range $K = 2, 6, 10, 14, \ldots, 3000$ (skipping every 4 digits) with 30,000 papers for K-means using P-Means with GloVe was calculated. It was found out that for every multiple of the square root of 30,000 (i.e. $173.2 \approx 173$), the Silhouette score did not increase substantially after 346 (i.e. twice the square root). The Silhouette score was 0.1–0.8 for $K = 2$–18. Clustering models were trained in the following combinations—Algorithm: K-means. Word embedding: GloVe 50d, Word2Vec 100d, FastText 300d. Sentence Embeddings: P-Means, Centroid. Cluster and number of papers pairs: (50, 3000), (250, 15,000), (350, 30,000). 250 and 350 are approx twice the square root of 15,000 and 30,000, respectively.

3.4 Clustering

Over 18 clustering models were tested for each set of parameters and number of papers, models took from 1 to 6 h to complete clustering. Google Colab Cloud Platform with NVIDIA P100 GPUs was used to train the models. The Silhouette score and Davies–Bouldin score of every model were computed.

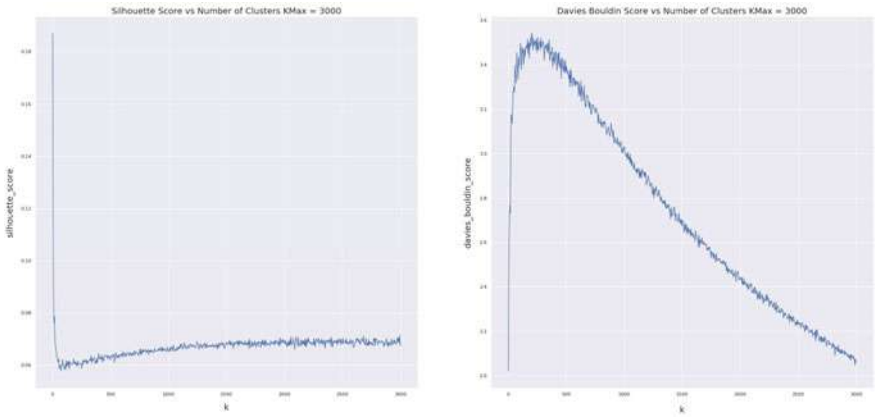
4 Results

The trained models are evaluated on clustering metrics—Silhouette score, Davies–Bouldin score and cosine distances of nearest papers. As a result of clustering, the dataset shows its underlying clustering patterns. All these characteristics of models are taken into consideration for evaluating, comparing them and recording outcomes. Glove word vectors used with P-Means and centroid embeddings are the two best performing models according to clustering metrics.

4.1 Clustering Quality with Respect to Dataset Size

As it can be seen in Fig. 2a, the clustering quality as measured by the Silhouette score decreases at first and then steadily increases with increase in the cluster size. Highest Silhouette score can be seen in the lower number of clusters.

As seen in Fig. 2b, Davies–Bouldin score is high for clusters 1 to 500, after that it decreases linearly.



(a) Silhouette Scores

(b) Davies Bouldin Scores

Fig. 2 Clustering metrics versus number of clusters

Table 1 Metrics for 30,000 papers and 350 clusters

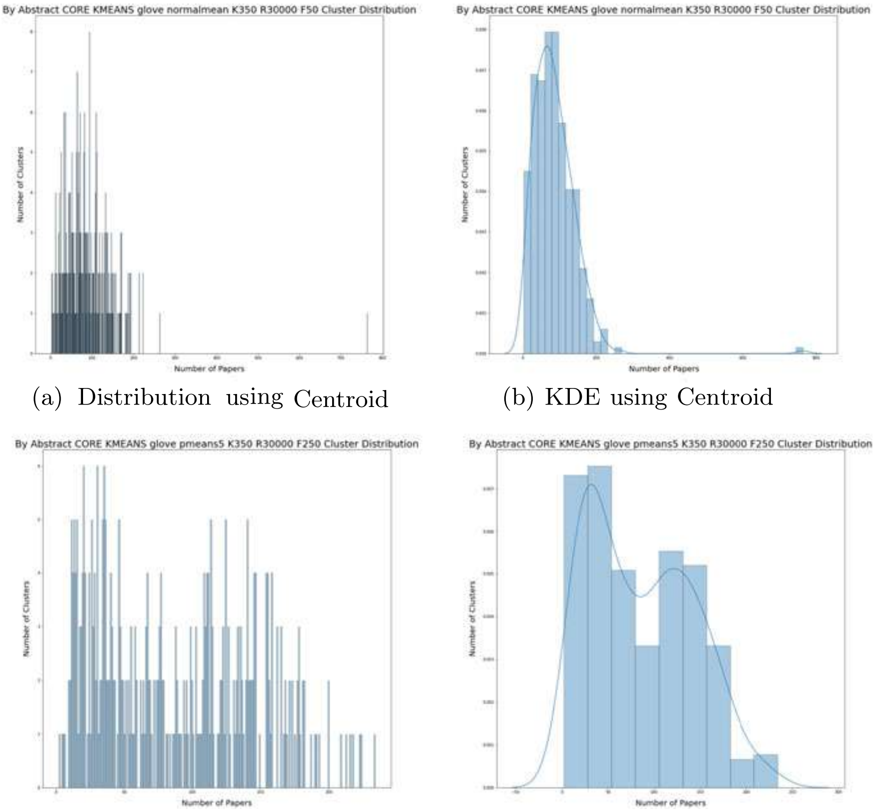
Sentence embedding	Word embedding	Silhouette score	Davis–Bouldin score
Centroid	GloVe	0.0814	2.4284
Centroid	Word2Vec	0.0740	2.5006
Centroid	FastText	0.0657	2.7902
P-Means	GloVe	0.0631	3.5175
P-Means	Word2Vec	0.0621	3.5023
P-Means	FastText	0.0574	3.8057

Table 2 Metrics for 15,000 papers and 250 clusters

Sentence embedding	Word embedding	Silhouette score	Davis–Bouldin score
Centroid	GloVe	0.0816	2.3878
Centroid	Word2Vec	0.0776	2.4808
Centroid	FastText	0.0674	2.7141
P-Means	GloVe	0.0635	3.3782
P-Means	Word2Vec	0.0617	3.3748
P-Means	FastText	0.0595	3.7380

4.2 Clustering Quality for Word and Sentence Embeddings

Clustering scores for different data sizes can be seen in Tables 1 and 2.



(a) Distribution using Centroid

(b) KDE using Centroid

(c) Distribution using P-Means

(d) KDE using P- Means

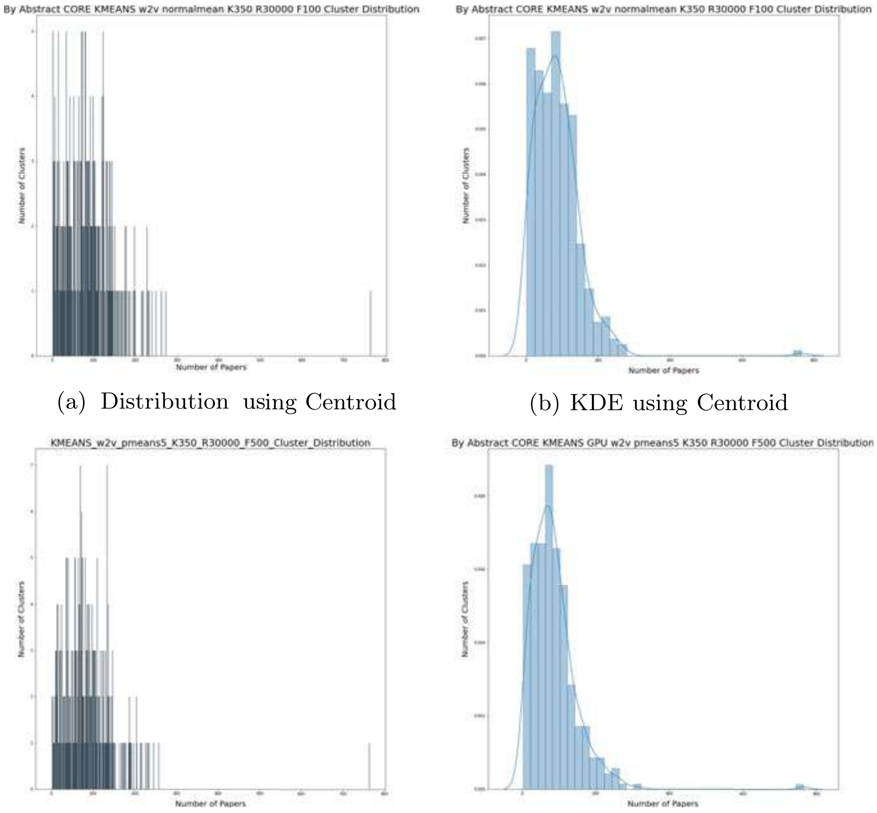
Fig. 3 Cluster distribution for CORE dataset using paper abstracts + GloVe (number of clusters vs. size of clusters)

4.3 Cluster Distributions of Datasets

In Figs. 3, 4 and 5, the distributions for the CORE dataset using the P-Means and centroid sentence embeddings for GloVe, Word2Vec and FastText embeddings are seen. The graphs show how many clusters are there containing a certain number of papers.

4.4 Similarity for CORE Dataset

Test paper and its top 2 closest papers from its cluster are given in Table 3.



(a) Distribution using Centroid

(b) KDE using Centroid

(c) Distribution using P-Means

(d) KDE using P-Means

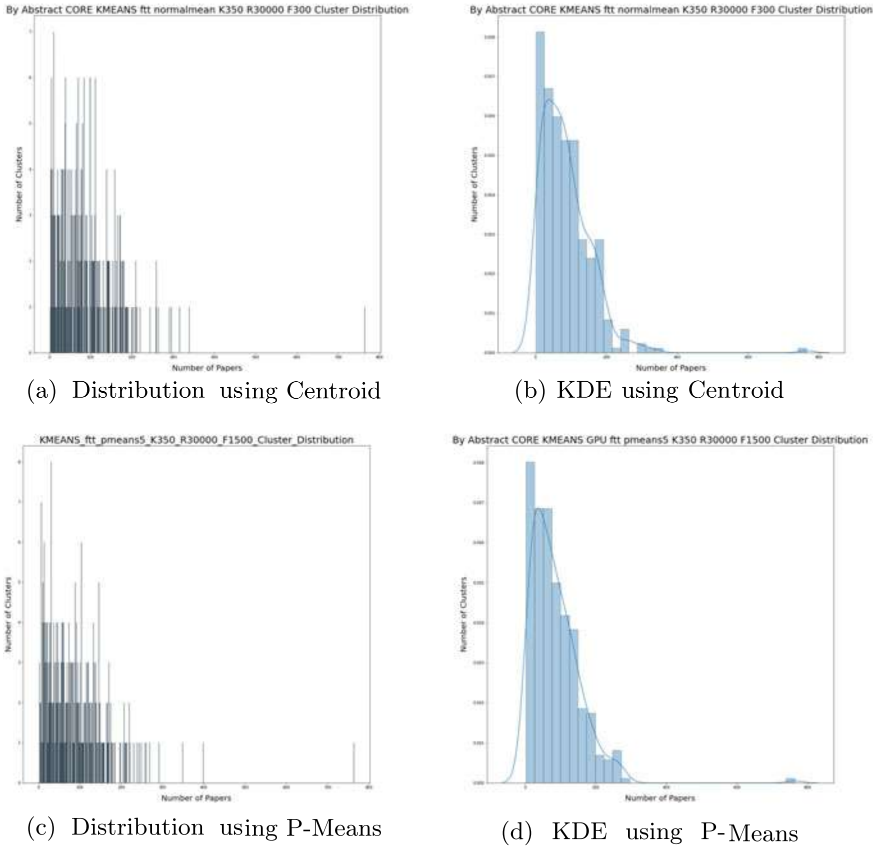
Fig. 4 Cluster distribution for CORE dataset using paper abstracts + Word2Vec (number of clusters vs. size of clusters)

5 Discussion

5.1 Clustering Quality as a Function of Dataset Size

Figure 2 shows that the silhouette score after 20 total number of clusters decreases suddenly. This inherently low clustering score is because of the so-called curse of dimensionality due to higher dimensional nature of sentence embeddings. Dimensions of vectors range from 50 to 1500. As the vectors go in higher dimensions, the points move further and further apart in high dimensional space. This leads to reduced clustering quality and performance.

The clustering score is relatively high for low total number of clusters, because the same number of papers can be fit in less number of clusters leading to more packed clusters. As the number of clusters increases, the average number of papers per



(a) Distribution using Centroid

(b) KDE using Centroid

(c) Distribution using P-Means

(d) KDE using P-Means

Fig. 5 Cluster distribution for CORE dataset using paper abstracts + FastText (number of clusters vs. size of clusters)

cluster decreases, leading to slow increase in clustering quality as there are multiple smaller distinct clusters. Although this is not desirable as this ultimately leads to clusters having only 10 papers per cluster ($K = 3000$) which is not at all useful for clustering papers in a segregating and practical sense, as this does not give any high to intermediate level of view of the whole dataset.

The intermediate range of number of clusters between square root and thrice the square root can be regarded useful for optimal number of clusters, since the clustering quality is steady and relatively acceptable and it is a practical number of clusters.

Table 3 Example similarity comparison in CORE dataset using cluster models made by different sentence embeddings

Word Emb and Sent. Emb.	Paper titles	Cosine distance from test paper
Test paper	Intracellular mechanisms underlying the nicotinic enhancement of LTP in the rat dentate gyrus	0
GloVe + centroid	The novel Syk inhibitor R406 reveals mechanistic differences in the initiation of GPVI and CLEC-2 signalling in platelets	0.0486
	An investigation into the role of neurotransmitter receptors in the function of human immune cells	0.0599
GloVe + P-Means	Phenylephrine preconditioning of isolated ventricular myocytes involves modulation of KATP channels through activation of survival kinases	0.0253
	The tyrosine phosphatase CD148 is an essential positive regulator of platelet activation and thrombosis	0.0254
Word2Vec + centroid	GLP-1 and Muscarinic Receptor Mediated Activation of ERK1/2 in Pancreatic B-cells	0.0271
	Biochemical investigation of phosphodiesterase type IV post-translational modification, cellular localisation and interaction with associated binding proteins	0.0280
Word2Vec + P-Means	A population of immature cerebellar parallel fibre synapses are insensitive to adenosine but are inhibited by hypoxia	0.0247
	Excitotoxic ATP and Glutamate Signalling during Central Nervous System Ischaemia	0.0251
Fastext + centroid	Phenylephrine preconditioning of isolated ventricular myocytes involves modulation of KATP channels through activation of survival kinases	0.0429
	Opposing changes in phosphorylation of specific sites in synapsin I during Ca2-dependent glutamate release in isolated nerve terminals	0.0447
Fastext + P-Means	Scanning peptide array analysis identify overlapping binding sites for the signaling scaffold proteins, beta-arrestin and RACK1 in the cAMP-specific phosphodiesterase	0.0256
	Novel areas of crosstalk between the cyclic AMP and PKC signalling pathways	0.0270

5.2 Clustering Quality for Word and Sentence Embeddings

It can be seen in Table 3, the difference between cosine distance for the nearest related paper for GloVe + P-Means model and GloVe + Centroid models is 47.94%. But the difference between in the Silhouette score, as seen in Table 1, for the same models is −22.49%. This shows that although the clustering quality for P-Means model is relatively less by a small margin, it is offset by the much improved Cosine distances of the most similar papers, thus showing a very high improvement in quality of individual clusters.

Relatively, the models which have very low dimensionality perform better than the models with high dimensionality. The models which use GloVe + Centroid and GloVe + P-Means sentence embeddings perform higher than the model which uses FastText + P-Means sentence embeddings, since first ones have dimension of only 50 and 250 while the latter has a dimension of 1500. It can be seen in Tables 1 and 2 that the Silhouette score decreases as we go from GloVe to Word2Vec to FastText. Silhouette score also decreases when we switch from centroid sentence embeddings to P-Means sentence embeddings. This happens because we go from lower dimensional embeddings to higher dimensional embeddings.

Authors recommend using low dimensional embeddings—from 50 to maximum 250.

5.3 Cluster Distributions of Datasets

When the papers are clustered by using their abstracts as keys, it can be seen in Figs. 3, 4 and 5 that a distinct cluster distribution for each model emerges. Even after using different sentence embeddings these distributions remain somewhat similar.

In Figs. 3, 4 and 5, it can be clearly seen that there is a skewed distribution of clusters, with a few clusters having a very high number of papers and then many clusters having a relatively low number of papers. This can be seen in the graph that there is a superimposed normal distribution near the left end of the X-axis and a single spike near the right end of the graph.

This spike near the right end of the graph in Fig. 3a represents clusters which have unusable or noisy data. After checking those clusters manually, it was found out that those clusters contain the papers which have ill-formatted abstracts with no content or content which only contains statements such as “full manuscript available at Springer/IEEE/Some Other Journal”, “Preprint Edition”, etc. These statements are put in place for incomplete, preprint and unpublished research papers. Since the CORE dataset is an industrial dataset sourced from many available public papers, it contains complete as well as incomplete papers. The distribution found near the left end of the X-axis for the CORE dataset represents the distribution of actual data.

These clustering distributions can be used as a fingerprint of the datasets which show their underlying structures of how the research papers are actually grouped into different subgroups other than just the ones manually set by keywords. These distributions can be used in exploratory data analysis and to compare different datasets for their quality and practical use. These distributions can also be used for noisy or unclean clusters removal.

5.4 Similarity

It can be seen in Table 3, the difference between cosine distance for the nearest related paper for GloVe + P-Means model and GloVe + Centroid models is 47.94%. This shows that there is a large improvement in clustering similar documents together.

As seen in Table 3, in which the papers closest to a given paper are shown, it can be seen that the clustering algorithm shows the papers which are related to the given query paper, in its respective field. Semantically, the clustering is effective as it gives the papers closest to the given query paper, despite having low Silhouette scores. The clustered research papers are very similar, they belong to the same field of science. This posits a hypothesis that the similarity of related research papers is calculated correctly, but the clusters of research papers are very far away from each other for this dataset.

6 Conclusion

The results show a clear increase of 47.94% in cosine distance of nearest papers using concatenated power means sentence embeddings with respect to baseline centroid embeddings for the highest performing GloVe models, with a marginal trade-off in Silhouette score. This shows that the computationally inexpensive sentence embeddings can be used for unsupervised clustering of scientific research papers using their abstracts.

It is also found that the proposed approach works well while clustering documents of smaller length, and it is further found that the abstracts clustered together represent similar topics even if the words in them may not be common, i.e. the semantics of the text is captured. As the document length increases, the information captured by the P-Means becomes murky.

The authors believe that the main drawback of using sentence embeddings is the very high dimensional representation of the underlying text. This is useful for other downstream tasks such as classification and sentiment analysis. But it is not useful for clustering because high dimensional data makes clustering inherently hard as points are spaced far apart from each other in higher dimensional space. Thus, sentence embeddings with dimensions between 50 and 250 are the most preferred as seen by GloVe + Centroid and GloVe + P-Means models in Tables 1 and 2.

This unsupervised clustering method can be used to fingerprint the datasets using their cluster distribution. The unsupervised clustering shows the underlying clusters of research papers which cannot be determined by simply grouping papers together by keywords or citation links as it uses the semantic meaning of the titles and abstracts for clustering.

7 Future Work

Performance can be improved by using word embeddings trained on relevant data. It should be noted that the approach of using sentence embedding to represent the textual data, and using these embeddings as the basis to find the similarity of the texts also works very well in many tasks under NLP like textual similarity, classification, text completion, summarising, translation, etc.

Word embeddings and sentence embeddings of reduced size can be used to cluster the papers, dimensionality reduction techniques such as principal component analysis, normalisation and scaling to redistribute the data points might also prove useful.

Concatenated power means should be tested on a labelled dataset, i.e. supervised clustering, to get accuracy, precision and recall scores to create a comparable metric. Reduction in dimensions of embeddings will also allow for rapid distribution fingerprinting.

References

1. Adamyan, L., Efimov, K.S., Chen, C., HHrdle, W.K.: Adaptive weights clustering of research papers. SSRN Electron. J. (2017). https://www.ssrn.com/abstract=2997061
2. Arora, S., Liang, Y., Ma, T.: A simple but tough-to-beat baseline for sentence embeddings (2017)
3. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Trans. Assoc. Comput. Linguist. **5**, 135–146 (2017). https://www.mitpressjournals.org/doi/abs/10.1162/tacl_a_00051
4. Boyack, K.W., Klavans, R.: Co-citation analysis, bibliographic coupling, and direct citation: which citation approach represents the research front most accurately? J. Am. Soc. Inf. Sci. Technol. **61**(12), 2389–2404 (2010). http://doi.wiley.com/10.1002/asi.21419
5. Boyack, K.W., Newman, D., Duhon, R.J., Klavans, R., Patek, M., Biberstine, J.R., Schijvenaars, B., Skupin, A., Ma, N., Börner, K.: Clustering more than two million biomedical publications: comparing the accuracies of nine text-based similarity approaches. PLoS ONE **6**(3), e18029 (2011). https://dx.plos.org/10.1371/journal.pone.0018029
6. Efimov, K., Adamyan, L., Spokoiny, V.: Adaptive nonparametric clustering. IEEE Trans. Inf. Theory **65**, 4875–4892 (2019)
7. Kessler, M.M.: Bibliographic coupling between scientific papers. Am. Doc. **14**(1), 10–25 (1963). https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.5090140103
8. Marshakova, I.V.: Citation networks in information science. Scientometrics **3**(1), 13–25 (1981). https://doi.org/10.1007/BF02021861
9. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv:1301.3781 [cs] (2013)

10. Newman, M.E.: Power laws, Pareto distributions and Zipf's law. Contemp. Phys. **46**(5), 323–351 (2005)
11. Nikfarjam, A., Sarker, A., O'Connor, K., Ginn, R., Gonzalez, G.: Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features. J. Am. Med. Inform. Assoc. ocu041 (2015). https://academic.oup.com/jamia/article-lookup/doi/10.1093/jamia/ocu041
12. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543. Association for Computational Linguistics, Doha, Qatar (2014). http://aclweb.org/anthology/D14-1162
13. Perone, C.S., Silveira, R., Paula, T.S.: Evaluation of sentence embeddings in downstream and linguistic probing tasks. arXiv:1806.06259 [cs] (2018)
14. Rücklé, A., Eger, S., Peyrard, M., Gurevych, I.: Concatenated power mean word embeddings as universal cross-lingual sentence representations. arxiv.org [cs] (2018)
15. Shibata, N., Kajikawa, Y., Takeda, Y., Matsushima, K.: Detecting emerging research fronts based on topological measures in citation networks of scientific publications. Technovation **28**(11), 758–775 (2008). http://www.sciencedirect.com/science/article/pii/S0166497208000436
16. Sjögårde, P., Ahlgren, P.: Granularity of algorithmically constructed publication-level classifications of research publications: Identification of topics. J. Informetr. **12**(1), 133–152 (2018). https://linkinghub.elsevier.com/retrieve/pii/S1751157717303371
17. de Solla Price, D.J.: Networks of scientific papers. Science **149**(3683), 510–515 (1965). https://www.sciencemag.org/lookup/doi/10.1126/science.149.3683.510

Semantic Sensitive TF-IDF to Determine Word Relevance in Documents

Amir Jalilifard, Vinicius Fernandes Caridá, Alex Fernandes Mansano, Rogers S. Cristo, and Felipe Penhorate Carvalho da Fonseca

Abstract Keyword extraction has received an increasing attention as an important research topic which can lead to have advancements in diverse applications such as document context categorization, text indexing and document classification. In this paper we propose STF-IDF, a novel semantic method based on TF-IDF, for scoring word importance of informal documents in a corpus. A set of nearly four million documents from health-care social media was collected and was trained in order to draw semantic model and to find the word embeddings. Then, the features of semantic space were utilized to rearrange the original TF-IDF scores through an iterative solution so as to improve the moderate performance of this algorithm on informal texts. After testing the proposed method with 160 randomly chosen documents, our method managed to decrease the TF-IDF mean error rate by a factor of 50% and reaching the mean error of 13.7%, as opposed to 27.2% of the original TF-IDF.

Keywords Semantic sensitive TF-IDF · Keyword extraction · Word relevance · Semantic similarity · TF-IDF

V. F. Caridá (✉) · A. F. Mansano · R. S. Cristo · F. P. C. da Fonseca
Data Science Team—Digital Customer Service, Itaú Unibanco, São Paulo, Brazil
e-mail: vinicius.carida@itau-unibanco.com.br

A. F. Mansano
e-mail: alex.mansano@itau-unibanco.com.br

R. S. Cristo
e-mail: rogers.cristo@itau-unibanco.com.br

F. P. C. da Fonseca
e-mail: felipe.fonseca@itau-unibanco.com.br

A. Jalilifard
Federal University of Minas Gerais, Belo Horizonte, Brazil
e-mail: jalilifard@ufmg.br

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_27

1 Introduction

In the information era when huge number of digital documents are gathered in a daily basis, going through documents and extracting the most relevant information, understanding the general concept and finding the other related documents is more necessary than ever. Keywords are several relevant words that provide a rich semantic information about a text for many natural language processing applications. Thereby, many researches have been carried out in order to extract the most relevant words from a text.

Some made use of the already-known supervised classification methods such as Support Vector Machine (SVM) and Naive Bayes [1, 2]. Although these supervised approaches methods provided good results, the need for training data, which often needs involving human resources, still remain a problem. Moreover, the word relevance score provided by a method like SVM may or may not be directly proportional to the importance of terms in a particular document. Therefore, an unsupervised method which provide local weights considering a class of documents is desirable.

Term frequency–inverse document frequency (TF-IDF) is one of the most commonly used term weighting schemes in information retrieval systems. Despite its popularity, TF-IDF has often been considered an empirical method, specifically from a probabilistic point of view, with many possible variations [3]. TF-IDF is a numerical statistics that, by scoring the words in a text, indicates how important a word is in a document considering the corpus that document belongs to. This method was studied in several researches for keyword and word relevance extraction (Fig. 1).

In this paper, we propose STF-IDF, a novel semantic sensitive method based on the conventional TF-IDF. The key idea is readjusting the conventional TF-IDF scores based on the semantic representation of most relevant words. Thereby, we assume that if a set of terms is considered important by TF-IDF, all the semantically similar words related to this set should be considered more important than those ones with less semantic relevance to the context. The next section explains the theoretical basis of TF-IDF and discuss about the related works. In the section three is explained the proposed method. The results and discussion are presented in the last section.

2 Related Works

The Vector Space Model, generally attributed to Salton et al. [4] and stemming from the Information Retrieval community, is arguably the most successful and influential model to encode words and documents as vectors [5]. Salton et al. [4] suggest an encoding procedure whereby each document in a collection is represented by a t-dimensional vector, each element representing a distinct term contained in that document. These elements may be binary or real numbers, optionally normalized using a weighting scheme such as TF-IDF, to account for the difference in information provided by each term.

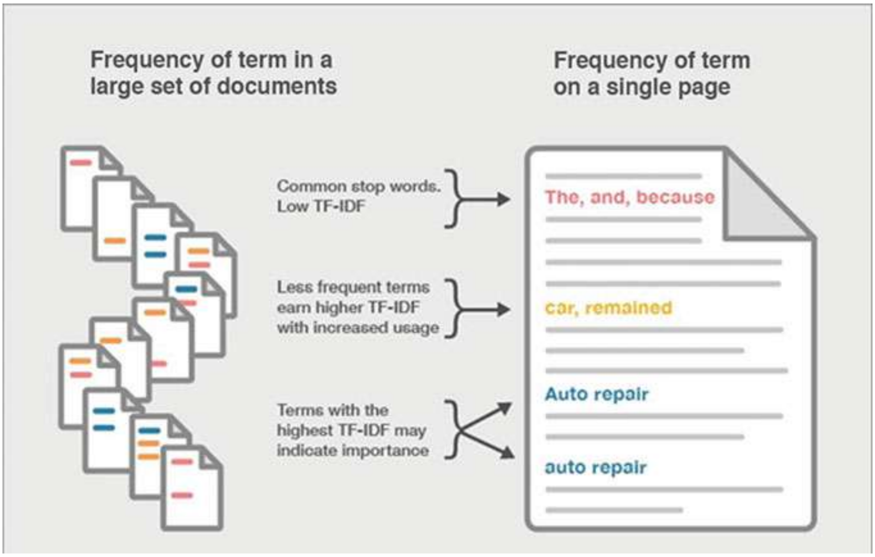

Frequency of term in a large set of documents
Frequency of term on a single page
Common stop words. Low TF-IDF
Less frequent terms earn higher TF-IDF with increased usage
Terms with the highest TF-IDF may indicate importance
The, and, because
car, remained
Auto repair
auto repair

Fig. 1 A simple example of TF-IDF. Adapted from: https://moz.com/blog/the-technical-seo-renaissance

2.1 An Overview of TF-IDF

Inverse document frequency (IDF) [6] is one of the most important and widely used concepts in information retrieval. It is used in combination with the term frequency (TF). The result is a very effective term weighting scheme that has been applied for information retrieval systems [7]. Essentially, TF-IDF works by determining the relative frequency of words in a specific document compared to the inverse proportion of that word over the entire document corpus. Intuitively, this calculation determines how relevant a given word is in a particular document. Words that are common in a single or a small group of documents tend to have higher TF-IDF numbers than common words such as articles and prepositions [8]. The formal procedure for implementing TF-IDF has some minor differences over all its applications, but the overall approach works as follows: given a document collection D, a word w, and an individual document $d \epsilon D$, calculate:

$$w_d = f_{w,d} * \log(|D|/f_{w,D}) \tag{1}$$

where $f_{w,d}$ equals the number of times w appears in d, $|D|$ is the size of the corpus, and $f_{w,D}$ equals the number of documents in which w appears in D [9, 10]. There are a few different situation that can occur here for each word, depending on the values of $f_{w,d}|D|$, and $f_{w,D}$, the most prominent of which we'll examine. Assume that $|D|$ $f_{w,d}$ i.e. the size of the corpus is approximately equal to the frequency of w over D. If $1 < \log(|D|/f_{w,D})$ for some very small constant c, then w_d will be smaller than $f_{w,d}$ but still positive. This implies that w is relatively common over the entire corpus but still holds some importance throughout D. For example, this could be the case if TF-IDF would examine the word Jesus over the New Testament. More relevant to us, this result would be expected of the word United in the corpus of United Nations documents. This is also the case for extremely common words such as articles, pronouns, and prepositions, which by themselves hold no relevant meaning in a query (unless the user explicitly wants documents containing such common words). Such common words thus receive a very low TF-IDF score, rendering them essentially negligible in the search [8]. Finally, suppose $f w, d$ is large and $f w, D$ is small. Then $\log(|D|/f w, D)$ will be rather large, and so w_d will likewise be large. This is the case were most interested in, since words with high w_d imply that w is an important word in d but not common in D. This w term is said to have a large discriminatory power. Therefore, when a query contains this w, returning a document d where w_d is large will very likely satisfy the user.

The code for TF-IDF is elegant in its simplicity. Given a query q composed of a set of words w_i, we calculate $w_{i,d}$ for each w_i for every document $d \epsilon D$. In the simplest way, this can be done by running through the document collection and keeping a running sum of $f_{w,d}$ and $f_{w,D}$. Once done, we can easily calculate w_{id} according to the mathematical framework presented before. Once all w_{id}'s are found, we return a set $D*$ containing documents d such that we maximize the following equation

$$\sum_i w_{i,d} \tag{2}$$

Either the user or the system can arbitrarily determine the size of $D*$ prior to initiating the query. Also, documents are returned in a decreasing order according to Eq. (2). This is the traditional method of implementing TF-IDF [8].

2.2 Related Works

TF-IDF, in general, is one of the most used techniques to quantify a word in documents. However, there are variations in some methods. Singh and Dwivedi [7] compared four methods of IDF found in literature using a set of TREC based queries. After computing the value of the IDF using keyword based search, they concluded that the IDF value computed by [11] gives better TF-IDF weight of terms compared to other methods.

Ramos and his colleagues [12] examined the result of applying TF-IDF in determining the word relevance in document queries and concluded that this simple method efficiently classifies relevant words. Li et al. [13] applied TF-IDF for keyword extraction in Chinese texts based on analyzing linguistic characteristics of documents and providing several strategies including uni-, bi- and tri-gram extraction, new word finding and refinement. Wu et al. [14] proposed a probabilistic model based on TF-IDF which makes local relevance decisions for each location in a document and combines these local relevant decisions into a document wide relevance decision. Lee and colleagues [15] presented several variants of conventional TF-IDF for a more effective keyword extraction and topic summarization. They used cross-domain comparison for removing meaningless or irrelevant words.

Despite the advances in word embeddings (i.e. word vector representations), capturing sentence meaning is an open question due to complexities of semantic interactions among words [16]. Even papers that use embeddings, some times, use TF-IDF to help to fitted The weights of the series. Another challenge in document classification is insufficient label information and sparse, unstructured format. With the intention of increasing the variety of resource sets for classification, the authors [17] used three methods of document representation, including TF-IDF and proposed multi-co-training (MCT). In order to help the categorization of documents into hierarchical structures showing the relationship between variables [18] proposes a novel TF-IDF algorithm with the temporal Louvain approach.

Although the aforementioned methods improved the performance of conventional TF-IDF by providing probabilistic solutions or the use of multi-strategies, they are highly dependent on the original TF-IDF idea which is giving more weight to the words with high local and less global probability. This consideration specially fails when it comes to finding word relevance in informal documents, which are important sources of information in the era of social networks [19]. As an example, if a text contains informal words related to a specific ethnic communities, or words that have been used in a specific period of time, but not in the whole corpus, due to some changes in cultural expressions, both conventional TF-IDF and the related methods that attempt to improve its performance fail to find relevant words with high accuracy. Another example is informal conversations regarding formal topics like medical communities that provide rich information for users. Censuring the general semantic context, TF-IDF fails to detect the context-sensitive content which plays an important role in informal texts [20].

3 Materials and Method

In this section we explain the materials and the mathematical definition of our proposed method. We start with the data acquisition and then we explain how our algorithm tries to improve the conventional TF-IDF through a finite numbers of iterations.

Data of nearly four million pages of online medical communities was gathered and after being pre-processed (i.e. removing stopwords, punctuation, etc.), they were fed to word2vec [21] in order to learn the semantic space and words' distribution. Having the semantic distribution of terms of the corpus, our algorithm generates the word relevance score as following:

$$S_{\mathrm{wj}}^{(k)} = S_{\mathrm{wj}}^{(k-1)} \times \frac{1}{1 + \left\| e(w_j) \right\| \left\| \overline{e(w)}) \right\| \cos(\Theta)} \tag{3}$$

where $S_{\mathrm{wj}}^{(k)}$ is the vector of word scores in Kth iteration and the initial scores are calculated using the conventional TF-IDF:

$$S_{\mathrm{wj}}^{(0)} = P(W_j)_d * \mathrm{Log}(P(W_j)_c) = \mathrm{TFIDF}_{\mathrm{Wj}} \tag{4}$$

and $\overline{e(w)}$ is the weighted expected value of the first $\lfloor\sqrt{n}\rfloor$ most relevant words in the previous iteration and is calculated as follow:

$$\overline{e(w)} = \frac{1}{\lfloor\sqrt{n}\rfloor}\sum_{i=1}^{\lfloor\sqrt{n}\rfloor}\left(\frac{1}{1-\frac{S_{wj}^{(k-1)}}{\sum_{j=1}^{n} S_{wj}^{(k-1)}}} \times e(w_j)\right) \tag{5}$$

The algorithm is initiated with conventional TF-IDF scores. In each iteration, first, the mean embedding of $\lfloor\sqrt{n}\rfloor$ most relevant words from previous iteration are selected and the weighted mean embedding of them is calculated. The idea is that words with higher scores represent the context more than those which are less relevant. In order to calculate the weighted mean embedding, for each word its score in the previous iteration is divided by all the scores in order to get a number in the range [0, 1]. This weight then is converted to a number greater than 1 and is considered the weight by which each word pushes the mean embedding toward itself. Afterwards, the previous scores are recalculated by being multiplied on the cosine distance of the word and the mean embedding. The idea is to repeatedly replace the words that have poor representation of general text context with those that are more related to the document context. The new scores are then passed to the next iteration and the scores are rearranged so that the words with better representation are moved toward the top of ranking.

By considering the embedding of the first $\lfloor\sqrt{n}\rfloor$ most relevant words as a multivariate unknown probability distribution, the less-relevant words are those instances which increase the variance of the distribution. Therefore, the goal is to decrease the variance by constantly replacing the outliers with those words that are semantically more related to the most important words and move the expected value of embeddings toward the value that perfectly matches the context. First, we prove that in each iteration the variance of the distribution from the mean in the previous iteration is decreased. Then, it is shown that in each iteration the mean value of set moves toward the mean of ideal distribution until it converges.

We define the mean embedding of the observed data in each iteration and the unknown distribution that best fits the context with $\mu^{(i)}$, and μ, respectively, as following:

$$\mu^{(i)} = \frac{X_1^{(i)} + X_2^{(i)} + X_3^{(i)} + \cdots + X_{m-1}^{(i)} + X_m^{(i)}}{m} \tag{6}$$

where $m = \lfloor\sqrt{n}\rfloor$ and $X_m^{(i)}$ represents the embedding of mth word of the set in the ith iteration and $\mu^{(0)}$ as the mean of the initial distribution generated by conventional TF-IDF, and $X_m^{(i)} \neq X_m^{(i-1)}$ if a word is substituted.

In each iteration either the set maintains the current members or some of them are replaced with words that have a less cosine distance. Multiplying the word score from previous iteration $S_{\mathrm{wj}}^{(k-1)}$ to the inverse of cosine distance guarantees that a word with higher cosine distance from the expected value of embeddings is decreased more than a word with less distance. A factor of 1 is added to the equation in order to eliminate the reverse effect of distances lesser than 1.

In the simplest case, we assume that in each iteration one word is replaced with another. As it was explained, Eq. (3) guarantees that the new word has less cosine distance to the expected value of embeddings than the replaced word. As a result:

$$\frac{1}{m}\sum_{k=1}^{m}(X_k^{(i)} - \mu^{(i-1)})^2 < \frac{1}{m}\sum_{k=1}^{m}(X_k^{(i-1)} - \mu^{(i-1)})^2 \Rightarrow \mathrm{Var}(X^{(i)}) < \mathrm{Var}(X^{(i-1)}) \tag{7}$$

$$\Rightarrow \lim_{i\to\infty}\frac{\mathrm{Var}(X^{(i-1)})}{\mathrm{Var}(X^{(i)})} = 1 \tag{8}$$

Let's assume that the real context mean is bigger than the initially estimated mean by TF-IDF and one distant member per iteration is replaced with a closer one, say X_m in ith iteration, Then:

$$\begin{aligned} X_m^{(i)} > X_m^{(i-1)} \Rightarrow & \frac{X_1^{(i-1)} + X_2^{(i-1)} + X_3^{(i-1)} + \cdots + X_{m-1}^{(i-1)} + X_m^{(i)}}{m} \\ & > \frac{X_1^{(i-1)} + X_2^{(i-1)} + X_3^{(i-1)} + \cdots + X_{m-1}^{(i-1)} + X_m^{(i-1)}}{m} \end{aligned} \tag{9}$$

and consequently:

$$\begin{aligned} \mu^{(i)} > \mu^{(i-1)} > \cdots > \mu^{(1)} > \mu^{(0)} \Rightarrow (\mu - \mu^{(i)}) < (\mu - \mu^{(i-1)}) < \cdots \\ < (\mu - \mu^{(1)}) < (\mu - \mu^{(0)}) \end{aligned} \tag{10}$$

approaching the correct context's expected embedding μ in each iteration. In case of $\mu < \mu^{(0)}$:

$$\begin{aligned} & \mu^{(i)} < \mu^{(i-1)} < \cdots < \mu^{(1)} < \mu^{(0)} \\ & \Rightarrow (\mu^{(i)} - \mu) < (\mu^{(i-1)} - \mu) < \cdots < (\mu^{(1)} - \mu) < (\mu^{(0)} - \mu) \end{aligned} \tag{11}$$

Finally, from (8):

$$\lim_{i\to\infty}\frac{X_k^{(i-1)}}{X_k^{(i)}} = 1 \Rightarrow \lim_{i\to\infty} \mid X_k^{(i)} - X_k^{(i-1)} \mid = 0 \Rightarrow \lim_{i\to\infty} \mid \mu^{(i)} - \mu^{(i-1)} \mid = 0 \quad (12)$$

and the condition of almost surely convergence is met after enough number of iterations:

$$\left|\mu - \mu^{(i)}\right| \leq \varepsilon \quad (13)$$

There are two ways that the proposed method can fail in improving the word rank. First, if the mth and $(m+1)$th words have exactly same score. In this case, the choice of words may change the expected embedding value and consequently lead to a totally different approximation of the document context. In order to solve this problem, the algorithm may simply check the score of mth and the $(m+1)$th words and in case of encountering the same scores, it can enter the $(m+1)$th word into the set as well. Second, if before starting the refining process the conventional TF-IDF produces scores with very high error rate, STF-IDF fails to find the correct context. Nevertheless, our results show that such a high error rate is not a common case and a moderate performance of TF-IDF is enough for the current method to produce significantly good results.

4 Results and Discussion

The algorithm was tested for 160 randomly chosen informal medical documents. For both conventional TF-IDF and STF-IDF, the scores were evaluated with human annotated labels. Since the precision of word importance can be subjective, here, we define and analyze the ranking error which measures the number of words were put between the first $\lfloor\sqrt{n}\rfloor$ most relevant terms while they have the least relevance based on human evaluation.

As it is seen in Fig. 2, by replacing better words in the ranking table, STF-IDF has less error rate in comparison with TF-IDF. Since STF-IDF is initially constructed upon the TF-IDF scores, in the rare cases (less than 5% of times) when TF-IDF has abnormally big error rate, STF-IDF performs worse than TF-IDF.

We measured the error rate of STF-IDF against the original TF-IDF. As it is show in Fig. 3, our method improved the error rate by more than 50%, decreasing the error rate of TF-IDF from 27.68 to 13%. Among the rankings generated for 160 documents, in 50% of times the error rate of STF-IDF is significantly less ranging from 0 to 13% as opposed to high error of conventional TF-IDF ranging from 20 to 30% for 50% of tested documents.

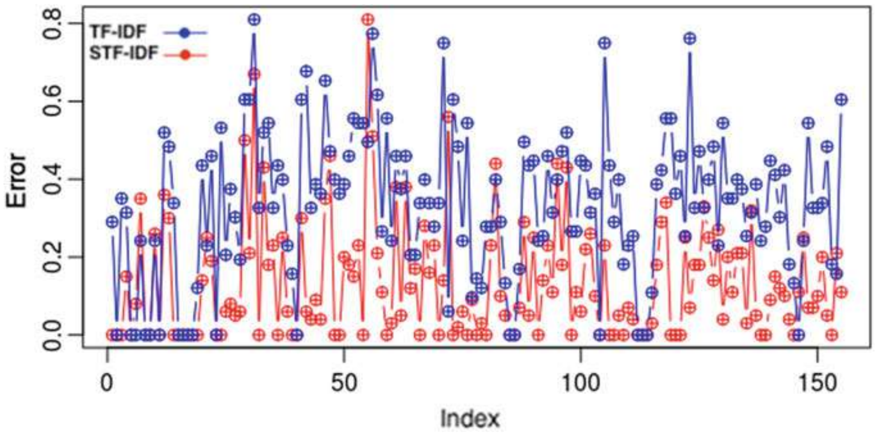


Fig. 2 The error rate of STF-IDF in comparison with conventional TF-IDF for each document

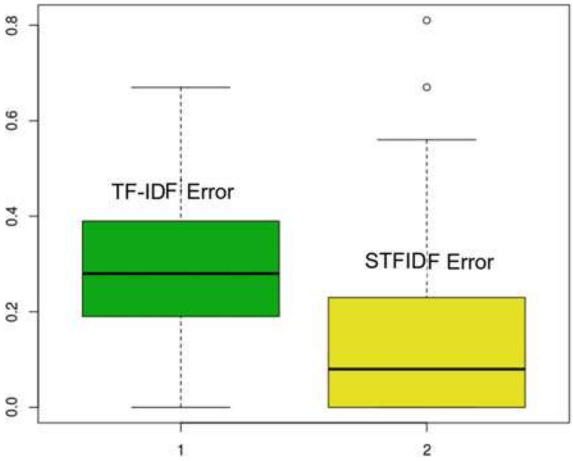


Fig. 3 The boxplot of ranking error for STF-IDF and conventional TF-IDF

5 Conclusion

TF-IDF is an efficient and simple algorithm for matching words in a query to documents that are relevant to that query. From the data collected, TF-IDF returns documents that are highly relevant to a particular query. If a user were to input a query for a particular topic, TF-IDF can find documents that contain relevant information on the query. Furthermore, encoding TF-IDF is straightforward, making it ideal for forming the basis for more complicated algorithms and query retrieval systems [10].

Despite its strength, TF-IDF has its limitations. In terms of synonyms, notice that TF-IDF does not make the jump to the relationship between words. Going back to [22]. In some experiments, TF-IDF could not equate the word .drug. with its plural .drugs., categorizing each instead as separate words and slightly decreasing the word.s wd value. For large document collections, this could present an escalating problem.

In this study we proposed a novel method based on semantically weighted TF-IDF scores for finding word relevance between a collection of documents. Textual data of nearly 4 million online medical communities were gathered and preprocessed. Afterwards, the corpus was fed to word2vec algorithm in order to generate word embedding. Initially, the words were ranked by conventional TF-IDF algorithm. Then these scores were repeatedly modified based on a semantic weight of each word proportional to the cosine distance of the world and the expected value of embedding of a set of most relevant words in each iteration. The algorithm stops when it reaches a predefined threshold which is a measure of dislocation of mean embedding distribution. Our results show a significant decrease in error rate when STF-IDF is utilized. The future works will be focused on the convergence proof of the algorithm as well as replacing automatic tests with human-involved evaluation.

Conflict of Interest The current method was proposed and tested by a group of data scientists from Itaú Unibanco. Any opinions, findings, and conclusions expressed in this manuscript are those of the authors and do not necessarily reflect the views, official policy or position of Itaú Unibanco.

References

1. Zhang, K., Xu, H., Tang, J., Li, J.: Keyword extraction using support vector machine. In: International Conference on Web-Age Information Management, pp. 85–96. Springer (2006)
2. Uzun, Y.: Keyword extraction using naive Bayes. In: Bilkent University, Department of Computer Science, Turkey. www.cs.bilkent.edu.tr/guvenir/courses/CS550/Workshop/Yasin_Uzun.pdf (2005)
3. Aizawa, A.: An information-theoretic perspective of TF-IDF measures. Inf. Process. Manag. **39**(1), 45–65 (2003)
4. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM **18**, 613–620
5. Almeida, F., Xexéo, G.: Word embeddings: a survey. arXiv eprint 1901.09069 (2019)
6. Papineni, K.: Why inverse document frequency. In: Proceedings of the North American Association for Computational Linguistics, pp. 25–32 (2001)
7. Singh, J., Dwivedi, S.K.: Comparative analysis of IDF methods to determine word relevance in web document. Int. J. Comput. Sci. **11**(1) (2014). ISSN: 1694-0784
8. Ramos, J.: Using TF-IDF to Determine Word Relevance in Document Queries (2003)
9. Salton, G., Buckley, C.: Term-weighing approaches sin automatic text retrieval. Inf. Process. Manag. **24**, 513–523 (1988)
10. Berger, A., et al.: Bridging the lexical chasm: statistical approaches to answer finding. In: Proceedings of the International Conference on Research and Development in Information Retrieval, pp. 192–199 (2000)
11. Jung, Y., Park, H., Du, D.: An effective term-weighting scheme for information retrieval. Technical Report TR00-008, Department of Computer Science and Engineering, University of Minnesota (2000)

12. Ramos, J., et al.: Using TF-IDF to determine word relevance in document queries. In: Proceedings of the First Instructional Conference on Machine Learning, vol. 242, pp. 133–142 (2003)
13. Li, J., Zhang, K., et al.: Keyword extraction based on TF/IDF for Chinese news document. Wuhan Univ. J. Nat. Sci. **12**(5), 917–921 (2007)
14. Wu, H.C., Luk, R.W.P., Wong, K.F., Kwok, K.L.: Interpreting TF-IDF term weights as making relevance decisions. ACM Trans. Inf. Syst. (TOIS) **26**(3), 13 (2008)
15. Lee, S., Kim, H.-j.: News keyword extraction for topic tracking. In: Fourth International Conference on Networked Computing and Advanced Information Management, 2008. NCM'08, vol. 2, pp. 554–559. IEEE (2008)
16. Arroyo-Fernández, I., Méndez-Cruz, C.-F., Sierra, G., Torres-Moreno, J.-M., Sidorov, G.: Unsupervised sentence representations as word information series: revisiting TF-IDF. Comput. Speech Lang. **56**, 107–129 (2019)
17. Kim, D., Seo, D., Cho, S., Kang, P.: Multi-co-training for document classification using various document representations: TF-IDF, LDA, and Doc2Vec. Inf. Sci. **477**, 15–29 (2019)
18. Iwendi, C., Ponnan, S., Munirathinam, R., Srinivasan, K., Chang, C.-Y.: An efficient and unique TF/IDF algorithmic model-based data analysis for handling applications with big data streaming. Electronics **8**, 1331 (2019)
19. Morris, M.R., Teevan, J., Panovich, K.: A comparison of information seeking using search engines and social networks. ICWSM **10**, 23–26 (2010)
20. Wöllmer, M., Eyben, F., Graves, A., Schuller, B., Rigoll, G.: Bidirectional LSTM networks for context-sensitive keyword detection in a cognitive virtual agent framework. Cogn. Comput. **2**(3), 180–190 (2010)
21. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
22. Berger, A., Lafferty, J.: Information retrieval as statistical translation. In: Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval, pp. 222–229 (1999)

Web-Based Interactive Neuro-Psychometric Profiling to Identify Human Brain Communication and Miscommunication Processing

Arthur F. Carmazzi and Phakkharawat Sittiprapaporn

Abstract This study investigated the effect of individual brain communication processes on interpersonal communication and potential miscommunication by using a web-based interactive neuro-psychometric profiling tool, named Colored Brain Communication Inventory (CBCI). The brain communication process is influential patterns of individual communication and cultivates the potential of individuals to career or work in the future. The methodology used in this study was quantitative, surveys, and observation studies. The aim of this study was then to explore the brain clarity processing and distinguishing the miscommunication assessed by web-based interactive neuro-psychometric profiling instruments. All respondents involved in this study were equally divided between gender; males (50%) and females (50%). The impact of this study has practical implications for the respondents' communication behavior toward individual developments. The impact on science was to know the correlation between the brain communication process on gender, creativity, and communication behavior by means of potential miscommunication process. The output of this study was a type of brain communication process that shows the tendency of nature, attitudes, and individual potential.

Keywords Brain · Colored brain · Directive communication psychology

Contributors A. F. Carmazzi originated the concept; collected data; conducted the research. P. Sittiprapaporn designed the study; critically analyzed data; conducted the research. A. F. Carmazzi and P. Sittiprapaporn performed a literature search; drafted the manuscript. P. Sittiprapaporn critically revised the final version. A. F. Carmazzi and P. Sittiprapaporn contributed equally to this work; approved the final version.

A. F. Carmazzi
Directive Communication International, Bali, Indonesia
e-mail: arthur@directivecommunication.com

P. Sittiprapaporn (✉)
Brain Science and Engineering Innovation Research Group, Mae Fah Luang University, Bangkok, Thailand
e-mail: wichian.sit@mfu.ac.th

Neuropsychological Research Laboratory, Department of Anti-Aging and Regenerative Science, School of Anti-Aging and Regenerative Medicine, Mae Fah Luang University, Bangkok, Thailand

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_28

1 Introduction

Generally known that the human brain is the center of the all internal communication process of commands in which it moves messages to all parts of the body. So, the human brain as consisting of 1,000,000,000,000 neurons is then the greatest processor [1]. However, communication is still difficult to define where the world is abstract and like most terms possesses multiple meanings [1]. Communications can be separated into several communication styles which are a window to understand how the world views us affecting our relationship, career, and emotional well-being [1]. Studying the communication style with the personal communication behavior trends of a person can be learned from several methods. Among others, the brain communication and processing were introduced by Carmazzi in 1999 [2] which were becoming increasingly popular as an inventory for an organization's individual profile including attitude, creativity, leadership qualities, how to deal with conflicts, relationships with others, how to communicate based on skills and measure personal responsibilities, respectively. The implementation phenomenon in the field of mapping brain processing to the communication style and the suitability of potential in work is still an interesting new thing to prove, for example, people with different brain communication and processing have a tendency behavior creative, innovative, communicative, and full attention, respectively [1]. In 2017, Iswanti and colleagues studied the effectiveness of communication, creativity, nature, and potential based on brain communication. They classified several types of communications including assertive communication, aggressive communication, and passive or submissive communication. Iswanti and colleagues reviewed that the assertive communication has confidence communication, responsible, medium voice, open gesture, respect others while the aggressive communication has most important, fast, voice loud, gesture bigger, explosive, demanding, and out to win the others. While the passive-aggressive communication is acting out their anger in indirect or behind the scenes ways, sarcastic, complaining, indirectly aggressive, gossip, voice often sweet voice, and gesture quick with a sweet and innocent expression, the passive or submissive or passive communication behaved other people needs are more important and more contribute, apologetic, avoiding any confrontation, voice volume including soft, gesture twist and fidget, and no eye contact [1].

As indicated by Carmazzi's investigations, Carmazzi demonstrated a functional model of how individuals were recognized from others by their own particular method of processing their general surroundings, handling information, learning and critical thinking, communicating, and identifying with others [2–4]. Carmazzi speculated that individuals were recognized by others by their own particular method of processing their general surroundings, handling information, learning, solving problems, communicating, and identifying with others [5, 6]. This brain communication process which is in the form of a classification of "neuro-genetic processing" is defined as the way each individual experiences the environment, retrieving and interpreting information, resulting in specific ways of observing and evaluating events and situations as well as showing additional communication with others [6]. In other words, this brain communication refers to the way individual encounters and interprets information in the brain, which brings about explicit methods of perceiving and assessing information [6, 7]. Carmazzi has then distinguished four essential hereditary brain processes that were related to these past investigations to build up a profiling instrument known as the web-based interactive neuro-psychometric profiling apparatus to recognize the brain processing indicator [2, 6]. This brain communication process comprises four different brain processing including chaotic, linear, relational, and intuitive brain communication processes, respectively. These brain processing indicators are utilized to depict the various attributes of our hereditary processing and mental adaptability or communication style [2, 5, 8]. People with a chaotic brain communication process perceive information as a whole big picture. Their thinking is a non-straight random chunk where nothing is associated; these chunks of data structure a variety of numerous methods of moving toward the same thing. The chaotic brainers are related to inventive and imaginative methods of solving a problem. The linear brainers perceive and process information in a direct manner through cross-referencing. They are calm, work on logic, and are objective-driven. They are specific about gathering data, arranging them in an organized way, and inspecting them diagnostically before settling on any decision. The linear brainers work on a clear sense of purpose and direction [7]. The function of studying brain communication and processing includes the workings of the brain are genetically easier to learn and teach, overcome the problem misconceptions, overcoming conflict, and more harmony, and waste of individual and group potential performance, respectively [3]. Therefore, the objective of the present study was to investigate the brain clarity processing and identify the brain miscommunication process assessed by web-based interactive neuro-psychometric profiling tool. The research question investigated the intercession impact of brain processes by utilizing the web-based interactive neuro-psychometric profiling apparatus.

2 Materials and Methods

2.1 Participants

The study employed purposive sampling with respondents with specific characteristics. The sample included respondents who were online registered to the web-based interactive neuro-psychometric profiling tool until 1999. The group of respondents were equally divided into male and female respondents. All respondents consented to be assessed their thinking process utilizing the Colored Brain Communication Inventory (CBCI) which was the primary instrument in this investigation.

2.2 Design

This is a descriptive report that utilized a quantitative research methodology. Correlational analysis was used to explore the relationship between the brain communication and potential miscommunication processes to each other by using the statistical program. A survey design was utilized for information collection in this investigation. The implementation adopted the descriptive design to review and survey previous studies as well as studies related to specific factors as identified in both the brain process and the web-based interactive neuro-psychometric profiling. A survey design was adopted to identify the respondents' brain communication process to facilitate dividing them into potential miscommunicate brain processing groups regarding to their gender differences. The survey design was chosen in light of the fact that the current study was resembles an experimental study. According to Cook and Campbell [9], although the independent variable was manipulated, respondents were not randomly assigned to conditions or orders of conditions. The instruments used to gather the information were embraced from The Directive™ Communication Colored Brain Inventory on 5 Likert-scale which was created by Carmazzi [4]. The colored brain is an accredited methodology by the American Institute of Business Psychology (https://coloredbrain.com/benefits-of-colored-brain) which aimed toward learning the foundations of hereditary brain processing. The Colored Brain Communication Inventory (CBCI) was used as the main instrument in this study in order to identify respondents' brain communication processes. CBCI is thus the tool to identify and categorize those different brain communication processes [10]. CBCI is also the apparatus to recognize and arrange those diverse mind correspondence measures [10]. It is a profiling tool and different from other tools as the focus of this tool addresses both "nature" and "nurture" in the formation of processing characteristics in discovering the genetic architecture that supports the brain communication process. This tool also establishes the "brain flexibility" (unnatural characteristics that have been developed through the environment) that crosses into the realm of environmental emotional factors [10–13]. There are four measured criteria in CBCI including (a) natural mental processing strength: the natural genetic strengths that determine how the candidate's brain processes information, and how he/she will take action in a given career situation, (b) learned mental processing strengths: the learned ability to get results in an area unrelated to the candidate's natural brain processing, (c) mental flexibility: the ability to productively work and communicate with others that operate with different mental processors, and (d) communication improvement areas: determining what brain processing types are difficult for a candidate to mix with is instrumental in preventing potential problems in management and teamwork, respectively [12, 13] (see Fig. 1).

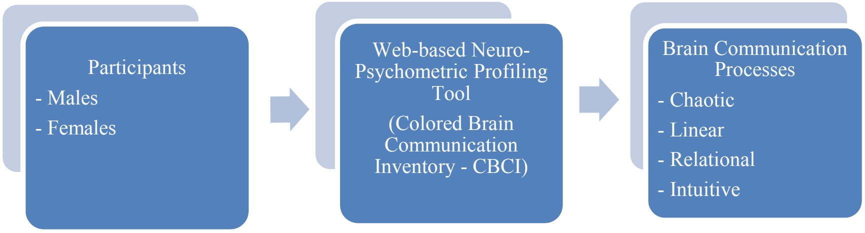

Participants
- Males
- Females
Web-based Neuro-Psychometric Profiling Tool
(Colored Brain Communication Inventory - CBCI)
Brain Communication Processes
- Chaotic
- Linear
- Relational
- Intuitive

Fig. 1 Graphical presentation of the design of the study

2.3 *Statistical Analysis*

Investigated the fundamental demographic information of respondents by utilizing descriptive examination with qualitative information summarized in terms of frequency and percentage. Descriptive analyses were performed for demographics utilizing characteristic measures. Inferential statistic was used to compare each brain communication process and miscommunication process frequency and percentage. Additionally, descriptive and inferential statistics were used to examine the correlation between the variables by using a statistical program.

3 Results

3.1 *Demographic Information of the Respondents*

Respondents who registered to CBCI were involved in this study. All respondents were equally divided between gender; males (50%) and females (50%). For brain communication process identification, we explored the relationship between brain communication process identification and communication root causes using the web-based interactive neuro-psychometric profiling. All respondents agreed to be assessed their way of brain communication process using this web-based interactive neuro-psychometric profiling tool. As shown in Fig. 2, the observations of the characteristics of each of the four identified brain communication processes were labeled to simplify classification with their specific characteristics. The following numbers were the numbers of different brain communication process viewed by all respondents. Each brain communication process with each total score was the respondents' foundational brain communication process and communication root characteristics.

Based on Table 1 and Fig. 2 (Left), of the respondents who were assessed by the web-based interactive neuro-psychometric profiling tool, the highest percentage number of 64 (male = 34; female = 30) were identified as "chaotic brain communication process" while about 52% of respondents (male = 26; female = 26) were

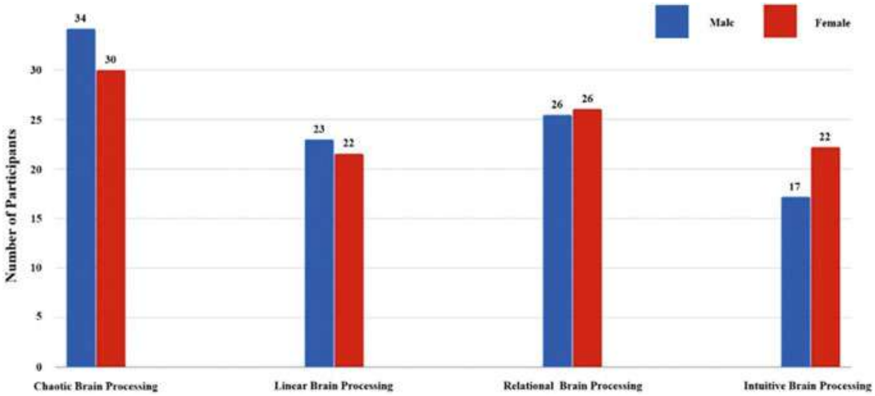


Fig. 2 Demographic information of each brain communication process identified by gender

Table 1 Demographic information of each brain communication process identified by gender

Brain communication process	Male (%)	Female (%)
Chaotic communication	34	30
Linear communication	23	22
Relational communication	26	26
Intuitive communication	17	22
Total	100	100

identified as "relational brain communication process." The less of 45% of respondents (male = 23; female = 22) were identified as "linear brain communication process" while only 39% of respondents (male = 17; female = 22) were identified as "intuitive brain communication process," respectively.

In addition, after considering the factor of gender differences, it was found that of all male respondents who were assessed by the web-based interactive neuropsychometric profiling tool, the highest percentage number of 34 of respondents were identified as "chaotic brain communication process" while about 26% of respondents were identified as "relational brain communication process." The less of 23% of respondents were identified as "linear brain communication process" while only 17% of respondents were identified as "intuitive brain communication process," respectively. Similar to female respondents, the highest number of 30% of respondents were identified as "chaotic brain communication process" while about 26% of respondents were identified as "relational brain communication process." The less of 22% of respondents were equally identified as "linear brain communication process" and "intuitive brain communication process," respectively (Table 1).

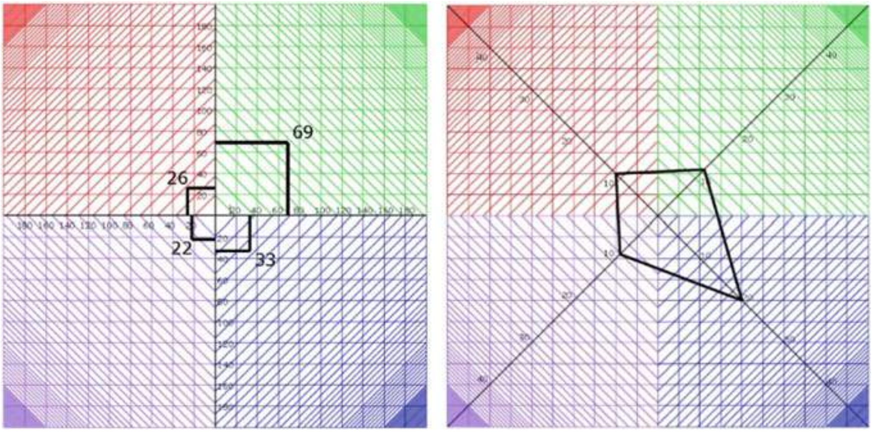


Fig. 3 (Left) Example of one respondent from all respondents involved in this study with chaotic brain communication process (green) identification and other brain communication process categories, e.g., linear (red), relational (purple), and intuitive (blue), respectively, as assessed by the web-based interactive neuro-psychometric profiling tool. Isolated number in each colored square means the indicator of dominant brain communication process; (Right) Example of one respondent from all respondents involved in this study with potentially miscommunication process of the relational (purple) brain communication process to other brain communication process categories, e.g., chaotic (green), linear (red), and intuitive (blue) brain communication process, respectively, as assessed by web-based interactive neuro-psychometric profiling tool

3.2 *Potentially Miscommunication Process Identification*

The observations of the potential miscommunication profiling of each of the above four identified brain communication process from all respondents in this study were also labeled to simplify their characteristics (Table 2; Figs. 3 (Right) and 4). As shown in Fig. 3, the number represents the highest and lowest concentration of each

Table 2 Percentage of the brain communication process which might be miscommunicated with other brain communication process categories assessed by web-based interactive neuro-psychometric profiling tool. Number represents percentage of respondents

Brain processing	Chaotic processing	Linear processing	Relational processing	Intuitive processing
Chaotic processing	18.00	27.74	24.38	18.07
Linear processing	13.33	11.99	9.29	16.99
Relational processing	55.13	44.05	55.47	53.57
Intuitive processing	13.53	16.22	10.86	11.36

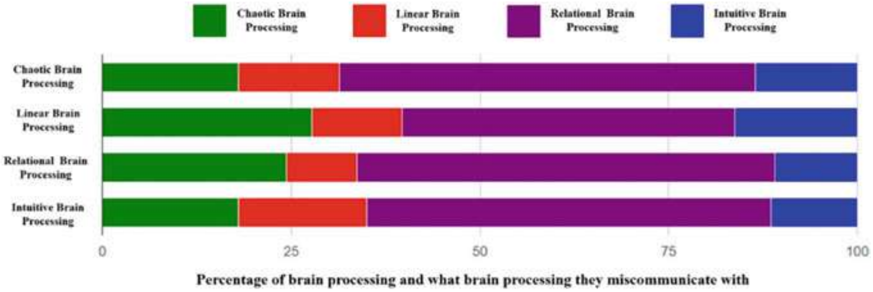


Fig. 4 Percentage of the brain communication process and what brain communication process they miscommunicate with and potentially miscommunication process of each brain communication process to other brain communication process categories, e.g., chaotic (green), linear (red), relational (purple), and intuitive (blue) brain communication process, respectively, assessed by web-based interactive neuro-psychometric profiling tool

brain communication process indicator that might have a difficult time in relating to other brain communication processors. That is, each brain communication indicator may be confusing their action process according to their own perspectives as opposed to utilizing their strengths to other people. The highest concentration is 55.47, implying that the relational brain communication process respondents might have a potential miscommunication process with relational brain communication process respondents. Then again, by demonstrating the most reduced convergence of 9.29, the linear brain communication respondents may have less potential miscommunication with relational brain communication respondents (see Table 2; Figs. 2 and 4).

3.3 Chaotic Brain Communication Process

Of the all respondents who were identified as chaotic brain communication process, these sixty-four percent chaotic brain communication process respondents might have a high primary miscommunication process with linear brain communication process respondents (27.74%), relational brain communication process respondents (24.38%), intuitive brain communication process respondents (18.07%), and chaotic brain communication process respondents (18.00%), respectively (see Table 2). It was also found that the highest concentration was shown in respondents with linear (27.74%) brain communication process followed by those respondents with relational (24.38%) and intuitive (18.07%) brain communication process, while the lowest concentration was shown in respondents with chaotic (18.00%) brain communication process, implying that respondents with chaotic brain communication process might have a most difficult time in relating to those respondents with the linear brain communication process and orientation followed by those respondents with the relational, intuitive, and chaotic brain communication process and orientation, respectively. Additionally, these respondents with chaotic brain communication process or orientation might be misinterpreting their action process based on their own views instead of leveraging on their strengths to those respondents with linear, relational, intuitive, and chaotic brain communication processes or orientation, respectively.

3.4 Linear Brain Communication Process

Of the all respondents who were identified as linear brain communication process, these forty-five percent of linear brain communication process respondents might have a high primary miscommunication process with intuitive brain communication process respondents (16.99%), chaotic brain communication process respondents (13.33%), linear brain communication process respondents (11.99%), and relational brain communication process respondents (9.29%), respectively (see Table 2). It was also found that the highest concentration was shown in respondents with intuitive (16.99%) brain communication process followed by those respondents with chaotic (13.33%) and linear (11.99%) brain communication process, while the lowest concentration was shown in respondents with relational (9.29%) brain communication process, implying that respondents with linear brain communication process might have most the difficult time in relating to those respondents with Intuitive brain communication process and orientation followed by those respondents with chaotic, linear, and relational brain communication process and orientation, respectively. Additionally, these respondents with linear brain communication processes or orientation might be misinterpreting their action process based on their own views instead of leveraging on their strengths to those respondents with intuitive, chaotic, linear, and relational brain communication process or orientation, respectively.

3.5 Relational Brain Communication Process

Of the all respondents who were identified as relational brain communication process, these thirty-two percent of relational brain communication process respondents might have a high primary miscommunication process with relational brain communication process respondents (55.47%), chaotic brain communication process respondents (55.13%), intuitive brain communication process respondents (53.57%), and linear brain communication process respondents (44.05%), respectively (see Table 2). It was also found that the highest concentration was shown in respondents with relational (55.47%) brain communication process followed by those respondents with chaotic (55.13%) and intuitive (53.57%) brain communication process, while the lowest concentration was shown in respondents with linear (44.05%) brain communication process, implying that respondents with relational brain communication process might have the most difficult time in relating to those respondents with relational brain communication process and orientation followed by those respondents with chaotic, intuitive, and linear brain communication process and orientation, respectively. Additionally, these respondents with a relational brain communication process or orientation might be misinterpreting their action process based on their own views instead of leveraging on their strengths to those participants with relational, chaotic, intuitive, and linear brain communication process or orientation, respectively.

3.6 Intuitive Brain Communication Process

Of the all respondents who were identified as intuitive brain communication process, these thirty-nine percent of intuitive brain communication process respondents might have a high primary miscommunication with linear brain communication process respondents (16.22%), chaotic brain communication process respondents (13.53%), intuitive brain communication process respondents (11.36%), and relational brain communication process respondents (10.86%), respectively (see Table 2). It was also found that the highest concentration was shown in respondents with linear (16.22%) brain communication process followed by those respondents with chaotic (13.53%) and intuitive (11.36%) brain communication process, while the lowest concentration was shown in respondents with relational (10.86%) brain communication process, implying that respondents with Intuitive brain communication process might have the most difficult time in relating to those respondents with linear brain communication process and orientation followed by those respondents with chaotic, intuitive, and relational brain communication process and orientation, respectively. Additionally, these respondents with intuitive brain communication processes or orientation might be misinterpreting their action process based on their own views instead of leveraging on their strengths to those respondents with linear, chaotic, intuitive, and relational brain communication processes or orientation, respectively.

4 Discussion

From all respondents who did the inventory and were assessed by the web-based interactive neuro-psychometric profiling tool, the highest respondents was chaotic (64%) brain communication process while the less of respondents was relational (32%) brain communication processes, respectively. The observations of the potential miscommunication process profiling of each identified brain communication process from all respondents in this study were also labeled to simplify their characteristics. The highest concentration is 55.47, implying that the relational brain communication process respondents might have a potential miscommunication process with relational brain communication process respondents. On the other hand, by showing the lowest concentration of 9.29, the linear communication process respondents might have less potential miscommunication process with relational communication process respondents. In light of web-based interactive brain communication process neuro-psychometric profiling, each brain communication processing may be confounding their action processes according to their own perspectives as opposed to utilizing on their strengths to other people.

As indicated by Carmazzi [4], the miscommunication cycle or misjudging alluded individuals who make us insane and need to turn out to be more adaptable. As an explanation of the miscommunication process, Carmazzi clarified that if we have a higher focus in the other brain communication process, at that point we may struggle to identify in relating to people with that color orientation. Furthermore, we may be misconstruing our action process dependent on our view as opposed to utilizing our strengths [2, 4]. As a universal mechanism for everyone who expresses his actions is described in the sociopsychological tradition, brain communication and processing studies can be incorporated into how to plan information, how to plan the information matters, how the information is received and processed, and the message effects for individuals [1]. The most popular used approaches are trends that influence people to act and interact with each other [1]. Different communication situations can be learned from the sociopsychological tradition of the biological theory developed since 1990. Morisan explains that many influences by situational or learning factors but are influenced by biological factors the one is the brain as a great human processor for people [1].

As in 2018, Carmazzi found a new method of recognizing the human brain's clarity based on the foundations of hereditary neurosciences [3]. In Carmazzi's investigation, he proposed the particular phrasing of ambiguity relief to refer to the human brain's processing that supports assessing data to relate new context to existing information, streamlining thoughts, settling on choices, and critical thinking into significant components [3, 13]. The noteworthy component was claimed to be a fundamental component of human correspondence whereby directly related to the action taking and to the accomplishment of clarity [3]. The noteworthy component was professed to be a basic component of human correspondence whereby straightforwardly identified with the activity taking and to the achievement of lucidity [3]. As referenced by Carmazzi, all decisions, advancements, idea generation, memory retention, communications, and understandings are separated through the brain's ambiguity relief process [5, 13]. The implications of recognizing and understanding the brain's ambiguity relief process will uphold improved correspondence, collaboration, cooperative energy, and how to move individuals energetically to activity [5]. The attention to the brain communication process gives us a way to expand our intelligent action capacity. It is also influential patterns of individual communication and cultivate the potential of individuals to career or work in the future. In addition, the brain communication process also affects behavior and communication patterns to assess a change or innovation and creativity [1]. In 2013, Chermahini and colleagues examined the connection between learning styles and the scholarly presentation of students who went to English class to learn English as a second language in Iran [14]. Based on the findings, it can be concluded that the effect of learning styles on foreign language learning varies for different countries or varies according to the language being learned per se [1, 13, 14]. In 2017, Wei and colleagues investigated the intercession impact of left-brain and right-brain dominance on the connection between learning styles and Japanese language academic achievement. The outcomes indicated that left-brain and right-brain dominance had a full intervention impact on the connection between learning styles and Japanese language academic achievement [7]. Wei and colleagues had done the expressive information which was created to give a superior comprehension of the mean and standard deviation of the variables. This previous study explored how well the two measures of learning styles and brain dominance influenced Japanese language academic results. This previous study demonstrated that the respondents' recognition toward the left-brain dominance and right-brain dominance was over the average. Wei and colleagues [7] proposed that an investigation on instructors' teaching procedures and student learning styles in teaching a foreign language is recommended as an approach to investigate the connection between the instructor's techniques and foreign language academic achievement so that the powerful educating and learning methodologies that consider both instructors and students' learning styles can be proposed. The thought of the procedures of multiple intelligences, learning styles, left-brain, and right-brain learning methodologies would improve the consequences of Japanese language academic achievement. Moreover, previous neuroscientific studies have already demonstrated that the two hemispheres work together to perform important tasks. These two hemispheres communicate through the corpus callosum. The left hemisphere specializes in picking out the sounds that form words and working out the syntax of the phrase, but it does not have a monopoly on language processing, whereas the right hemisphere is more sensitive to the emotional features of the language, tuning into the slow rhythms of speech that carry intonation and stress [7].

As to the use of the brain communication process to the workplace, Carmazzi represented a distinctive brain communication process in grouping the hereditary neuro-processing so that every individual encounter their condition and takes in and interprets data bringing about explicit methods of perceiving and assessing occasions and circumstances [10, 13, 15]. Carmazzi classified four ambiguity relief processes including (a) the chaotic processing named as green brain, (b) linear processing named as red brain, (c) the relational processing named as purple brain, and (d) the intuitive processing named as blue brain, in view of hereditary [10, 13, 15]. These four color healthy processes characterization disentangle them through the so-called colored brain model [13, 15].

The most recent studies revealed impact of applying web-based interactive neuropsychometry in other aspects. One recent study presented a mixed learning context that makes a network of training. This group of network training presented a mix of face-to-face facilitated learning, e-learning, and self-study. Both in-class and online linguistics exercises were utilized to investigate the adequacy of performing these linguistic exercises in homogeneous and heterogeneous groups. The incorporation of target learners' brain color into their multiple intelligences depended on utilizing two inventories which distinguished learners' brain color and multiple intelligences. The two inventories were colored brain communication inventory (CBCI) and multiple intelligences inventory (MII) were administered to a group of learners at the Arab Open University (AOU) during their English language context instructional exercises. The findings showed that it helped the scientist discover the learners' preferred methods for learning and evaluation. The consequences of these two inventories (CBCI and MII) were statistically analyzed with the students' performance in dissecting and understanding the given semantic exercises. The results revealed particular points of working in heterogeneous groups with learners who possessed a variety of color brain and multiple intelligences [16]. The most recent investigation recognized the connection between brain processing and correspondence of financial officers in Thailand. In this recent investigation, the fifty participants were evaluated by utilizing the colored brain communication inventory (CBCI). This investigation was the first investigation to the impacts of brain processing and correspondence on the leadership and workplace by utilizing the colored brain communication inventory (CBCI) in Thailand. By employing the theoretical and theories of the colored brain model, any awareness by participants joined in the study may help their colleagues to receive a participation mentality as opposed to attempting to transform others their own processes which making dissatisfaction on both sides [17].

Furthermore, Carmazzi emphasized that with different brain processing, we might have a direct impact on our communication root causes with others such as miscommunications and misunderstandings. Misunderstanding frequently emerges from a misconception of an individual's brain communication processes toward objectives and prevents getting everybody from communicating in a similar language. These interchanges may influence our working relationship with others particularly our work performance [5]. In this way, the web-based interactive neuro-psychometric profiling instrument was intended to help the decrease of barriers to make simpler, quicker, and more viable communication over any relationship [5, 13]. By understanding our brain communication process evaluated by the web-based interactive neuro-psychometric profiling and our distinctive communication characteristics, we will have the option to create suitable individual communication techniques to gain by our strengths, just as utilizing the strengths of others to conquer our weaknesses [2, 4, 5, 8].

5 Conclusion

This study explored the brain communication process by using the web-based interactive neuro-psychometric profiling. Based on observations and investigated research, the brain's clarity seeking process. Ambiguity relief processes included chaotic, intuitive, linear, and relational ambiguity relief processes. This identifies the impacts of brain processing and communication in the workplace. The potential miscommunication process profiling of each identified brain communication process in this study revealed that the relational brain communication process respondents might have the most potential miscommunication with relational brain communication process respondents, whereas the linear brain communication process respondents might have less potential miscommunication process with relational brain communication process respondents. Therefore, the output of this study is a type of brain communication process that shows the tendency of nature, attitudes, and individual potential.

Acknowledgements This work was supported, in part, by Mae Fah Luang University, with resources from the School of Anti-Aging and Regenerative Medicine, Mae Fah Luang University, Thailand. We thank all of the participants who involved in this study as the main data provider.

References

1. Iswanti, M.E., Triadnyana, N.I., Rahmanto, A.: Mapping brain color to communications style: study at FIFGROUP region of DIY. In: Artaria, M.D., Wahyudi, S.I.I, Wadipalapa, R.P. (eds.) Proceedings of the International Post-Graduate Conference on Media and Communication, pp. 34–39. Surabaya, Indonesia (2017)
2. Carmazzi, F.A.: The Colored Brain Communication Field Manual: Practical Applications of Directive Communication Psychology and the Colored Brain to Work, Leadership, Business and Relationship. Directive Communication International, Bali (1999)
3. Carmazzi, F.A.: New research identifies the brain's clarity getting process has foundations in genetic neuroscience. EC Neurol. **10**(2), 20–21 (2018)
4. Carmazzi, F.A.: Observation and genetic foundations of the brain's clarity achieving "ambiguity relief" processes. Theranostics Brain, Spine Neural Disord. **2**(3), 26–34 (2017)
5. Carmazzi, F.A.: The colored brain communication inventory. (2008) [Online]. Available: www.directive.communication
6. Cloninger, C.R., Svrakic, D.M., Przybeck, T.R.: A psychobiological model of temperament and character. Arch. Gen. Psychiatry **50**(12), 975–990 (1993)
7. Wei, H.-S., Sulaiman, T., Baki, B., Roslan, S.: The relationship between brain dominance and Japanese language academic achievement. Int. Res. J. Educ. Sci. **1**(2), 48–51 (2017)
8. Chin, F.C.J., Ooi, M.L., Yip, M.W.: The effects of colored brain communication and brain processing interpretation on the academic performance of students: a literature review. Int. J. Inf. Educ. Technol. **6**(12), 945–948 (2016)
9. Cook, T.D., Campbell, D.T.: Quasi-Experimentation: Design and Analysis Issues in Field Settings. Houghton Mifflin, Boston, MA (1979)
10. Herrmann, N.: The Whole Brain Business Book: Harnessing the Power of the Whole Brain Organization and the Whole Brain Individual. McGraw-Hill Professional Publisher, New York City (1996)
11. Clark, K.L., Noudoost, B.: The role of prefrontal catecholamines in attention and working memory. Front. Neural Circ. **8**, Article ID. 33 (2014)
12. Horak, E., Toit, J.W.D.: A study of the thinking styles and academic performance of civil engineering students. J. S. Afr. Inst. Civ. Eng. **44**(3), 18–24 (2002)
13. Seagal, S.: Human Dynamics: A New Framework for Understanding People and Realizing the Potential in Our Organizations. Pegasus Communications, Cambridge (1997)
14. Chermahini, S.A., Ghanbari, A., Talab, M.G.: Learning styles and academic performance of students in English as a second-language class in Iran. Bul. J. Sci. Educ. Policy **7**(2), 322–333 (2013)
15. Ebner, N.C., Kamin, H., Diaz, V., Cohen, R.A., MacDonald, K.: Hormones as "difference makers" in cognitive and socioemotional aging processes. Front. Psychol. **5**, Article ID. 1595 (2015)

16. Milad, M.: Incorporating brain colour into the multiple intelligences to create a blended learning context: homogeneous and heterogeneous groups. IAFOR J. Lang. Learn. **4**(1), 41–55 (2018)
17. Sittiprapaporn, P., Carmazzi, A.F.: Classification of brain processing indicators in financial officers. Asian J. Med. Sci. **10**(5), 55–62 (2019)

Seventh International Symposium on Computer Vision and the Internet (VisionNet'20)

Deep Visual Attention Based Transfer Clustering

Akshaykumar Gunari, Shashidhar Veerappa Kudari, Sukanya Nadagadalli, Keerthi Goudnaik, Ramesh Ashok Tabib, Uma Mudenagudi, and Adarsh Jamadandi

Abstract In this paper, we propose a methodology to improvise the technique of Deep Transfer Clustering (DTC) when applied to the less variant data distribution. Clustering can be considered as the most important unsupervised learning problem. A simple definition of clustering can be stated as "the process of organizing objects into groups, whose members are similar in some way". Image clustering is a crucial but challenging task in the domain machine learning and computer vision. We have discussed the clustering of the data collection where the data is less variant. We have discussed the improvement by using attention-based classifiers rather than regular classifiers as the initial feature extractors in the Deep Transfer Clustering. We have enforced the model to learn only the required region of interest in the images to get the differentiable and robust features that do not take into account the background. This paper is the improvement of the existing Deep Transfer clustering for less variant data distribution.

A. Gunari (✉) · S. V. Kudari · S. Nadagadalli · K. Goudnaik · R. A. Tabib · U. Mudenagudi · A. Jamadandi
KLE Technological University, Hubballi, India
e-mail: akshaygunari@gmail.com
URL: http://kletech.ac.in/

S. V. Kudari
e-mail: shashidharvk100@gmail.com

S. Nadagadalli
e-mail: sukanyanadagadalli@gmail.com

K. Goudnaik
e-mail: keerthigoudnaik@gmail.com

R. A. Tabib
e-mail: ramesh_t@kletech.ac.in

U. Mudenagudi
e-mail: uma@kletech.ac.in

A. Jamadandi
e-mail: adarsh.cto@tweaklabsinc.com

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_29

Keywords Deep embedded clustering · Deep transfer clustering · Auto encoders · Manual attention · Temporal ensembling · Visual explanations of CNNs

1 Introduction

Clustering is central to many data-driven application domains and has been studied extensively in terms of distance functions and grouping algorithms. Relatively little work has focused on learning representations for clustering. One such methodology came up with Deep Embedded Clustering (DEC) where it simultaneously learns feature representations and cluster assignments using deep neural networks. DEC learns a mapping from the data space to a lower-dimensional feature space in which it iteratively optimizes a clustering objective. In this process every time a new set of images belonging to a class arrives, the autoencoder should be trained again including the new set of arrived images.

A new technique Deep Transfer Clustering [3] uses prior knowledge of related but different classes to reduce the ambiguity of clustering. The algorithm is improvised by introducing a representation bottleneck, temporal ensembling, and consistency. This also transfers knowledge from the known classes, using them as probes to diagnose different choices for the number of classes in the unlabelled subset. The retraining process to extract the features from a set of images belonging to a new class becomes redundant. But, this methodology results can be improvised for the image collection where the distribution is less variant.

A technique where we can visually validate what the model is looking at, verifying that it is indeed looking at the correct patterns in the image and activating around those patterns. If the model is not activating around the proper patterns/objects in the image, then we can revisit the process of training for required feature extraction. Grad-CAM is a tool [5], which is used to visualize the flow of gradients through a neural network so that we get to know what the model is actually learning. When the data distribution is less variant, the features to be extracted for clustering should be differentiable which can enhances the performance of clustering.

With this regard of the model to be able to learn those differentiable features and to concentrate on the required/instructed region of interest in the images, we propose this methodology to train the model for feature extraction which serves the purpose. Our method is an add—on to the existing Deep Transfer Clustering method [3] which helps the feature extractor to learn more robust and differentiable features rather than concentrating on background common features.

When we use GradCAM, GradCAM++ [2, 6] for the visualization of what the model is learning in case of classifiers, we observe that the model is trying to learn the only a part of the image in the dataset. But when we are using these classifiers as the initial feature extractors for the Deep Transfer Clustering where the inter-class variance is very less, the clustering performance decreases. So to overcome this, we introduce manual attention mechanism while training the classifiers for the labelled data which not only makes the feature vectors more robust to the region of

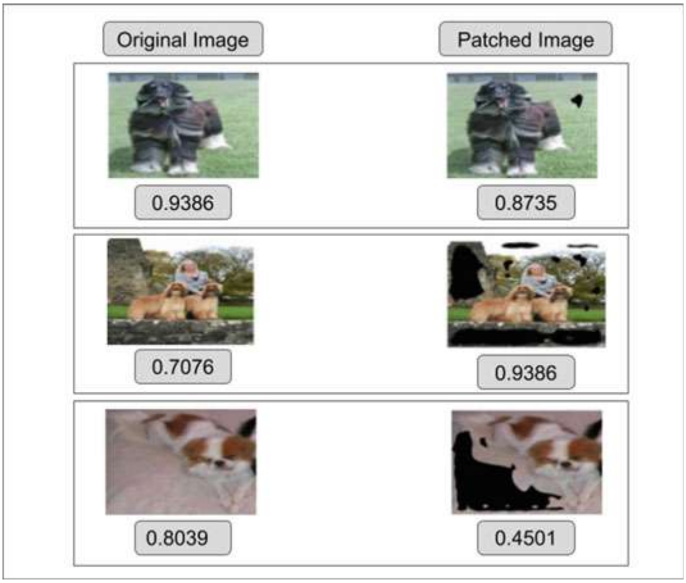


Fig. 1 Uncertain behaviour in the prediction accuracy of the classifier by adding a patch in certain background region of the images

interest, but also differentiable features are paid more attention. We introduce the manual attention in the training process of classifier by modifying the images, i.e. the unwanted background part of the images are discarded. GrabCut [4], is one such algorithm which is used for the removal of the unwanted background part of the image.

Features cannot be extracted using regular classifiers which are trained on the raw images, in clustering process where the data is less variant, because providing the raw images to the classifier does not imply what the model should focus to learn. This results in uncertain behaviour of the model when a small region is patched over the image as can be observed in Fig. 1.

Visualizing the model using some visualizing algorithms such as GradCAM, GradCAM++, it can be observed that the model concentrates on the part of the object rather on the important aspects of the image which is important in case of intra-class clustering (Fig. 2).

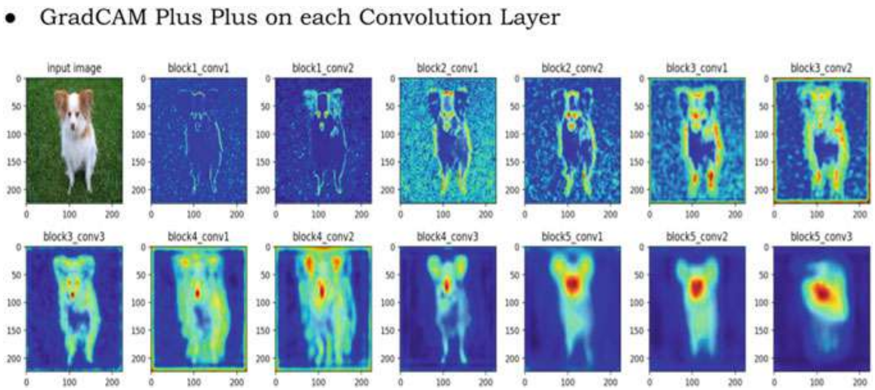


Fig. 2 Visualization of the each convolution layer of VGG16 classifier using GradCAM++

2 Related Work

2.1 Deep Adaptive Clustering

Deep Adaptive Clustering (DAC) [1] recasts the clustering problem into a binary pairwise-classification framework to judge whether pairs of images belong to the same clusters. In DAC, the similarities are calculated as the cosine distance between label features of images which are generated by a deep convolutional network (ConvNet). By introducing a constraint into DAC, the learned label features tend to be one-hot vectors that can be utilized for clustering images. The main challenge is that the ground-truth similarities are unknown in image clustering. This issue is handled by presenting an alternating iterative Adaptive Learning Algorithm where each iteration alternately selects labeled samples and trains the ConvNet.

2.2 Deep Embedded Clustering

Deep Embedded Clustering (DEC) [7] is the algorithm that clusters a set of data points in a jointly optimized feature space. DEC works by iteratively optimizing a KL divergence based clustering objective with a self-training target distribution. This method can be viewed as an unsupervised extension of semi-supervised self-training. This framework provides a way to learn a representation specialized for clustering without ground truth cluster membership labels. DEC offers improved performance as well as robustness for hyperparameter settings, which is particularly important in unsupervised tasks since cross-validation is not possible.

2.3 Deep Transfer Clustering

Deep Transfer Clustering [3] which is the improvisation of the DEC has introduced a simple and effective approach for novel visual category discovery in unlabelled data. This method can simultaneously learn data representation and cluster the unlabelled data of novel visual categories while leveraging the knowledge of related categories in labeled data. This method also proposes a novel method to reliably estimate the number of categories in unlabelled data by transferring cluster prior knowledge using labeled probe data. This method overcomes the additional training of the autoencoder with old and new classes. This method uses a classifier as a feature extractor instead of the autoencoders. Loss functions of Deep Embedded Clustering are imposed with certain constraints and temporal ensembling methods to get better results.

All the state of the art methods shows the results on the datasets (MNIST, Omniglot, CIFAR10, CIFAR100) where the class variance is very high. In this paper we show the results of our methodology on the dog dataset (a subset of Imagenet dataset) where the data distribution is less variant, experiments show that our improvements perform better than Deep Transfer Clustering (DTC).

3 Deep Visual Attention Based Transfer Clustering

Our methodology is the improvement of Deep Transfer Clustering (DTC) for intra-class clustering. A major step in DTC, i.e., training a classifier that is used as a feature extractor is improvised to get more robust and differentiable features. As we can see from the Observations section that the classifiers are learning some background instead of focusing more on important aspects of the image. This might be good for the classification where training and testing classes are the same, but in the case of the clustering of the intra-classes, this might affect the performance of clustering. So instead of providing the raw images for the training of the classifier, images where the background part is removed and only the part of the image which can help the model to focus on the differentiable features is provided as shown in Fig. 3.

So, to remove the background from the image, a tool using the GrabCut Algorithm, which can instruct the model to focus on the region of interest for data preprocessing is designed as shown in Fig. 4. This tool helps to get the ROI (Region of Interest) in an image. GrabCut is an image segmentation method based on graph cuts. It takes a user-specified bounding box around the object to be segmented, the algorithm estimates the color distribution of the target object and that of the background using a Gaussian mixture model.

DTC algorithm after the improvement is as follows:

- We train the classifier with the background-subtracted images. Background subtraction is done using the tool which we have created using the grabcut algorithm. Now we have the feature extractor for the upcoming new classes. Though the features may be good for new classes it might not be the best.

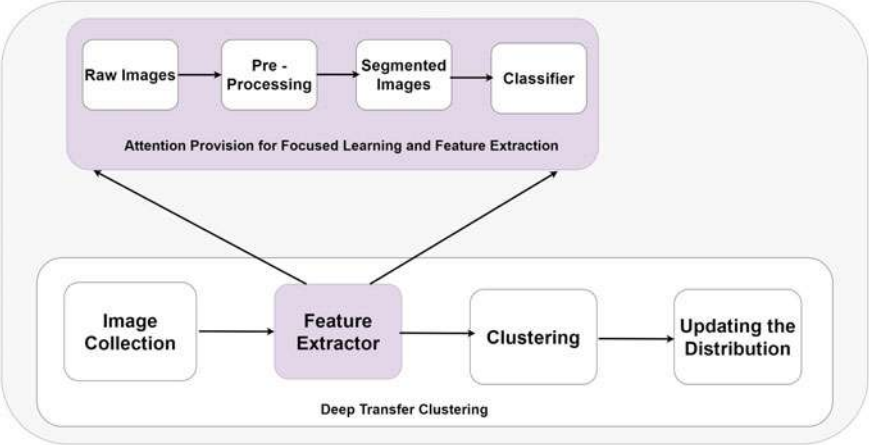


Fig. 3 Deep visual attention based transfer clustering

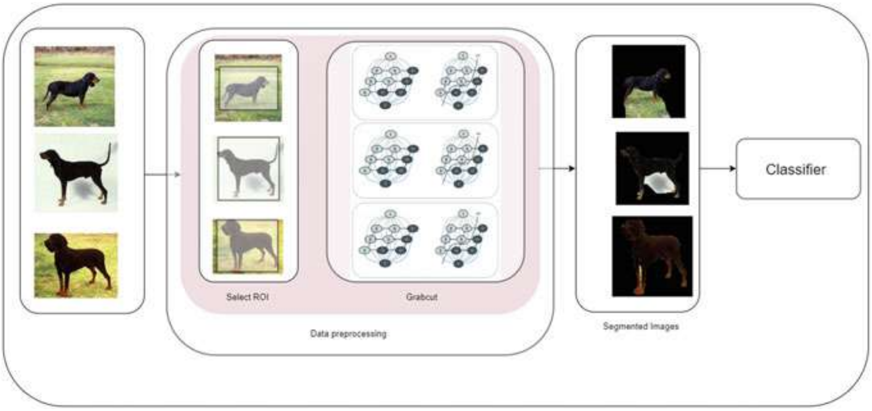


Fig. 4 Our method of preprocessing the data before training the classifier for the initial feature extraction

- Extract the features for the new unlabelled images using the previously trained classifier model, reduce the dimensions to the K dimensions using the PCA and initialize the cluster centers U using the K-means and incorporate the PCA as the last linear layer in the classifier model, where $U = \{\mu_k, k = 1, \ldots, K\}$ represents the cluster centers, f_θ represents the feature extractor.
- Train the θ of the feature extractor f_θ and cluster centers U using q as the target distributions and update the target distribution q. Here the loss function is combination of DEC loss and the consistency constraints loss which is as follows.
  let $p(k|i)$ be the probability of assigning data point $i \in \{1, \ldots, N\}$ to cluster $k \in \{1, \ldots, K\}$.

$$p(k|i) \propto (1 + \frac{\parallel z_i + \mu_k \parallel^2}{\alpha})^{-\frac{\alpha+1}{2}} \tag{1}$$

- Further assuming that data indices are sampled uniformly (i.e. $p(i) = 1/N$), we can write the joint distribution as $p(i, k) = p(k|i)/N$. DEC defines its objective as KL divergence loss between the soft assignments q and auxiliary distribution p as follows

$$E(q) = \mathrm{KL}(q \parallel p) = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} q(k|i) \log \frac{q(k|i)}{p(k|i)} \tag{2}$$

- We minimize the KL divergence between the auxiliary distribution and target distribution to attain the good solution. We have the formula to calculate the auxiliary distribution p. Target distribution q is calculated as follows:

$$q(k|i) \propto p(k|i) \cdot p(i|k) \tag{3}$$

- In this manner the assignment of image i to cluster k is reinforced when the current distribution assigns a high probability of going from i to k as well as of going from k to i. The latter has an equalization effect as the probability of sampling data point i in cluster k is high only if the cluster is not too large. Using Bayes rule for $p(k|i)$ as follows:

$$q(k|i) \propto \frac{p(k|i)^2}{\sum_{i=1}^{N} p(k|i)} \tag{4}$$

- DTC uses the above loss and also the additional function in the loss i.e. consistency constrain. Consistency constrains have shown the effective results in case of semi-supervised learning. A consistency constraint can be incorporated by enforcing the predictions of a data sample and its transformed counterpart (which can be obtained by applying data transformation such as random cropping and horizontal flipping on the original data sample). Hence the final loss function is defined as

$$E(q) = \mathrm{KL}(q \parallel p) = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} q(k|i) \log \frac{q(k|i)}{p(k|i)} + \omega(t) \frac{1}{\mathrm{NK}} \sum_{i=1}^{N} \sum_{k=1} K \parallel p(k|i) - p'(k|i) \parallel \tag{5}$$

where $p'(k|i)$ is either the prediction of the transformed sample or the temporal ensemble prediction $p^{\sim t}(k|i)$, and $\omega(t)$ is a ramp-up function as used to gradually increase the weight of the consistency constraint from 0 to 1.

- Predict the probability of assignment $p(k|i)$ of image $i \in \{1, \ldots, N\}$ to the cluster $k \in \{1, \ldots, K\}$ using the equation and get the index of the cluster to each image with the highest $p(k|i)$

DTC improves DEC smoothing of the cluster assignment via temporal ensembling i.e. the clustering model's p is computed at different epochs are aggregated by maintaining an exponential moving average (EMA) of the previous distributions.

$$P^t(k|i) = \beta \cdot p^{t-1}(k|i) + (1-\beta) \cdot p^t(k|i). \tag{6}$$

where β is a momentum term controlling how far the ensemble reaches into training history, and t indicates the time step. To correct the zero initialization of the EMA , P^t is rescaled to obtain the smoothed model distribution i.e.

$$p^{\sim t}(k|i) = \frac{1}{1-\beta^t} \cdot P^t(k|i) \tag{7}$$

Equations (1)–(7) were taken from [3].

DTC has improved the DEC algorithm by using Temporal ensembling and consistency constraints, we show that using our method of training the initial feature extractor by using the manual attention over the training the images gives better results than the regular classifiers for the data where there is less variance between the classes, we have changed the initial training process of the DTC, the remaining algorithm remains same as that of DTC.

Figure 5 is the complete algorithm of the Deep Transfer Clustering (DTC), which includes two major steps training of classifier and the fine tuning of the feature extractor model with respect to the cluster assignment loss and other constraints, we have changed the first step of the DTC algorithm which is best suitable for the Intra-class clustering.

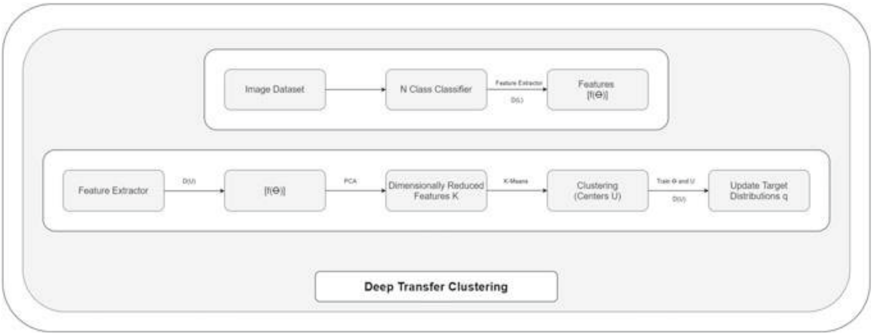


Fig. 5 Deep transfer clustering

4 Results

The results were generated for the dog dataset, the subset of the imagenet dataset. This dataset was chosen because it has very high similarities between the classes and also in many of the dog classes dogs face of one class is somewhat similar to other classes, so we need our classifier to concentrate on the whole dog in the image rather than just concentrating on the face of the dog which is always true in case of the classifier trained using raw images. Our improvisation was compared with DTC.

DTC has three different algorithms namely:

- DTC-PI: Model trained using DEC loss with consistency constraint between predictions of a sample and its transformed counterpart.

Pi model		
	DTC	Our proposed model
Dataset	Dog	Dog
Accuracy	0.7784	0.7791
NMI	0.7369	0.7378
ARI	0.6458	0.6466

The above results were generated for the DTC-PI model, where loss function has the DEC loss and the consistency constraint, our results are slightly better than the original DTC model's accuracy.

- DTC-Baseline: Model trained using the just DEC loss.

Baseline model		
	DTC	Our proposed model
Dataset	Dog	Dog
Accuracy	0.7508	0.7811
NMI	0.7172	0.7481
ARI	0.6442	0.6668

The above results were generated for the DTC-Baseline model, where the loss function is just the DEC loss, our results are 3% higher in case of accuracy and NMI and 4% higher in the case of ARI.

- DTC-TEP: Model trained using DEC loss with consistency constraint between the current prediction and temporal ensemble prediction of each sample.

TEP model		
	DTC	Our proposed model
Dataset	Dog	Dog
Accuracy	0.7534	0.7779
NMI	0.7211	0.7455
ARI	0.6285	0.6615

The above results were generated for the DTC-TEP model, where loss function consists of DEC loss and consistency constraint, the auxiliary distribution $p(k|i)$ is calculated using the temporal ensembling method. Our results are 2.5% higher in the case of accuracy and NMI 4% higher in the case of ARI.

5 Conclusion

In this paper, various facts could be concluded, some of them including that the traditional clustering techniques of using pre-trained models for image clustering will not lead to better results for all the kinds of data, especially for those classes where the similarity between the two classes is very high. In such cases where the dataset contains the classes in which it is very difficult to differentiate between the classes. It is very difficult to extract the differentiable features from the dataset.

A simple and effective approach for the categorization of unlabelled data is introduced, by considering it as a deep transfer clustering problem. The method proposed can simultaneously learn to extract the features that can differentiate between two classes, learn a data representation and cluster the unlabelled data of novel visual categories, while leveraging the knowledge of related categories in labeled data. The main focus was to cluster the image collection where the similarity between the two classes is very high. This we have achieved by plugging in a module that makes the network concentrate on the instructed region of interest rather than a complete image.

References

1. Chang, J., Wang, L., Meng, G., Xiang, S., Pan, C.: Deep adaptive image clustering. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5879–5887 (2017)
2. Chattopadhay, A., Sarkar, A., Howlader, P., Balasubramanian, V.N.: Grad-cam++: generalized gradient-based visual explanations for deep convolutional networks. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Mar 2018
3. Han, K., Vedaldi, A., Zisserman, A.: Learning to discover novel visual categories via deep transfer clustering (2019)
4. Rother, C., Kolmogorov, V., Blake, A.: "GrabCut" interactive foreground extraction using iterated graph cuts. ACM Trans. Graphics (TOG) **23**(3), 309–314 (2004)
5. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: visual explanations from deep networks via gradient-based localization. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 618–626 (2017)
6. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: visual explanations from deep networks via gradient-based localization. Int. J. Comput. Vis. **128**(2), 336–359 (2019)
7. Xie, J., Girshick, R.B., Farhadi, A.: Unsupervised deep embedding for clustering analysis. CoRR, abs/1511.06335 (2015)

Video Retrieval Using Residual Networks

U. Tejaswi Nayak, C. Sujatha, Tanmayi V. Kamat, and Padmashree Desai

Abstract With the growing size of data across various different forms in today's world, a lot of meaningful information needs to be extracted from huge amount of data. Specially the multimedia content on web is increasing rapidly; thus, the demand for searching and retrieval of the required multimedia data is also increasing. Hence, there is a need for a faster retrieval of required data for different queries such as image, video, audio and text. In this paper, we propose a video retrieval framework using residual networks (ResNet-34) based on the query image or video clip and retrieve the relevant or similar videos from the video dataset. The ResNet-34 with locality sensitive hashing algorithm provides a faster retrieval of the relevant or similar videos from the dataset. The retrieval efficiency is improved from quadratic to logarithmic efficiency class. We demonstrate the proposed method for nine different categories of you tube videos and obtain an overall precision rate of 84% which is comparable with the state of the art.

Keywords Content-based video retrieval (CBVR) · ResNet-34 · Locality sensitive hashing

1 Introduction

There is a growing demand for analyzing the contents of multimedia. So, there is a need for extracting rich contents from a set of a videos in a database. In most of the scenarios, the users will have a task to find out a set of videos that are relevant to the interest of the users. Hence, there has been an active and a growing research in the area of content-based video retrieval (CBVR). The applications which are based on content-based video retrieval will be useful only when it is possible to retrieve the relevant videos (of user's interest) from a large set of videos in a video database efficiently and as quickly as possible.

U. Tejaswi Nayak · C. Sujatha (✉) · T. V. Kamat · P. Desai
School of Computer Science and Engineering, KLE Technological University, Hubballi, Karnataka, India
e-mail: sujata_c@kletech.ac.in

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_30

There are different ways for querying the content-based video retrieval system. However, in most of the scenarios, the text-based search will not help in retrieving the required document. The query by image or the query by video clip is highly desirable here as the multimedia data has increased drastically, which makes text-based search very challenging. It is necessary to have the input either in the form of an image or a video clip from the user so that it can retrieve the relevant videos efficiently and quickly.

In literature, most of the works have considered low-level visual features such as color, texture, motion and shape [1, 2] while few other works have focused on detecting specific objects (high-level semantic features) for video retrieval. Authors in [3, 4] segment the video from the dataset to identify the key frames using low-level visual features such as color, shape, motion [5], texture and multimodal features [6]. Authors in [7] use the high-level feature where they detect the flowers as an object by suppressing the background of flowers and then identify important features of a flower to retrieve the relevant videos. The authors in [8] have presented a text-based video retrieval and video search where they have extracted textual metadata through video optical character recognition (OCR) on key frames and automatic speech recognition (ASR) on audio track. Authors in [9] use spatial and temporal features, and in [10], authors analyze the activities and interactions in surveillance videos for the retrieval. In [11], the authors have focused on retrieving specific video objects from the videos of a dataset based on salient object detection using histogram of Fourier coefficients (HFC) along with extreme learning machine. In [12], the authors used convolutional neural networks (CNN) and bag of visual words (BoVW) for designing an efficient approach for large scale video retrieval by query as an image.

Authors in [13] present a semantic video retrieval method where specifically focus on the retrieval of videos involving key persons in certain scenarios of interest. Authors use convolutional and recurrent neural networks for identifying the person and caption generation to retrieve the relevant videos by query as a text.

In literature, most of the works have considered low-level visual features such as color, texture, motion and shape while few other works have focused on detecting specific objects (high-level semantic features) for video retrieval. Also, most of the works have considered either the query as a text or image. Toward this, our main contributions are:

- Segment and identify the key frames from the video dataset.
- Build a learning model and match the query frame with the identified key frames.
- Retrieve the relevant videos from the dataset for the given query image or video clip based on the matched key frame.
- Demonstrate the proposed method on nine categories of dataset and compare with state-of-the-art methods.

In Sect. 2, we give the overview of the proposed method. In Sect. 3, we demonstrate and discuss the results. Finally, we conclude in Sect. 4.

2 Proposed Framework for Video Retrieval Using Residual Network

We propose a video retrieval framework using residual networks (ResNet-34) based on the query image or video clip and retrieve the relevant or similar videos from the video dataset. The overview of proposed video retrieval method is as shown in Fig. 1, which includes video segmentation, key frame extraction, residual network and locality sensitive hashing for video retrieval. In what follows we present the details of these processes.

2.1 Video Segmentation and Extraction of Key Frames

In the process of video retrieval, the initial step is the segmentation of the video from the dataset and also on the query if it is a video clip. Initially, we segment the videos into different segments based on the scene detection. We use the content-aware scene detection approach for segmenting the video. The frames are represented in the HSV space. If the difference between two successive frames with respect to the HSV values is more than the threshold (here the threshold is set to 30), then there will be a division between the two frames and both the frames are considered to be in different scenes; otherwise, both the frames will be considered under a single scene. Prior to the extraction of key frames from a video, we segment a given video into its individual scenes. The segmentation will occur at the point where there is a scene change in the video. Scene refers to a part of the video that is taken in one camera position at one go and scene change refers to the change of a scene or in the position of the camera. Once a given video is segmented, one key frame is extracted from every scene of a video so that a given scene can be represented by one single frame. In this way, all the scenes of a given video are represented by the set of key frames extracted from that video.

The frame-wise comparison between the query video segments and the segments of video from the dataset is time consuming due to excessive amount of computation. Hence, we propose perform key frame-wise for the comparison. A set of key frames are extracted from a given video, it refers to a set of images that provides an accurate representation of a given video, i.e., in other words, a single video can be represented in a set of a few key frames. The extraction of key frames from a video is a very important step in the process of video retrieval. The extracted key frames will assist in efficient and accurate retrieval of relevant videos. Hence, we intend to perform a key-frame comparison between the query video and the videos of the data set with the help of a deep learning method. Finally, a set of relevant videos are retrieved from the videos of the data set based on the similar key frames between the query and the videos of the data set.

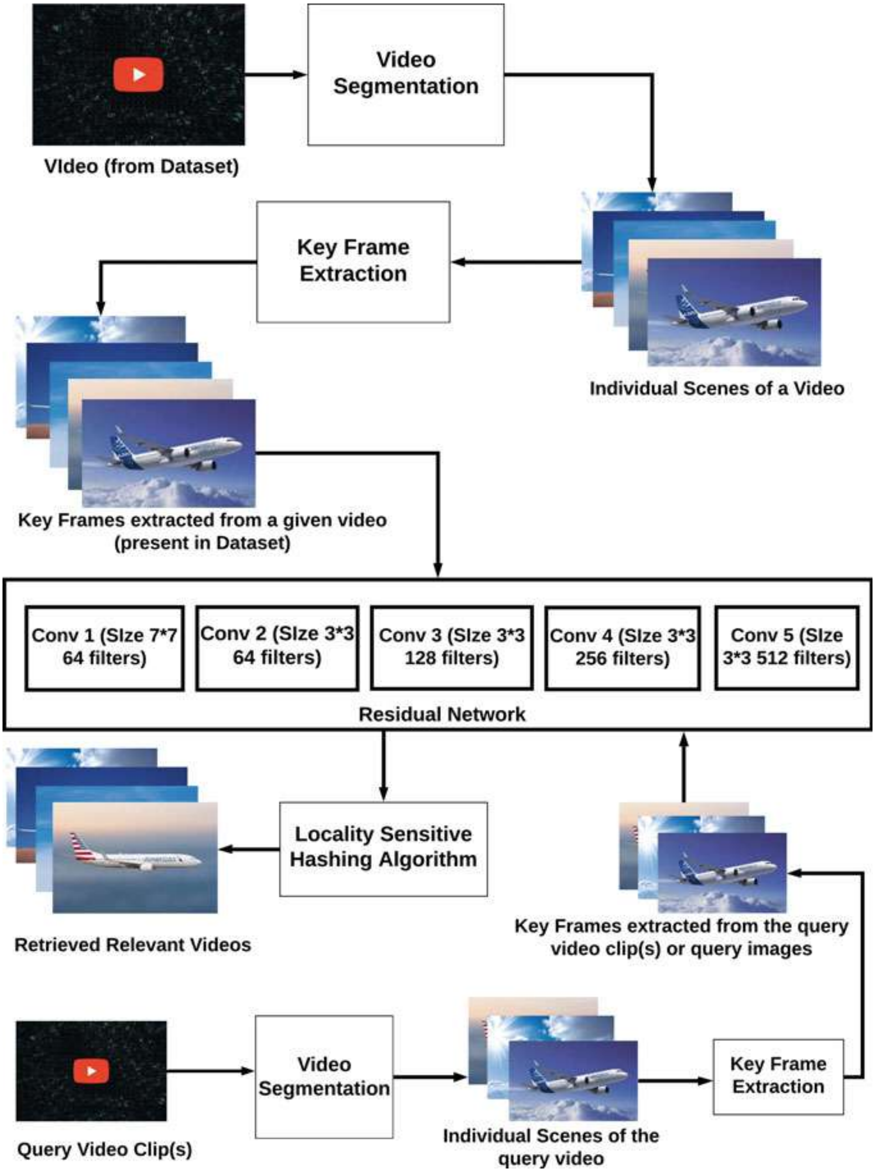

VIdeo (from Dataset)
Video Segmentation
Individual Scenes of a Video
Key Frame Extraction
Key Frames extracted from a given video (present in Dataset)
Conv 1 (SIze 7*7 64 filters)
Conv 2 (SIze 3*3 64 filters)
Conv 3 (SIze 3*3 128 filters)
Conv 4 (SIze 3*3 256 filters)
Conv 5 (SIze 3*3 512 filters)
Residual Network
Locality Sensitive Hashing Algorithm
Retrieved Relevant Videos
Key Frames extracted from the query video clip(s) or query images
Query Video Clip(s)
Video Segmentation
Individual Scenes of the query video
Key Frame Extraction

Fig. 1 Framework of video retrieval using residual network

2.2 Residual Network

Residual Network architecture is used to create representation for every key frame known as image embedding. The last two fully connected layers will generate the image embeddings for every key frame (both for the videos of the dataset and the query video clip). Image embeddings are nothing but a representation of an image in a 512 dimensional space. The embeddings are represented through data points. Each of the fully connected layers have 512 neurons in them, so the embeddings are generated in a 512-dimensional space. During the generation of the embeddings by the fully connected layers, all the others layers of the architecture are kept frozen. The image embeddings once generated by fully connected layers are extracted and converted to the form of multidimensional array and are saved in that form. Finally, a dictionary is created and stored where the path of the key frame (either of dataset or query) is stored as "key" and its corresponding embedding (in the form of multidimensional array) is stored as 'value'.

2.3 Locality Sensitive Hashing

LSH algorithm is an approximate nearest neighbor algorithm. This algorithm is used to reduce the time complexity from quadratic to logarithmic. This algorithm generates a hash value for every image embedding. It takes into consideration, the spatiality of the data in image embedding. As a result of this, similar key frames (extracted from either the same video or from similar videos) will have a higher chance of receiving similar hash values. The steps included in this algorithm are as follows:

- Generate K random hyperplanes in the embedding dimension (i.e., in the 512 dimensional space).
- A particular embedding is checked if it is above or below the hyperplane and is assigned a value 1 if it is above or 0 if it is below.
- The above step is repeated for each of the K hyperplanes to arrive at the hash value. The hash value is of size K.

The above three steps are repeated for "L" times. "L" here refers to the number of hash tables. Here, in our project, the value of K is taken as ten and the value of L is taken as 5. Once the key frames are extracted for a given video clip, embeddings are generated and are extracted for the key frames (of the input). For each embedding, an appropriate hash value is generated and an entry is added into the dictionary which contains the location of the image as key and its corresponding embedding as value. The LSH algorithm is used to compare the distance between the embeddings of the key frames from the video dataset and the embeddings of the input key frames.

2.4 Why ResNet-34 and Locality Sensitive Hashing?

Residual networks solve one of the well-known problems in deep learning, the problem of “vanishing gradients.” In deeper neural networks, due to the presence of activation functions along with large number of layers, the value of the gradients approaches 0, thus making it hard for the neural network to train as the weights never gets updated and hence, no learning takes place. In residual networks, the gradients flow directly through the skip connections from the concluding layers to the initial filters. Due to this, residual networks can achieve an exceptional performance even after training hundreds or thousands of layers.

The advantage of locality sensitive hashing algorithm is that instead of doing similarity search over the whole video dataset, a similarity search is performed only with a subset of key frames which shares the same hash value with that of the input key frames. Based on the key frames of the video dataset that are similar to the input key frames, the corresponding videos from the dataset are retrieved. Here, the videos retrieved will be similar or relevant to the input video clip or key frames.

2.5 Computational Complexity of Locality Sensitive Hashing (LSH)

The similarity check performed between the query image(s) and the key frames (extracted from the videos of the dataset) with the help of LSH involves two steps:

- The hash value (of size K) is computed for the query image
- The hash value of the query image is compared against every other hash value of all the key frames (extracted from the videos of the dataset).

The computational complexity of first step is $O(d*K)$ where d is the dimensionality of the embedding dimension and K is the no of hyperplanes. The computational complexity of second step is $O(N/(2\hat{}K))$ where N is the size of the key frames (extracted from the videos of the dataset) and $2\hat{}K$ is the no of regions in the partitioned space of the embedding dimension. As the entire procedure is repeated “L” times, the total computational cost is $O(LdK + ((LdN)/(2\hat{}K)))$. Here, the computation cost for K and L is taken to be in logarithmic time as a similarity search is performed only with a subset of key frames which shares the same hash value with that of the input key frames. So, the computational complexity of LSH algorithm is in logarithmic time. Without the LSH algorithm, the computational complexity would be in quadratic time as a similarity search would be performed against all other key frames.

3 Results and Discussions

We demonstrate the proposed method on nine categories of videos, the description of videos in the dataset that were used for our framework is as shown in Table 1.

We evaluate the proposed method using the precision rate which is represented as follows:

$$\text{Precision rate} = \frac{\text{Number of similar videos retrieved}}{\text{Number of similar videos retrieved} + \text{Number of dissimilar videos retrieved}}$$

The overall precision rate in this framework across all seven categories of videos is observed to be 0.84. The video retrieval results are observed with respect to the videos of each category in the video dataset. The precision rate for the input videos/images that belonged to any one of the "Cricket" (0.85), "Tank" (0.92) and "Sailboat" (0.89) category was found to be slightly better than the precision rate for the input videos/images that belonged to the rest of the categories. The retrieval time for a retrieval of 10 videos is typically less than 5 s. The precision rate with respect to the input images/videos of each category has been presented in the Table 2 and Table 3.

In comparison with a related works [3, 11, 14], our proposed method gives overall better precision rate (0.84) as shown in Table 4.

Sample results of video retrieval

Here, a set of relevant key frames have been retrieved from the total set of key frames (extracted from the video dataset) for every query image, and for an input video clip, a frame (image) is extracted from the video clip and is used as the query. Once the relevant key frames are retrieved, the corresponding videos are retrieved from the

Table 1 Description of dataset

Video category	No of videos	Source	Channel (uploader)
Aeroplane	50	YouTube	British Airways and Emirates
Animals	50	YouTube	National Geographic (Kids)
Flowers	50	YouTube	–
Food	50	YouTube	–
Furniture	50	YouTube	–
Cricket	50	YouTube	Star Sports (match highlights)
Sailboat	50	YouTube	–
Scuba diving	50	YouTube	National Geographic
Tank	50	YouTube	–

Table 2 Precision rate for different categories of videos

S. No.	Category of videos	No of videos in the category	Precision rate
1	Aeroplane	50	0.89
2	Animals	50	0.82
3	Cricket	50	0.85
4	Flowers	50	0.80
5	Food	50	0.81
6	Furniture	50	0.77
7	Tank	50	0.92
8	Sailboat	50	0.90
9	Scuba diving	50	0.80
	Overall	450	0.84

Table 3 Comparison of proposed method with other related works

Video Category	Animals	Flowers	Aeroplane	Cricket	Tank	Sail boat	Scuba diving
Methods							
[3]	0.8	0.7	0.6	0.8	–	–	–
[11]	–	–	1	–	0.72	0.95	–
[14]	0.82	–	–	–	–	–	0.79
Proposed method	0.82	0.80	0.89	0.85	0.92	0.89	0.80

Table 4 Comparison of proposed method with other related works (with respect to overall precision rate)

Authors	Overall precision rate
[3]	0.73
[11]	0.83
[14]	0.81
Proposed method	0.84

video dataset. An example for query image, the retrieved relevant key frames and accordingly, the retrieved videos (corresponding ones) for each of the three categories (i.e., sailboat, cricket and flowers) have been illustrated in the Figs.: Fig. 2, Fig. 3 and Fig. 4.

4 Conclusion

We have proposed a video retrieval framework where the key frames are extracted from the given input video clip and those key frames are checked for similarity

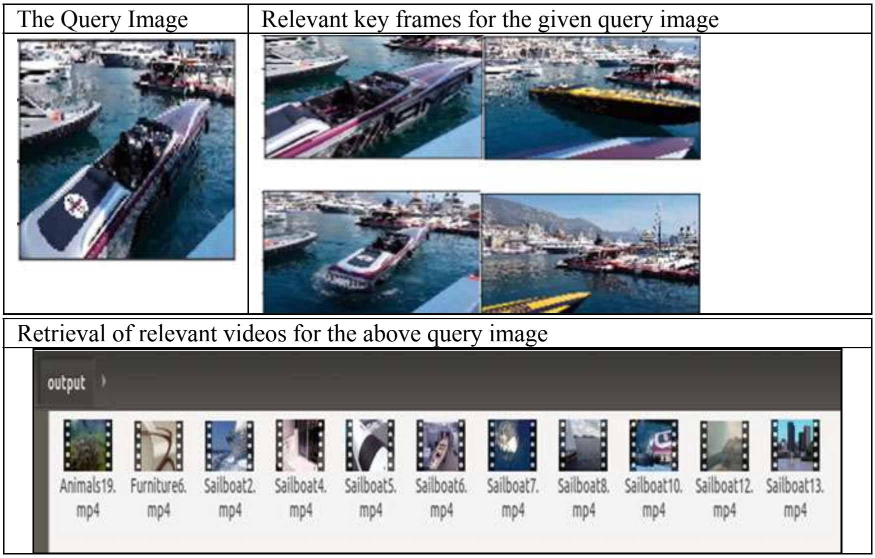


Fig. 2 Sample frames from the retrieved video sailboat category

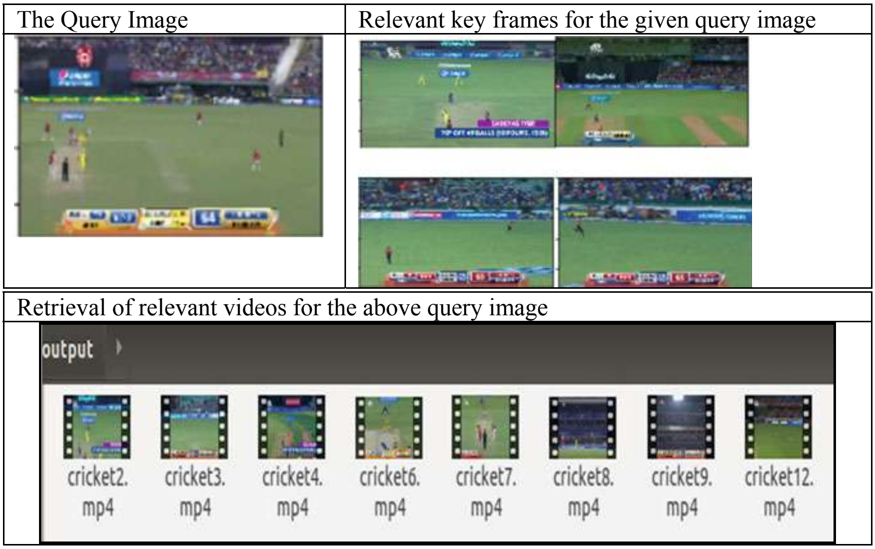


Fig. 3 Sample frames from the retrieved video cricket category

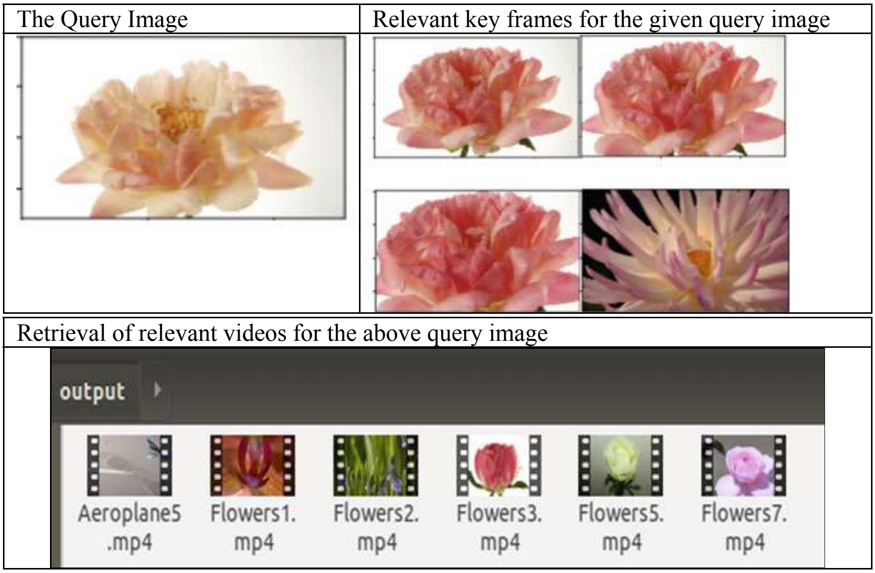


Fig. 4 Sample frames from the retrieved video flowers category

with that of the key frames from the videos of the data set, the corresponding relevant/similar videos are retrieved. The residual network model created a representation of every extracted key frame for which a hash value was generated by the locality sensitive hashing (LSH) algorithm. The LSH algorithm had an important role to play as it improved the retrieval efficiency from n^2 to log n efficiency class. We were able to achieve a precision rate of 0.84 for nine different categories of YouTube videos.

References

1. Desai, P., Pujari, J., Kinnikar, A.: Performance evaluation of image retrieval systems using shape feature based on wavelet transform. In: IEEE Second International Conference on Cognitive Computing and Information Processing (CCIP 2016), pp. 1–5 (August 2016). https://doi.org/10.1109/ccip.2016.7802876
2. Desai, P., Pujari, J., Yaligar, N.: Shape based features extracted using wavelet decomposition and morphological operators. Int. J. Adv. Res. Comput. Sci. (IJARC) **3**(3) (2012)
3. Asha, D., Lata, M., Reddy, V.S.K.: Content based video retrieval system using multiple features. Int. J. Pure Appl. Math. **118**, 287–294 (2018)
4. Patel, B.V., Meshram, B.B.: Content based video retrieval systems. Int. J. UbiComp (IJU) **3**(2) (April 2012)
5. Sujatha, C., Mudenagudi, U.: Gaussian mixture model for summarization of surveillance videos. In: National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), pp. 1–4 (2015)
6. Moula Husain, S. M Meena, Multimodal fusion of speech and text using semi-supervised LDA for indexing lecture videos. In: National Conference on Communications (NCC), pp. 1–9

(2019)

7. Jyothi, V.K., Guru, D.S., Sharath Kumar, Y.H.: Deep learning for retrieval of natural flower videos. Procedia Comput. Sci. **132**, 1533–1542 (2018)
8. Wagh, K.D., Dr. Kharat, M.U.: Content based video retrieval. Int. J. Adv. Res. Comput. Commun. Eng. **5**(1) (January 2016)
9. Mashtalir, S., Mikhnova, O., Stolbovyi, M.: Sequence matching for content-based video retrieval. In: 2018 IEEE Second International Conference on Data Stream Mining and Processing (DSMP), Lviv, pp. 549–553 (2018). https://doi.org/10.1109/dsmp.2018.8478597
10. Yang, M.Y., Liao, W., Cao, Y., Rosenhahn, B.: Video event recognition and anomaly detection by combining gaussian process and hierarchical dirichlet process models. Photogram. Eng. Remote Sens. **84**. https://doi.org/10.14358/pers.84.4.203
11. Sasithradevi, A., Roomi, S.M.M., Maragatham, G.: Content based video retrieval via object based approach. In: TENCON 2017 – 2017 IEEE Region 10 Conference, Penang, pp. 781–787 (2017)
12. Zhang, C., Lin, Y., Zhu, L., Liu, A., Zhang, Z., Huang, F.: CNN-VWII: an efficient approach for large-scale video retrieval by image queries
13. Yasin, D., Sohail, A., Siddiqi, I.: Semantic video retrieval using deep learning techniques. In: 2020 International Bhurban Conference on Applied Sciences and Technology (Jan 2020)
14. Fujii, T., Yoshida, S., Muneyasu, M.: Video retrieval by reranking and relevance feedback with tag-based similarity. In: 2018 IEEE 7th Global Conference on Consumer Electronics (GCCE), Nara, pp. 1–2 (2018). https://doi.org/10.1109/gcce.2018.8574510

Dynamic Search Paths for Visual Object Tracking

Srivatsav Gunisetty, Vamshi Krishna Bommerla, Mokshanvitha Dasari, Vennela Chava, and G. Gopakumar

Abstract The long-term sub-track of visual object tracking challenge comprises of some of the most challenging scenarios like occlusion and target disappearance and reappearance. To this end, many deep learning solutions with multiple levels of detection have been proposed. Most of these solutions tend to re-identify a wrong target during the occlusion or disappearance as they start looking for the target in the entire frame. Instead, through this work, we intend to prove that predicting a probable search region for the target by understanding its trajectory and searching for a target in it will help in reducing the misidentifications and also aid in the increase of IoU. For this, we have utilized the trajectory modeling capabilities of the Kalman filter. With this proof of concept work, we achieved an average improvement of 37.37% in IoU in the sequences where we overperformed MBMD.

Keywords Visual object tracking · Long term · Kalman filter · Occlusion · Mis-classification

1 Introduction

Object tracking is embedded in most of the modern computer vision tasks like gesture recognition, self-driving cars [1, 4], the person following robots [3, 10, 19], etc. It is a well-established field, broadly classified into multiple object tracking and single object tracking based on the tracking objectives. We, in this work, focus on single object tracking where a target provided in the first frame has to be tracked throughout the video sequence handling various challenges. To this end, computer vision scientists have created various benchmarking datasets like OTB50, OTB100 [13], and challenges like Visual Object Tracking (VOT2017 [6], VOT2018 [8], VOT2019 [7])

S. Gunisetty (✉) · V. K. Bommerla · M. Dasari · V. Chava · G. Gopakumar
Department of Computer Science and Engineering, Amrita Vishwa Vidyapeetham,
Amritapuri, India
e-mail: srivatsav1998@am.students.amrita.edu

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_31

to promote research in this direction. The VOT challenge consists of various sub-tracks like short-term, long-term, infrared, and RGB-D, each dealing with a different set of challenges.

We focused on developing a robust solution to the long-term sub-track. VOT2018LT dataset consists of 35 video sequences, each containing an average of 12 long-term target disappearances, lasting on an average of 40 frames [8]. As a whole, this dataset consists of 14,687 frames with 433 target disappearances. The video sequences of long-term sub-track span from 5 to 10 min and pose challenges like occlusion, camera motion, target motion, target disappearance, reappearance, etc., necessitating the tracker to keep note of target disappearances and reappearances. This sub-track observes most of its solutions based on deep learning models [9, 16].

After a careful study on failures of various trackers previously submitted to this sub-track, we noticed that in case of target disappearance, most solutions search the entire frame for re-identifying the target in the consecutive frames, which in most cases ended up identifying a wrong target with almost similar features and led to failure. This approach by the trackers does not work well in scenarios with heavy clutter of objects with similar features to that of target. This issue can be solved if the tracker is provided with information about the possible location of the target object during and after occlusion.

To tackle this, in our work, we exploit the localization capacity of Kalman filter to predict a probable location for the target to reappear in case of occlusion or disappearance. Some approaches use Kalman filter as a standalone solution [11]. We differ from these in our approach as we aim at predicting the search region and then look for the target in it, whereas other solutions directly predict the target location.

Our contribution lies in proving that the approach of predicting a search region by considering the target motion and searching for target in it can achieve scores similar to State-Of-The-Art (SOTA).

2 The Proposed Method

Kalman filter [12], a robust mathematical tool which estimates the state of a process by minimizing the mean squared error of its linear system has been utilized in our case to predict a probable search region. This can estimate present and future states of a process by employing a recursive algorithm among the two categories of equations involved in it, time update, and measurement update. Its localization capability has huge applications like [20].

Structural similarity index (SSIM) [15] is a well-known technique to estimate the quality of digital content [18]. Its applications are in a wide spectrum of problems like compression, restoration, and pattern recognition [17]. Complex wavelet transform variant of the SSIM (CW-SSIM) has been used in our approach as it is more robust to scaling, translation, and rotation [5].

Our pipeline is as follows: we extract the target template denoted by a bounding box in the first frame of the video sequence. For the consecutive frames, if the target is not occluded in the previous frame, we consider 4 times the region where it was found in the previous frame as the search region. Else, we predict a probable search region for the target, scale it based on below defined criterion, and use it as a search region for the current frame. Our template matching strategy is applied to the current frame's search region to identify the target. If the target is found, we update the Kalman with the found location. Else, we update the Kalman with its prediction and set the occluded flag to true. Algorithm 1 shows the pseudo code of our pipeline.

```
for frame in frames do
    if frame is first then
        Template = ExtractTarget (frame);
        IsOccluded = False;
    else
        if IsOccluded then
            PredictedSearchRegion = PredictSearchRegion(); // Kalman prediction
            SearchRegion = ScaleSearchRegion (PredictedSearchRegion);
        else
            SearchRegion = CreateSearchRegion(); // 4 times the previous target location
        end
        TargetFound = SearchForTemplate (SearchRegion);
        if TargetFound then
            UpdateKalman (foundLocation);
            IsOccluded = False;
        else
            UpdateKalman (KalmanPredictedLocation);
            IsOccluded = True;
        end
    end
end
```

Algorithm 1: Overall Workflow

All the video sequences in this sub-track are curated at 30 fps. We observed that the target objects in this dataset do not go beyond the 4 times the region where it was found in the previous frame. Hence, we have used the strategy of taking 4 times the region as search region, which has proved its competence in other SOTA solutions [2].

We represent our process state using an 8 dimensional vector, $[x_1\ y_1\ x_2\ y_2\ v_{x1}\ v_{y1}\ v_{x2}\ v_{y2}]$ where (x_1, y_1) and (x_2, y_2) are coordinates of top left corner and bottom right corner of the bounding box around the target and v_{x1}, v_{y1}, v_{x2}, v_{y2} are their respective velocities. We modeled our situation to fundamental laws of motion, $s = ut + (1/2)at^2$ and $v = u + at$, and obtained values of A and B as Eq. 1, 2, and u_k of time update Kalman equations. u_k, which is acceleration in our case is treated as constant. We have used Eq. 3 as process noise co-variance (Q) and a 4×4 Identity matrix, R as measurement noise co-variance. Using all these values in the Kalman equations, we obtain a search region at frame k. We employ a strategy of scaling up the predicted search region for every 2 occluded frames to incorporate the uncertainty generated due to Kalman prediction and this also helps us in reducing the overhead of searching whole frame for short-term disappearances of target. This scaled search region is treated as probable search region for the frame k.

For template matching, we are using a cross-correlation method. To detect the occlusion we find the SSIM scores of the target found at frame k and the target template. If the score is less than 0.55, we treat it as occlusion. To handle the changing target features throughout the video sequences, we have employed the strategy of updating our base template for every 50 frames based on the SSIM score between target at frame k and base template at frame $k - 50$. If this score is less than 0.6, we do not update our base template. We have exhaustively tried with different values across different video sequences and finally fixed upon these values as they gave a more generalized performance.

$$A = \begin{bmatrix} 1\,0\,0\,0\,t\,0\,0\,0 \\ 0\,1\,0\,0\,0\,t\,0\,0 \\ 0\,0\,1\,0\,0\,0\,t\,0 \\ 0\,0\,0\,1\,0\,0\,0\,t \\ 0\,0\,0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,0\,0\,0\,1 \end{bmatrix} \quad (1) \qquad B = \begin{bmatrix} (t^2)/2 \\ (t^2)/2 \\ (t^2)/2 \\ (t^2)/2 \\ t \\ t \\ t \\ t \end{bmatrix} \quad (2)$$

$$Q = \begin{bmatrix} (t^4)/4 & 0.0 & 0.0 & 0.0 & (t^3)/2 & 0.0 & 0.0 & 0.0 \\ 0.0 & (t^4)/4 & 0.0 & 0.0 & 0.0 & (t^3)/2 & 0.0 & 0.0 \\ 0.0 & 0.0 & (t^4)/4 & 0.0 & 0.0 & 0.0 & (t^3)/2 & 0.0 \\ 0.0 & 0.0 & 0.0 & (t^4)/4 & 0.0 & 0.0 & 0.0 & (t^3)/2 \\ (t^3)/2 & 0.0 & 0.0 & 0.0 & t^2 & 0.0 & 0.0 & 0.0 \\ 0.0 & (t^3)/2 & 0.0 & 0.0 & 0.0 & t^2 & 0.0 & 0.0 \\ 0.0 & 0.0 & (t^3)/2 & 0.0 & 0.0 & 0.0 & t^2 & 0.0 \\ 0.0 & 0.0 & 0.0 & (t^3)/2 & 0.0 & 0.0 & 0.0 & t^2 \end{bmatrix} \quad (3)$$

3 Results

Object tracking utilizes intersection over union (IoU) as a metric to represent robustness of a tracker. We have tested our tracker's performance on 20 carefully curated video sequences of VOT2018 LT dataset in such a way that each of them represents different set of challenges. We have compared our tracker's performance with the winner of that challenge, MBMD [14] to check the legitimacy of our assumption. All the experiments are done at the original scale and the reported images are at a lower scale to save space. Our tracking outputs have dual bounding boxes, blue

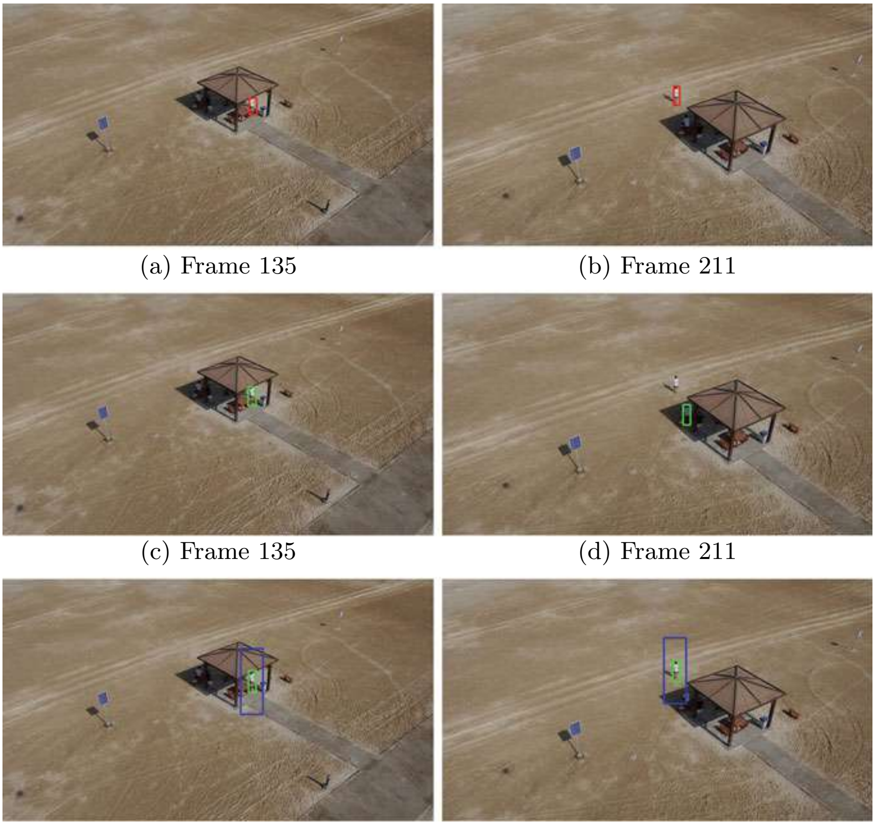
Table 1 List of video sequences with improved scores

Sequence name	No. of frames	Our IoU	MBMD IoU	Improvement in %
ballet	1389	0.554	0.368	50.50
bicycle	2842	0.529	0.413	27.78
bird1	2437	0.02	0.01	110
carchase	9928	0.368	0.284	29.3
person14	2924	0.748	0.705	6.11
rollerman	1712	0.679	0.516	31.64
tightrope	2291	0.601	0.565	6.27

representing our search region and green is the identified target. We have segregated our observations into two tables, Table 1 where our tracker performed better than MBMD [14] and Table 2 where we fall behind MBMD [14]. In Table 1, we have calculated the improvement gained using the formula $\frac{(x-y)*100}{y}$ where x is Our IoU and y is MBMD's IoU.

From Table 1, we observe our tracker overpowering the performance of MBMD [14] in few video sequences with our most accurate search regions. Figure 1 depicts a scenario where the target passes through a hut and comes out after a few seconds. The main challenge in this sequence lies in preventing the tracker from tracking the other person standing beside the hut with almost similar features to that of the target. Most of the tracking solutions, including MBMD [14] search the entire frame for the target soon after encountering occlusion or disappearance. This may lead to mis-identification of target as shown in Fig. 1c and d where Fig. 1a and b shows the expected tracking output. Our solution was successful in understanding the trajectory of the target and was able to track the target soon after the occlusion as shown in Fig. 1e and f. Our predicted search regions also help in reducing the need for searching the whole frame during short-term disappearances, there by reducing the computations. Figure 2 is part of person17 video sequence where a person walks in the park and sometimes gets occluded by trees. Figure 2a shows the beginning of one such occlusion cases. The target was occluded for 57 frames, i.e., starting from frame 560 to 617. During the course of occlusion, our tracker was constrained to search only certain region of the frame as shown in Fig. 2b and this has reduced the unnecessary overhead of searching the whole frame for short-term disappearances. Figure 2c shows the successful re-identification of the target.

Our solution can tackle moderate cases of scaling, clutter, change in angle, and change in direction even without a dedicated implementation for the same and achieved comparable results to that of MBMD [14] as shown in Table 2. This was possible because of our current strategy which updates our base template for every 50 frames based on the threshold as discussed in Sect. 2. This approach has provided us with good results in some sequences as shown in Fig. 4. Figure 3 depicts one of



(a) Frame 135 (b) Frame 211

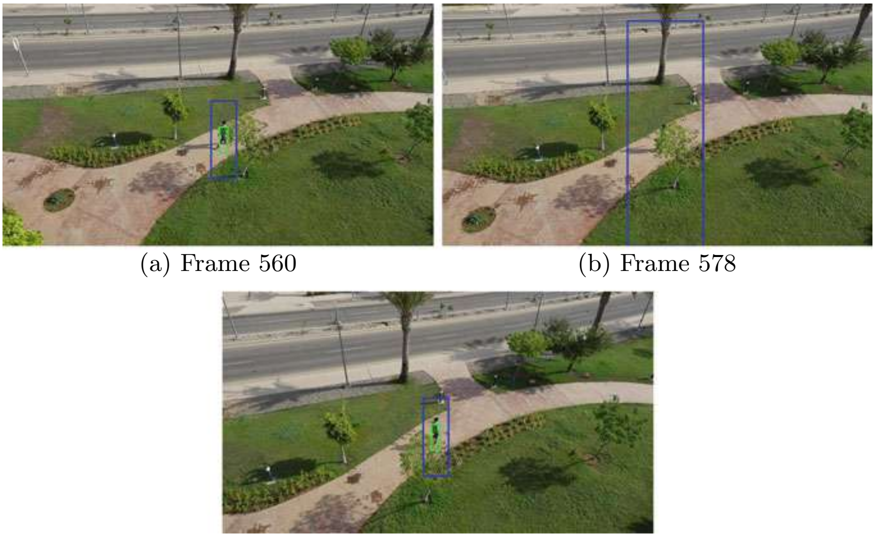
(c) Frame 135 (d) Frame 211

(e) Frame 135 (f) Frame 211

Fig. 1 Tracking results during occlusion. **a** and **b** The ground truths. **c** and **d** MBMD predictions. **e** and **f** Our predictions

the scenarios where our tracker was able to contain the target in our search region but was not able to adapt to the changing scale of the target.

Despite achieving notable results as shown in Tables 1 and 2, our tracker fails at sequences with adverse levels of challenges mentioned above. This is shown in Fig. 5c and d. With our observations, we understood that a scale aware and motion aware template update strategy can aid in reducing the failures due to change in angle, change in direction and scaling. Our template matching strategy directly works on the pixel values, leaving scope for being fooled by other objects with similar pixel values within the search region as shown in Fig. 5a and b. To this end, we hope to implement a Siamese-based network and devise a strategy to handle our varying sizes of search region in our future work.

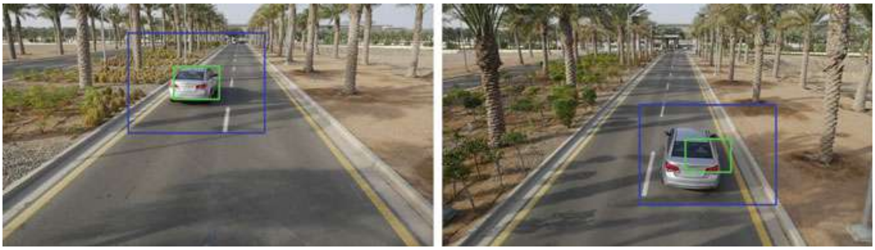


(a) Frame 560

(b) Frame 578

(c) Frame 617

Fig. 2 Successful re-identification of target in person17



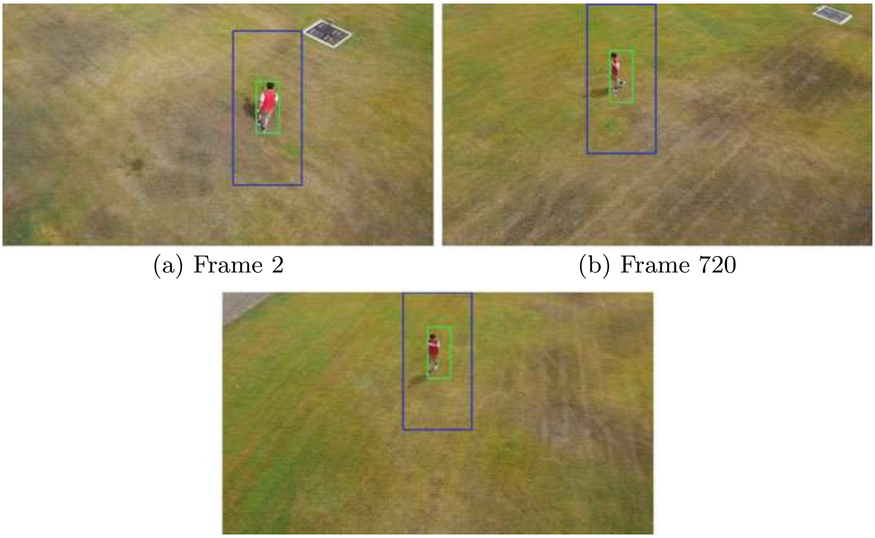
(a) Frame 69

(b) Frame 377

Fig. 3 Tracking results on car6 with current strategy

Table 2 List of video sequences with comparable scores

Sequence name	No. of frames	Our IoU	MBMD IoU	Reason for failure
group3	5527	0.678	0.694	Clutter
yamaha	3143	0.489	0.537	Scaling
skiing	2654	0.610	0.635	Scaling
person5	2101	0.547	0.723	Scaling
longBoard	7059	0.448	0.561	Change in angle
person17	2347	0.412	0.755	Change in angle
person4	2743	0.669	0.819	Change in direction
person7	2065	0.588	0.603	Change in direction

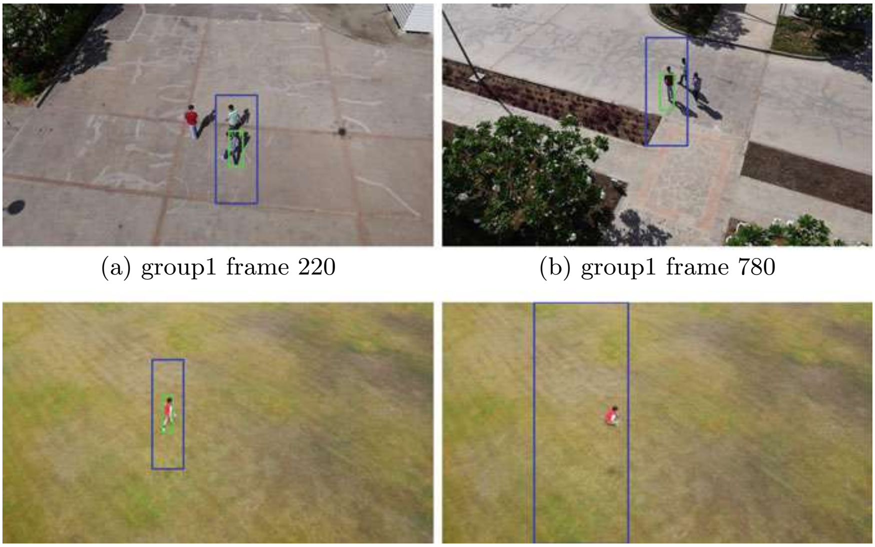


(a) Frame 2

(b) Frame 720

(c) Frame 1022

Fig. 4 Tracking results on person5 with current strategy



(a) group1 frame 220

(b) group1 frame 780

(c) person7 frame 818

(d) person7 frame 831

Fig. 5 Issues in our solution

4 Conclusion

With the above results and observations, we conclude that predicting a search region accurately and localizing the target inside it aids in improving the tracking results by reducing the mis-identifications of target. With further research in the directions as suggested in the above section, we hope that our solution will progress into a more robust, high performing, and model-free tracker for the long-term sub-track.

References

1. Agarwal, N., Chiang, C.-W., Sharma, A.: A study on computer vision techniques for self-driving cars. In: International Conference on Frontier Computing, May 2019, pp. 629-634 (2019). ISBN: 978-981-13-3647-8. https://doi.org/10.1007/978-981-13-3648-5-76
2. Bertinetto, L., et al.: Fully-convolutional siamese networks for object tracking. CoRR abs/1606.09549 (2016). arXiv:1606.09549
3. Chen, B.X., Sahdev, R., Tsotsos, J.K.: Person following robot using selected online Ada-Boosting with stereo camera. In: 2017 14th Conference on Computer and Robot Vision (CRV), pp. 48–55 (2017)
4. Cho, H., et al.: A multi-sensor fusion system for moving object detection and tracking in urban driving environments. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 1836–1843 (2014)
5. Gao, Y., Rehman, A., Wang, Z.: CW-SSIM based image classification. In: 2011 18th IEEE International Conference on Image Processing, pp. 1249–1252 (2011)
6. Kristan, M., et al.: The visual object tracking VOT2017 challenge results. In: 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), pp. 1949–1972 (2017)
7. Kristan, M., et al.: The Seventh Visual Object Tracking VOT2019 Challenge Results (2019)
8. Kristan, M., et al.: The sixth Visual Object Tracking VOT2018 Challenge Results (2018)
9. Lee, H., Choi, S., Kim, C.: A memory model based on the Siamese network for long-term tracking. In: Leal-Taixé, L., Roth, S. (eds.) Computer Vision—ECCV 2018 Workshops, pp. 100–115. Springer, Cham (2019). ISBN: 978-3-030-11009-3
10. Ren, Q., et al.: Real-time target tracking system for person-following robot. In: 2016 35th Chinese Control Conference (CCC), pp. 6160–6165 (2016)
11. Saho, K.: Kalman filter for moving object tracking: performance analysis and filter design. In: de Oliveira Serra, G.L. (ed.) Kalman Filters. IntechOpen, Rijeka (2018) (chap. 12). https://doi.org/10.5772/intechopen.71731
12. Welch, G., Bishop, G., et al.: An introduction to the Kalman filter (1995)
13. Wu, Y., Lim, J., Yang, M.-H.: Online object tracking: a benchmark. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2013)
14. Zhang, Y., et al.: Learning regression and verification networks for longterm visual tracking (2018). arXiv: 1809.04320 [cs.CV]
15. Wang, Z., et al.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
16. Zhu, Z., et al.: Distractor-aware Siamese networks for visual object tracking. CoRR abs/1808.06048 (2018). arXiv: 1808.06048
17. Ahmed, S., Krishnnan, N., Ganta, T., Jeyakumar, G.: A video analytics system for class room surveillance applications. Int. J. Recent Technol. Eng. (IJRTE) **7**(5S3) (2019)
18. Sowmya, V., Govind, D., Soman, K.P.: Significance of contrast and structure features for an improved color image classification system. In: 2017 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), pp. 210–215 (2017)

19. Jose, E.K., Veni, S.: YOLO classification with multiple object tracking for vacant parking lot detection. J. Adv. Res. Dyn. Control Syst. **10**, 683–689 (2018)
20. Sudheesh, P., Jayakumar, M.: Non linear tracking using unscented Kalman filter. Adv. Intell. Syst. Comput. **678**, 38–46 (2018)

Thermal Facial Expression Recognition Using Modified ResNet152

Aiswarya K. Prabhakaran, Jyothisha J. Nair, and S. Sarath

Abstract Facial expression for emotion detection has taken wide popularity with visible images using machine learning techniques and convolutional neural networks. However, emotion recognition from visible images is not much plausible as they are sensitive to light conditions and people can easily fake expression. In this paper, we propose a method for facial expression recognition with thermal images using ResNet152. Residual networks are easier to optimize, and can gain accuracy from considerably increased depth. The objective of this paper is to use a pre-trained modified ResNet152 to train thermal facial images in order to predict different emotions. We use natural visible and infrared facial expression (NVIE) dataset for emotion classification.

Keywords ResNet152 · Thermal images · Thermal patterns · Transfer learning · Residual representation functions

1 Introduction

Facial expressions are a reflection of human feelings. It is a nonverbal way of emotional expressions. Automated emotion recognition is considered to be a solid evidence to detect if a person is lying during interrogation. It is also used for tracing a patient's pain analysis and also for stress analysis and counseling. There are several drawbacks for facial emotion recognition with visible images such as people can easily fake expressions and quality of lighting is major factor for these images

A. K. Prabhakaran · J. J. Nair (✉) · S. Sarath
Department of Computer Science and Engineering, Amrita Viswa Vidyapeetham, Amritapuri, India
e-mail: jyothishaj@am.amrita.edu

A. K. Prabhakaran
e-mail: kpaiswarya10@gmail.com

S. Sarath
e-mail: saraths@am.amrita.edu

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_32

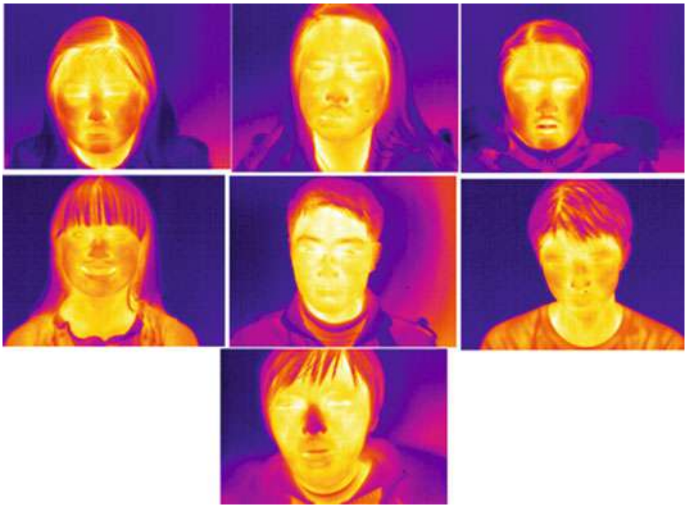


Fig. 1 Types of emotion in the dataset anger, disgust, fear, happiness, neutral, sadness, and surprise [2]

[1]. In order to overcome this, emotion recognition with thermal images happens to be steadfast technique. The process of converting heat into visible images is called thermal imaging. Heat signature is the infrared energy emitted by an object. We can say that the radiation emitted by an object depends upon the hotness of the object (Fig. 1).

Thermal camera can be defined as a sensor that detects even the smallest difference in temperature. There are two types of thermal cameras, namely uncooled IR cameras and cooled IR cameras. Long wave IR that is LWIR spectrum that ranges between 7.5 and 14 μm specifically uses uncooled cameras whereas middle wave IR MWIR that ranges between 3 and 5 μm uses cooled cameras. Majority of researches uses LWIR and MWIR sub-band-supported cameras. In general, the human face and body temperature range from 35.5 to 37.5 °C which means that the thermal signature is consistent. The pattern of superficial blood vessels under the skin, the vein, and tissue structure of the face is the major contributors of thermal patterns [5]. These thermal patterns will be unique to each person which makes the IR images distinctive. Therefore, facial emotion recognition with thermal images are independent of skin color and facial features. There happens an increase in heat of facial temperature when an external sound, object, or action takes place. Thermal images has smaller within class variance. It is very useful for detecting disgust but it is not much accurate when the subject is wearing a spectacle since a large portion of glass is blocked by thermal energy. There are several other challenges with thermal images such as thermal patterns with identical twins are almost similar [5], also it reduces the affect of obstruction due to beard and mustache.

Each individual pixel in a thermal image represents a thermal energy emitted by a particular datapoint. These pixels are colored pallets that holds some temperature value in the form of various colors. Thermogram outputs are RGB images that holds a particular value of temperature colored by a heat map. We here use an ironbow palette which is most commonly used for thermal anomalies and heat generated from a body. Hotter objects are represented with warm colors such as yellow, orange, and red while cold objects are represented with cool colors such as blue and green violet.

Convolutional neural networks are inspired from the connectivity pattern of neurons in the human brain. This method is adopted by various other authors for several other applications [11, 12]. AlexNet, GoogleNet, VGGNet, ResNet, and ZFNet are some of the architectures of convolutional neural network that strengthens to build algorithms which are more powerful in nature. Among which VGGNet, ResNet, and GoogleNet [13] are some of the most popular models for image recognition because of its performance and architectural innovations. Here, in this paper, we are following a method of transfer learning to accomplish the task of emotion recognition using thermal images. Transfer learning is otherwise called as process of solving a problem with the help of the knowledge gained while solving a problem of similar type. ImageNet is a very large database that is used for research purpose, which consists of 14 million images which belongs to more than 20K classes. Trained models constitute of model architecture and model weights. There exists very few thermal databases hence getting enough data is infeasible. So we use pre-trained model which loads the weight trained on imageNet data. There arise a question of whether we can use pre-trained model which are trained on visible images for classifying thermal images, absolutely yes networks that are trained on a particular set of images can be used for transfer learning which has a different types of image [4].

In our research, we first opted VGG16 and ResNet152 to solve this problem, but later, ResNet152 outperformed VGG16 due to its fully connected layers and its depth. It is around 574 mb which makes it difficult to deploy. Whereas ResNet152 uses global average pooling instead of fully connected layer and the model size is over 102 MB. which makes it easy to deploy and much faster.

2 Related Works

Basu et al. [6] proposes a method in which the data is preprocessed and this pre-processed data is been given to multi-class SVM for classification. Data preprocessing steps consist of Hu's moment invariant method to calculate the feature vector from six subregions which includes forehead, left cheek, right cheek, eyes, mouth, and nose to create a ROI for emotion. They used KTFE database and have an accuracy of 87.5%.

Another method consists of three major steps for data preprocessing; these are PCA for dimensionality reduction, ADA boost algorithm to boost calculation by reducing the number of weight and SIFT algorithm to extract eyes nose and lips for detailed description of image and GLCM for holding the number of pixels and position details [7]. The pre-processed image is been fed to feed forward neural network for classification. This method had an accuracy of 90% for 80 pictures.

The next method is a video image processing with thermal images to detect emotions [8]. Data pre-processing step is similar to the previous ones that is discussed in above-mentioned papers. Once the ROI is been taken, it is analyzed based on fuzzy logic by considering four inputs, namely eyes, forehead, cheeks, nose, and maxillary. After some time span, these ROI are again taken and analyzed. The next step is the calibration technique where these two ROI are taken and compared with each other while forming the emotion and the final conclusion is placed in either of three groups that is low, normal, and high. If the temperature is decreased, then that particular ROI will fall in low class, if its temperature is still, then ROI will be in normal class, and high means the ROI temperature is increased. Finally, this ROI is been given to a classifier to classify it into one of five emotions that is joy, disgust, fear, anger, and sadness. The overall accuracy was around 89.9%.

The next method is proposed by Burak et al. [1]. It uses a deep learning model YOLO Darknet V3 which consists of 53 convolutional layers and pretrained on imageNet database. YOLO basically works by dividing the image into SxS grid and each grid will encounter an object if the center of the object falls in that grid [10]. It uses logistic regression to calculate the objectness score for each bounding box, so if there are multiple labels in a picture, then the bounding box with the highest objectness score is been considered. It goes around 9000 epochs and comes with an accuracy of 90% for anger, disgust, happiness, surprise, and sadness.

The paper proposed by Olivier et al. [4]is based on machine health monitoring system which uses deep learning for infrared images. The objective of this paper is to automatically discover the condition of the machine. There are two cases for this; first one is the machine fault detection and second comes the oil level prediction for which it got an accuracy of 95% and 91.67%, respectively. The peculiarity of this paper is that it uses trained neural network to get this accuracy with infrared images. It explains how neural networks trained on ImageNet can be used for classifying thermal images. It uses pre-trained VGG net for classification.

3 Proposed Method

A set of multiple variables called as pixels constitute an image. An image is considered to be a complex data; therefore, in order to train these complex data, we need to have a complex networks. Since there are very less thermal image dataset for expression recognition. Here, we use a pre-trained network to classify the emotion to various classes. As VGGNet and ResNet are popular for image classification, we considered both of this network to train our dataset. But finally, in terms of time taken and space required by the network while implementing, ResNet was outperforming VGGNet. Increasing the network layer is not a good idea to get better accuracy. This will actually increase the training error. ResNet is introduced to solve this problem, which is actually similar to highway networks but gives a better result. ResNet will

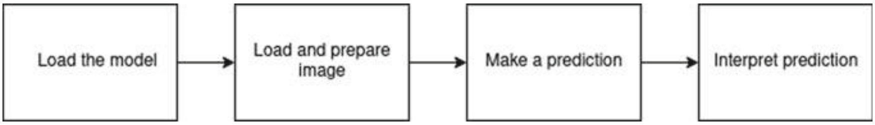


Fig. 2 Steps followed in transfer learning

skip one or more layers and help to avoid the vanishing/exploding gradient problem using a technique called identity shortcut connection. The advantage of this method is this will skip the layer which affects the performance of the network. Hence, we use pre-trained ResNet 152 to train our dataset.

This process of using a pre-trained network is called as transfer learning in deep learning concept. Transfer learning is a process where we are taking the advantage of a model created for a solution to solve a similar type of problem. Where the inputs will be the same but the output differs. Transfer learning will reduce the time taken to train a network. In this method, a pre-trained network which can be commonly used for different image processing applications can be used to solve our particular problem. This method will reduce the time taken to train a network to solve our problem. We can also get better accuracy with this smaller dataset by using the transfer learning technique. The process is first we will train the neural network with a larger dataset and train the network to a particular application then the network will be fine-tuned for the specific problem using the smaller dataset. Figure 2 shows the methods followed in transfer learning.

ResNet is deep network but substantially smaller due to global average pooling layer. It has won the first place in ILSVRC and COCO 2015 competition in ImageNet Detection, ImageNet localization, Coco detection, and Coco segmentation. ImageNet is a large image dataset. ResNet contains 152 layers that were trained on natural images. The objective of ResNet is to classify the images to one of thousand categories. Each layer consists of Convolutional (conv), Rectified Linear Unit (RELU) and Batch Normalization (Batch Norm) layers [3]. ResNet makes use of the Identity Connection as shown in Fig. 3, which helps to protect the network from vanishing gradient problem. ResNet preserves neural network that has learnt by not applying diminishing transformation or if the layer is able to learn, it will add on to what the network has learnt. ResNet will take care of maximum learning of weights possible without over fitting or negative deviation of accuracy.

In short, we can say that the methodology which is adapted in this paper is transfer learning. We followed each steps as in Fig. 2 to implement this system. Primary step was to load the pre-trained ResNet with input parameters such as batch size, number of epochs, steps per epoch and number of classes to which it is classified and then we preprocessed the data and divided the data to training and validation sets. This pre-processed data is given as the input to the model. In order to fine tune the model, which is already trained to perform the given task, and to make it perform a similar task, we gave two fully connected layers. In the last, fully-connected layer softmax activation functions is used. This layer with softmax activation functions provides a

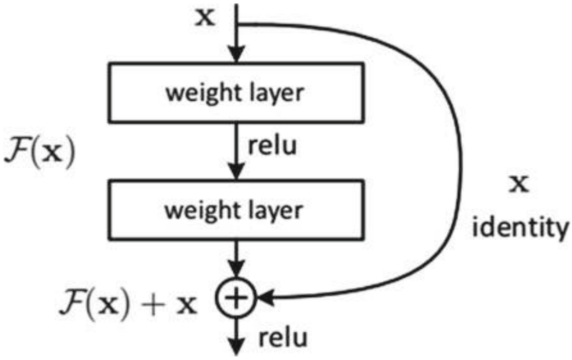

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Fig. 3 Building block of residual learning [3]

probabilistic mutually exclusive classification. We gave categorical cross entropy as a loss function. It is used in multi-class classification tasks where an object can only belong to one out of many possible categories, and the model must decide which one. We also used SGD optimizer which is a variant of gradient decent. SGD only computes on a small subset or random selection of data. We first considered for Adams optimizer but since it did not give a satisfactory result, we opted for SGD. Recent research papers have noted that Adams can fail to converge to an optimal solution under specific settings and also it generalizes poorly compared to SGD [9].

4 Results and Analysis

In this paper, we have proposed a method of emotion classification of thermal images using ResNet152.

NVIE dataset is been fed to VGG16 which gave an accuracy of nearly 57 % for 100 epochs. It was very slow because of fully connected layers. Ilikci et al. [1] proposed that YOLO Net had a classification precision of 90% for each class except fear and neutral for 9000 iteration. Whereas our modified ResNet152 gave an accuracy of 87.3% and a precision of 94% for the same dataset for a lesser number of epochs as shown in Figs. 4 and 5. Hence, our model is considered to be much faster and reliable than other networks.

Here we used a natural visible and infrared facial expression database as shown in Fig. 1, which consists of both posed and spontaneous expression data [2]. These expressions are recorded by a visible and infrared cameras simultaneously, with light illumination from three different direction. We considered a total of 1712 images in which 1392 images are categorized as training set images and 320 images are categorized as validation set images. There are seven posed expression data in the database, namely anger, disgust, fear, happiness, neutral, sadness, and surprise.

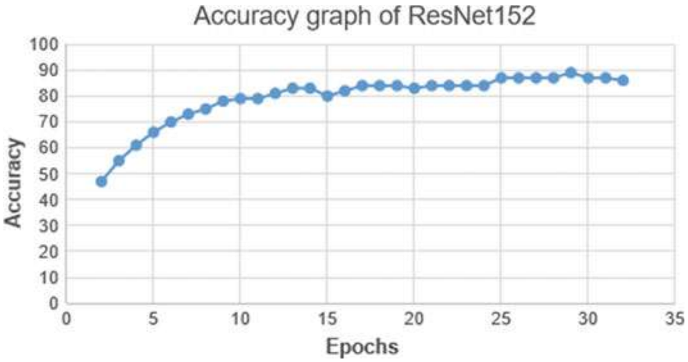


Fig. 4 Accuracy plot

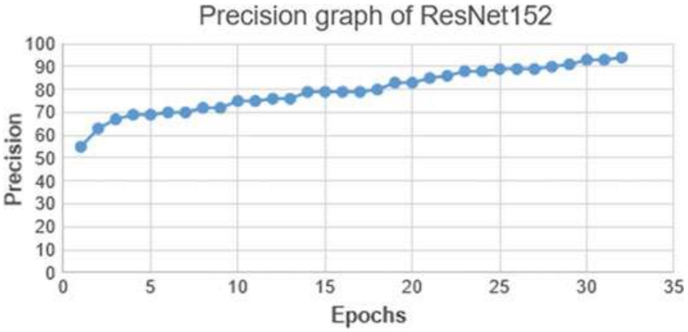


Fig. 5 Precision plot

5 Conclusion

In this paper, we have seen that neural network can be used as feature learning tool which can detect various emotions from thermal images. Deep neural networks need large amount of data to train; hence, to resolve this issue, we use transfer learning technique, which is a method of reusing layers of a pre-trained neural network. In this paper, we came across a method of facial expression recognition using thermal images with pre-trained modified ResNet152. We noticed that this dataset has a better accuracy and precision with proposed model when compared to other neural networks such as YOLODarknet and VGG Net. Our proposed model is faster than VGG and is better in classifying than YOLONet.

References

1. Ilikci, B., et al.: Heat-map based emotion and face recognition from thermal images. In: 2019 Computing, Communications and IoT Applications (ComComAp). IEEE (2019)
2. Wang, S., et al.: A natural visible and infrared facial expression database for expression recognition and emotion inference. IEEE Trans. Multimed. **12**(7), 682–691 (2010)
3. He, K., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
4. Janssens, O., et al.: Deep learning for infrared thermal image based machine health monitoring. IEEE/ASME Trans. Mechatron. **23**(1), 151–159 (2017)
5. Bhowmik, M.K., et al.: Thermal infrared face recognition—a biometric identification technique for robust security system. Reviews, Refinements and New Ideas in Face Recognition, p. 7 (2011)
6. Basu, A., et al.: Human emotion recognition from facial thermal image based on fused statistical feature and multi-class SVM. In: 2015 Annual IEEE India Conference (INDICON). IEEE (2015)
7. Priya, M.S., Kadhar Nawaz, G.M.: Modified emotion recognition system to study the emotion cues through thermal facial analysis (2017)
8. Cruz-Albarran, I.A., et al.: Human emotions detection based on a smart-thermal system of thermographic images. Infrared Phys. Technol. **81**, 250–261 (2017)
9. Keskar, N.S., Socher, R.: Improving generalization performance by switching from ADAM to SGD. arXiv preprint arXiv:1712.07628 (2017)
10. Sarath, S., Chaitanya, Malavika, Prasanna, Karthik: Human emotions recognition from thermal images using Yolo algorithm. In: 2020 International Conference on Communication and Signal Processing (ICCSP). IEEE (2020)
11. Sreekumar, A., Nair, K.R., Sudheer, S., Ganesh Nayar, H., Nair, J.J.: ‘Malignant lung nodule detection using deep learning. In: 2020 International Conference on Communication and Signal Processing (ICCSP), pp. 0209–0212. IEEE (2020)
12. Aloysius, N., Geetha, M.: A review on deep convolutional neural networks. In: 2017 International Conference on Communication and Signal Processing (ICCSP). IEEE (2017)
13. Sathyan, H., Vinitha Panicker, J.: Lung nodule classification using deep ConvNets on CT images. In: 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE (2018)

Real-Time Retail Smart Space Optimization and Personalized Store Assortment with Two-Stage Object Detection Using Faster Regional Convolutional Neural Network

Nitin Vamsi Dantu and Shriram K. Vasudevan

Abstract In the present-day scenario of the retail environment, there is a tendency of customers to do prolonged shopping. During their constant efforts to purchase products of their choice, they bat around the entire store. They choose a product and continue exploring for more. While their further exploration, there is a possibility that they might encounter a better product that may satisfy their needs. So, there is a tendency that they may pick up the new product, compare with the existing product and leave the latter behind if they find the new product to be of a better purpose to their use, causing the initial product to be misplaced. There might be a possibility that empty spaces be created between products that might look sparse and lower stock display if not properly monitored. The primary goal of this research is smart space management and personalized store assortment by using computer vision. That is, wherever there are empty spaces created, we constantly monitor, and whenever there is any product misplaced, we send an automated notification to corresponding staff. We use state-of-the-art computer vision technology to address this issue. All the processing is done in real time and the system is found to be functionally very stable and works under all ideal conditions.

Keywords Retail products recognition · Faster-RCNN · Transfer learning · Localization · Classification · Artificial intelligence

1 Introduction

Managing inventory by retailers in retail stores is one of the import tasks and needs to replenish the shelves with proper items. Sometimes, a customer picks up an item and changes his/her mind and inadvertently places the picked-up item in the nearest

N. V. Dantu
Department of Computer Science and Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India

S. K. Vasudevan (✉)
K. Ramakrishnan College of Technology, Samayapuram, Tamilnadu, India
e-mail: Shriramkv@gmail.com

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_33

rack which has different items. Retailers need to arrange again all misplaced items in proper shelves based on the item and its brand. As per the below statistics, the retail sales increase year on year. In 2018, the overall revenue from retail sales in the US was $6 trillion. Still, in 2019, 90% of retail sales are happening in offline retail stores.

Problems faced by the retail industry are:

1. Keeping up the customer expectations.
2. Manage customer loyalty.
3. Managing internal communication.
4. Retaining and engaging employees.
5. Digital disruption.
6. For finding the best technology solutions for the retail industry.

We are trying to address the sixth problem; retailers have been looking for best-automated solutions to simplify their business processes and to overcome retail challenges that they face in daily activities. It is observed that retailers spend a lot of time in identifying which shelves are empty and which items are misplaced for rearrangement.

We are trying to address the below problems:

1. Empty space localization across the isle
2. Misplaced items identification in an isle
3. Items that need to be replenished in an isle.

We would need to set up cameras at multiple points along every shelf in the store. We can use deep learning techniques to train models to detect when certain types and brands of products are picked up from the shelves indicating that there is a space and need to be replenished the same brand and item. The machine vision algorithm behind the cameras finds any misplaced items on different shelves. Instead of an employee wasting his/her time to find out the empty spaces and misplaced items in the isles, we may use the intelligent computer vision system to alert the manager that replenishment is required and misplaced items found at the specific aisle.

2 State of the Art

Joseph Redmon et al. have introduced to a very fast single-stage object detection model, you only look once (Yolov3). This can process input video stream real time at 45 frames per second and is considered to be one of the fastest object detection algorithms but is hit by lower mean average precision of 44.0 on the MSCOCO dataset [1].

Wei Liu et al. have researched object detection models and used the SSD: Single shot multiBox detector (SSD) model for object detection. This is an outstanding model for object detection which is capable of localizing and classifying objects in one go. It provides fast object detection and has moderate mean average precision of (50.4) on MSCOCO dataset [2].

There are many other object detection techniques proposed and developed. But, the accuracy and time taken to detect the objects have always been a concern [3].

We propose our solution using Faster-RCNN which has a mean average precision of 57.7 on the MSCOCO dataset which is considered to be the most accurate object detection algorithm compared to the rest. Though compromised a bit on processing speed, it is the most accurate object detection model. We need to be extra careful and very precise when detecting different products and ensure we avoid any sort of false positives and true negatives. For this reason, we chose the most accurate, the two-stage object detection model, Faster-RCNN with inception v2 as the backbone.

3 Proposed Approach

We constantly monitor the target area by placing cameras. The cameras can either live stream to our model or can send pre-recorded videos. The video is then sent to our two-stage object detection model, Faster-RCNN for localization and classification which is capable of detecting six different products, and this can be further extended through training on an extensive dataset. The object detection model precisely does product detection, misplaced product identification and space localization, and sends an automated notification to corresponding staff if any action is required. The model can classify and localize precisely irrespective of the shape, irrespective of the color and irrespective of the angle that the object is present in. The staff can get an analysis of the sales, number of items that are misplaced, number of slots that are empty and could be filled with more products at any moment, and based on this, we can also do item demand prediction for the next day. The entire system is designed in such a way that there is no additional effort is required by the user and it is just plugged and play.

4 Architecture

We have used a two-stage object detection model for localization and classification of objects, refer Fig. 1. We send the images to the Faster-RCNN for the model to predict and localize different products along with giving information about misplaced products and empty slots where newer items could be filled. This is done in real time with 15 FPS on a decent GPU and 30FPS on a high-end GPU. Let us understand the architecture of the state-of-the-art Faster-RCNN object detection model.

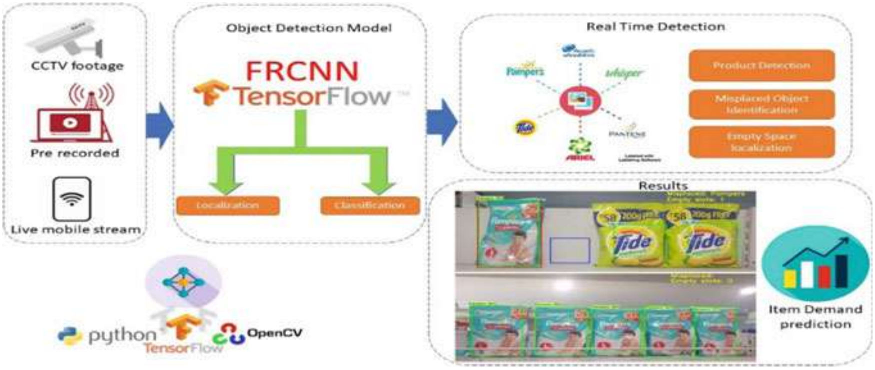


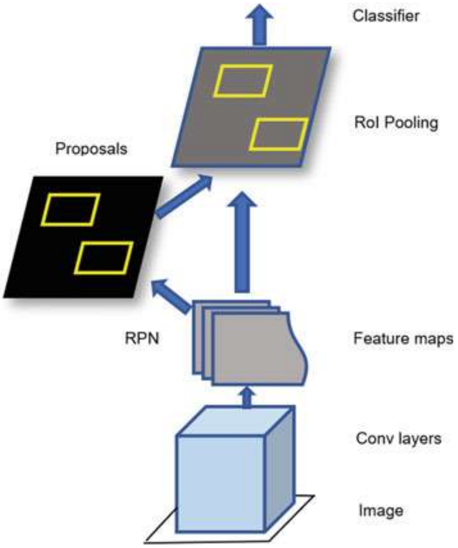
Fig. 1 Architecture diagram

4.1 Faster RCNN

As we have already discussed, the Faster-RCNN is a two-stage object detection model, meaning it would have to predict the bounds where the object might be present at the first stage and in second stage classify the regions as specific object. To predict the regions, classical methods use selective search and it is very time consuming but the Faster-RCNN uses the region proposal network (RPN), refer Fig. 2.

The Faster RCNN architecture consists of two prime components RPN as a region proposal component and the Fast RCNN as object detection network with inception v2 as feature extractor with transfer learning model. We shall first look into the

Fig. 2 Region proposals



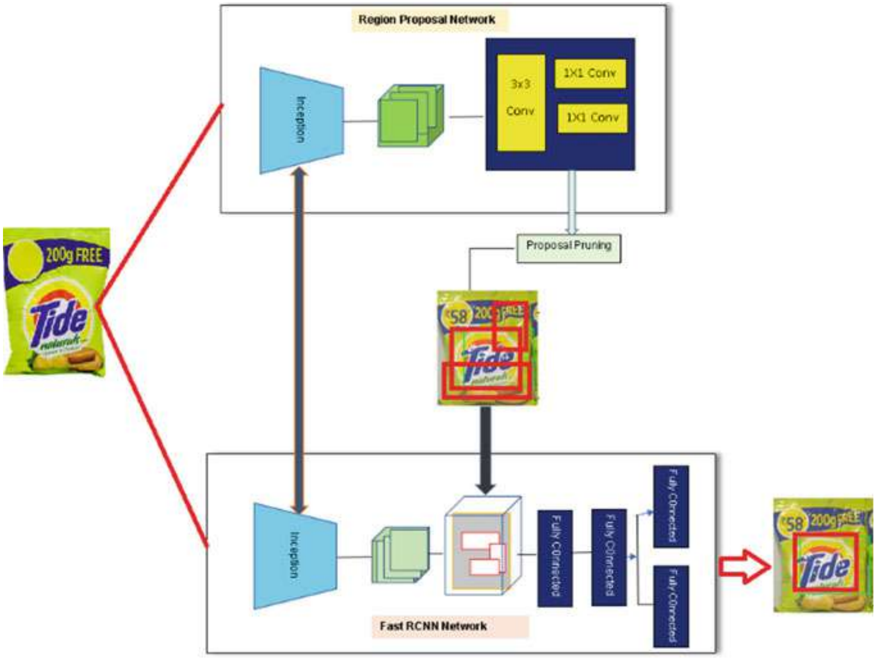


Fig. 3 Faster RCNN architecture

backbone inception network in Sect. 4.1.1 and then move on to regional proposal network in Sect. 4.1.2 and Faster-RCNN as object detector in Sect. 4.1.3 (Refer Fig. 3).

4.1.1 The Inception Network

Inception network is a very deep convolutional neural network used for feature extraction, refer Fig. 4. This network is created to increase the performance of the

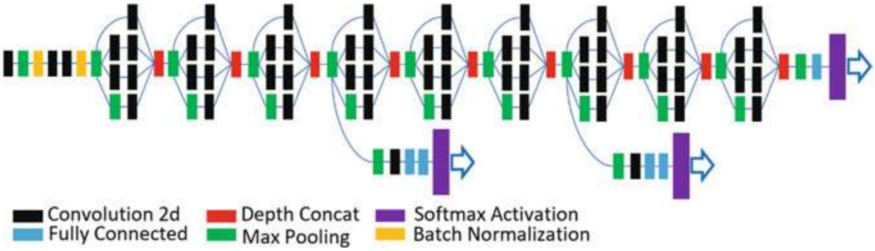


Fig. 4 Inception version-1

object classification and capture global as well as local features in the feature maps. The concept behind the inception network is to use filters with multiple sizes at the same level, making it horizontal rather than vertical so that global as well local features will be captured. So, this makes the inception network to work a great feature extractor. Inception v1 is very deep with number of layers. At each level, there are 1 × 1, 3 × 3, and 5 × 5 filters. It uses global average pooling at the end of the last inception module and forwards propagation proceeds.

Inception v2 an improvement over version 1, in inception version 2, 5 × 5 convolution is changed into two 3 × 3 convolution filters to improve computational speed. So, placing two 3 × 3 convolutions, in fact, leads to increase efficiency. The following represents the three variants of inception modules in inception v2. The complete inception network is combination of below three modules, hence doing extremely good at feature extraction (Fig. 5).

4.1.2 Regional Proposal Network (RPN)

In RPN, all the images are fed to the pre-trained inception v2 transfer learning model. The output of the model is the decreased size of featured maps. The size of the feature maps depends on the stride of the inception network. The next step in the process is to place bounding boxes on the input image. These bounding boxes point to objects in various sizes. If IoU with ground truth is greater than certain threshold, then prediction is considered as true positive; otherwise, it is labeled as negative. Boxes that are neither positive nor negative are discarded for RPN training.

The region of objects is proposed by the RPN. All identified region proposals will be inputted to the fast RCNN for classification.

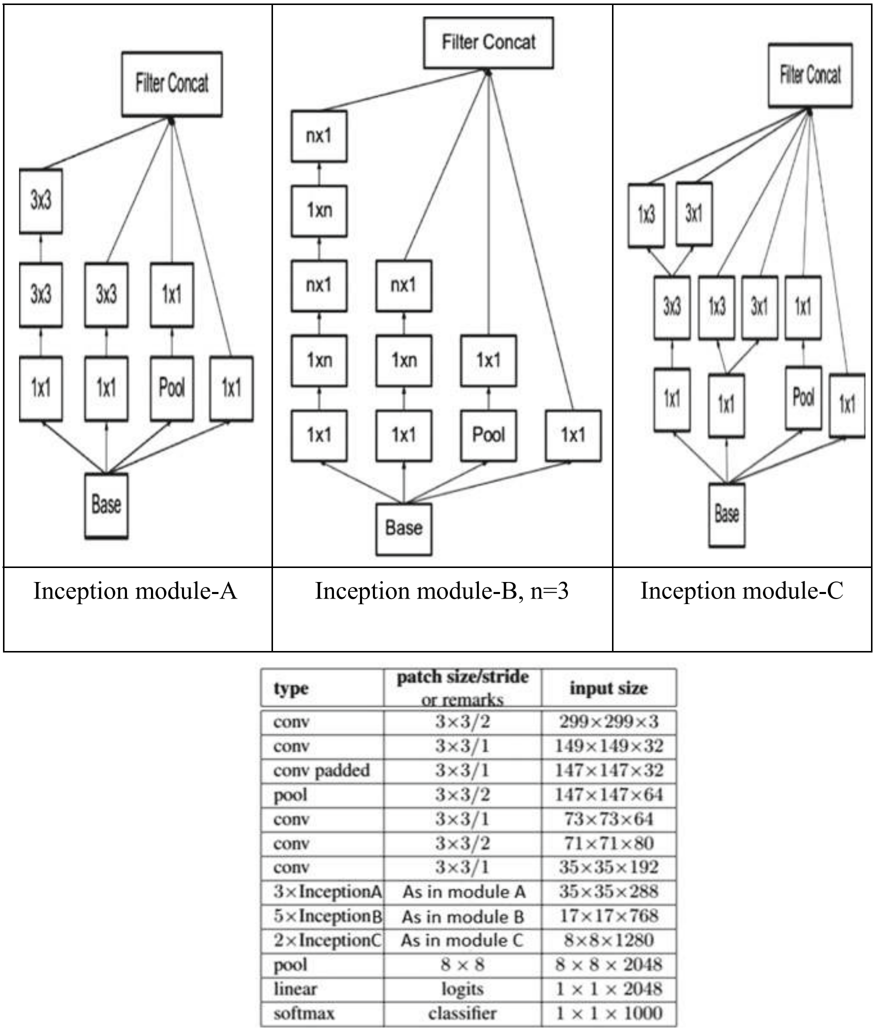
4.1.3 Detector

Fast RCNN process uses the features and region proposal information to classify the objects t in the bounding box, refer Fig. 6. Let us understand the process of the Fast RCNN.

The Fast RCNN detector made of up of a pre-trained inception v2 model, an ROI pooling layer and fully connected layers followed by two layers for classification.

Features are obtained by passing the images through the inception v2. As a next step, pooling is done by the ROI pooling layer. It is a type of pooling layer that performs max pooling on inputs of non-uniform sizes and produces a small feature map of fixed size. The main purpose of doing such pooling is to improve both the training and test time.

Two fully connected layers get the out from ROI pooling layer; the features are fed into the classification and regression layers to classify and localize objects in the images.



Inception module-A

Inception module-B, n=3

Inception module-C

type	patch size/stride or remarks	input size
conv	3×3/2	299×299×3
conv	3×3/1	149×149×32
conv padded	3×3/1	147×147×32
pool	3×3/2	147×147×64
conv	3×3/1	73×73×64
conv	3×3/2	71×71×80
conv	3×3/1	35×35×192
3×InceptionA	As in module A	35×35×288
5×InceptionB	As in module B	17×17×768
2×InceptionC	As in module C	8×8×1280
pool	8 × 8	8 × 8 × 2048
linear	logits	1 × 1 × 2048
softmax	classifier	1 × 1 × 1000

Fig. 5 Inception network version 2

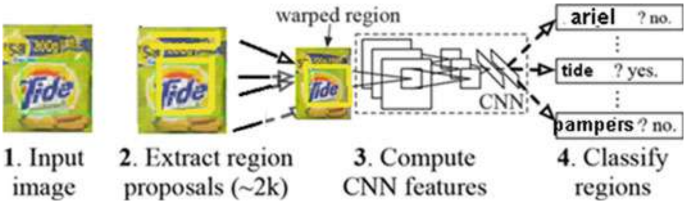


Fig. 6 Faster RCNN classification

Table 1 Performance

Item	Tide	Ariel	Head and shoulder	Pantene	Pampers	Whisper
mAP	80.9	80.2	75.4	76.7	80.4	79.3

5 Methodology

All the objects are to be initially placed at their designated locations. The video stream is given as input to the model and its processes frame by frame. The model records the initial position of the objects and assumes a virtual bounding box around every object that has been detected. On continuous object detection being performed, the system checks if there is any initialized virtual bounding box not intersecting with at least one of the objects that is detected at a given frame. If any object is taken out from its location, there would at least one particular virtual bounding box that remains un-intersected by any of the boxes that bound the other objects, so the system counts it as space and records the count of the objects taken away from their respective positions. This is how we find empty slots. To find the misplaced items, the system keeps count of each class of object that has been detected in each frame and consider all the subsidiary classes to be misplaced objects, for instance, a frame containing four products of type A, two products of type B and one product of type C; on inference, the system would predict that product B and product C are misplaced. Both space and misplaced objects would be precisely identified. The Faster-RCNN is trained for approximately 2000 steps on a custom dataset having over a thousand images of six different retail products with 78.81% mAP on average. We achieve a frame rate of 15FPS on a heavy two-stage Faster-RCNN having inception as a backbone classifier. The regions classified with over 90% confidence threshold are considered, and based on the intersection of the obtained bounding boxes and initialized bounding box, space and misplaced products are identified. One can have a look at the performance details from Table 1.

6 Machine Environment

See Table 2.

Table 2 Machine environment

Hardware and software	Characteristics
Memory	24 GB
Compute	Intel i7 9th Gen, 4.1 GHz × 12
Graphics	Nvidia GTX 1660Ti 6 GB
OS	Windows 10
Developing environment	Python 3.6, TensorFlow 2.1.0

Table 3 Machine environment

Hardware and software	Characteristics
Memory	4 GB
Compute	Intel i3 7th Gen, 1.6 GHz × 4
Graphics	Integrated
OS	Windows 7, Ubuntu 16.04
Developing environment	Python 3.6, TensorFlow 2.1.0

Table 4 Machine environment

Hardware and software	Characteristics
Memory	8 GB
Compute	Intel i5 8th Gen, 1.8 GHz × 4
Graphics	Nvidia GTX 1050 Ti
OS	Windows 10, Ubuntu 20.04
Developing environment	Python 3.6, TensorFlow 2.1.0

Minimum requirements (Table 3):
Recommended requirements (Table 4):

7 Results

7.1 Object Detection

One could see the screenshots presented below with the object detection being carried out. The object detection has been perfect and the accuracy is close to 100% as one could see. The products identified are Whisper, Tide, Head and Shoulders, Pampers, Ariel, etc. Irrespective of the color of the cover, i.e., the wrapper, the object shall be detected and this is real time too. Any size, any color and any shape for the product are completely detected and recognized. One could see the results presented in Fig. 7.

7.2 Misplaced Recognition and Empty Space Localization

Similarly, the second part of the research is involved in the empty space localization and the misplaced object recognition. One could see in below the result screenshot where the misplaced items are properly recognized, i.e., the Pampers is misplaced in the Tide shelve. So, it is to be highlighted. This update can be used by the store keeper or the manager to pick this misplaced item and restore it in a proper location. Also, the number of empty slots in a particular shelve is presented. One could see

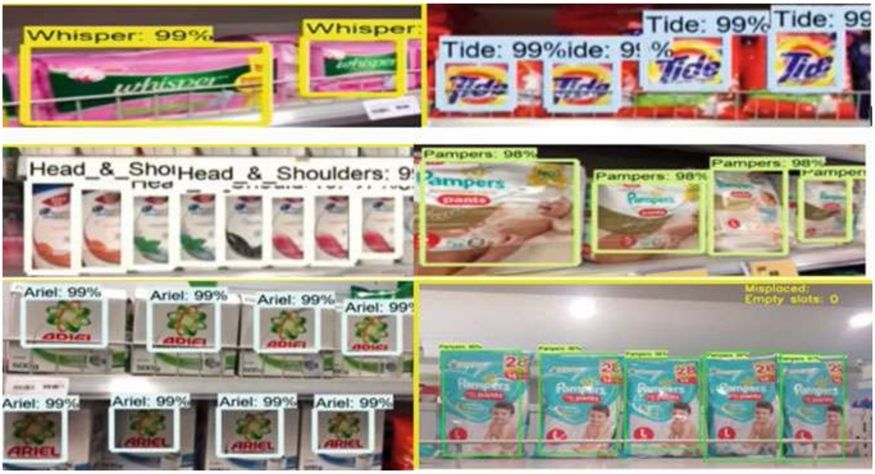


Fig. 7 Faster RCNN object detection and classification

that in the result, screenshots presented as Figs. 8, 9, 10 and 11. With this alert, the storekeeper can refill the shelve as appropriately.

The overall losses are as follows:

- classification loss was found to be 1.01
- localization loss was found to be 0.14
- Total loss was 1.5.

Fig. 8 Empty slots detection and real-time update

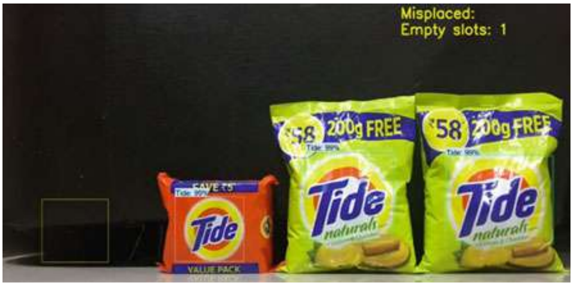
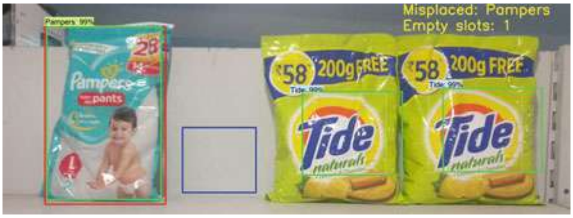


Fig. 9 Misplaced object identification



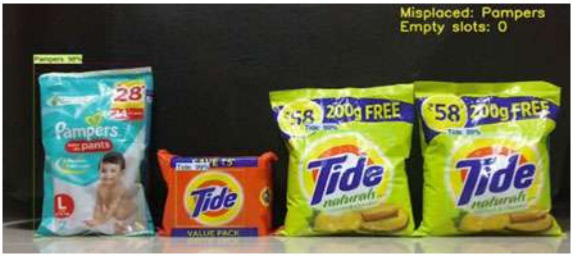


Fig. 10 Misplaced object identification, but no shelf space

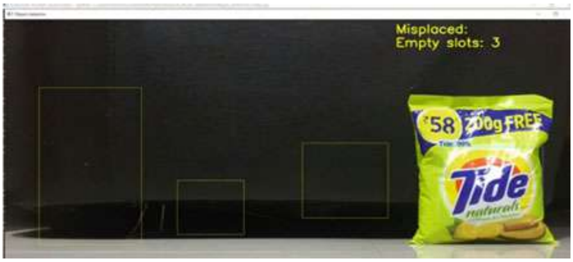


Fig. 11 More empty slots—real-time updates

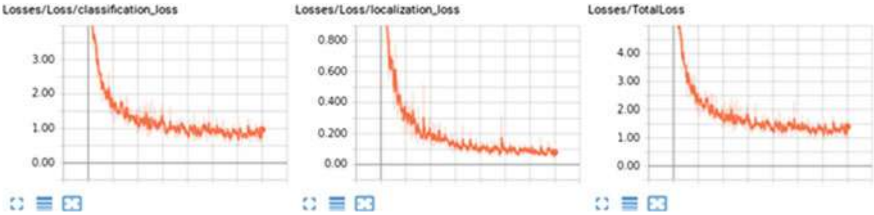


Fig. 12 Losses

One could have a complete look at the losses graph from Fig. 12.

8 Conclusion

The growth of deep learning and related techniques in the recent past is appreciable [4, 5]. The applications developed across various sectors stand as a proof for the growth [6–8]. Authors of this article have explored an application for the retail sector. The amount of time spent shopping is ever increasing and resulting in product misplacements while doing shopping. To address this issue, our system based on advanced computer vision technology, using a two-stage object detection model

FRCNN, can precisely localize empty spaces helping in space optimization and identify misplaced products and warn the corresponding staff so that they can replace that particular product, thus helping in personalized store assortment. The uniqueness of the system is the ease of usage and delivering the results in real time with very minimal latency. The entire system is plug and play and the user just needs to send an input video stream to the model and can expect the model to perform object detection for space localization and misplaced object identification. It can be further ameliorated by training on a more extensive dataset and changing the backbone transfer learning model for latency improvement.

Annexure—A—Abbreviations

CNN	Convolutional Neural Network
FRCNN	Faster Regional CNN
mAP	Mean Average Precision
IoU	Intersection over Union
RPN	Region Proposal Network

References

1. Redmon, J., et al.: You only look once: unified, real-time object detection
2. Liu, W., et al.: SSD: single shot multibox detector
3. Felzenszwalb, P.F., Girshick, R.B., Mcallester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Mach. Intell. **32**(9), 1627 (2010)
4. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. In: ACM MM (2014)
5. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: NIPS (2012)
6. Sung, K.K., Poggio, T.: Example-based learning for view-based human face detection. IEEE Trans. Pattern Anal. Mach. Intell. **20**(1), 39–51 (2002)
7. Wojek, C., Dollar, P., Schiele, B., Perona, P.: Pedestrian detection: an evaluation of the state of the art. IEEE Trans. Pattern Anal. Mach. Intell. **34**(4), 743 (2012)
8. Kobatake, H., Yoshinaga, Y.: Detection of spicules on mammogram based on skeleton analysis. IEEE Trans. Med. Imag. **15**(3), 235–245 (1996)

2D-Image Super-Resolution on Heritage Site

Sheetal Pyatigoudar, S. M. Meena, Sunil V. Gurlahosur, and Uday Kulkarni

Abstract One proposed method for image enhancement is single image super-resolution. For this task, many convolutional neural networks-based models were designed. These convolutional neural networks-based models perform better than the other approaches in quality measurements like structural similarity and peak signal-to-noise ratio (PSNR). Resulting super-resolved image quality is dependent on choice of a loss function. Ongoing work is to a great extent dependent on advancing mean squared reconstruction error. But PSNR and structural similarity values cannot give fine details in an image and provide higher values with unsatisfying quality. Hence, generative adversarial networks model was introduced for this problem in recent years. In this paper, image super-resolution (SR) is done with a generative adversarial network (GAN). It is the first method used for 4 × upscaling factors. Proposed approach calculates loss function which is combination of two loss functions like content and adversarial loss.

Keywords High-resolution image · Generative adversarial network · Super-resolution · Low-resolution image · PSNR · MSE

S. Pyatigoudar (✉) · Meena S. M. · S. V. Gurlahosur · U. Kulkarni
School of Computer Science and Engineering, KLE Technological University, Hubballi, India
e-mail: sheetal.pyatigoudar@gmail.com

Meena S. M.
e-mail: msm@kletech.ac.in

S. V. Gurlahosur
e-mail: svgurlahosur@kletech.ac.in

U. Kulkarni
e-mail: uday_kulkarni@kletech.ac.in

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_34

1 Introduction

If you want to reconstruct any image from its low resolution to high resolution accurately, then it is necessary to have all the high frequency elements from its low-resolution image. In this process, if the image is enlarged beyond a certain limit, a blurred image with no crucial information is obtained as result. One of the main causes behind this problem is hardware limitations of sensors. So, to overcome this problem, super-resolution technique is used.

In image recognition, the difficult task is to deal with low-resolution images. In many applications like military and surveillance, low-resolution images are taken as input and used for recognition. Whenever the image is captured from farther distance, then the quality of image would be a low, and it is unrecognizable. But it is important in some sectors like Department of Defence to have good quality of images. Super-resolution is one of the best methods to overcome this problem. Producing high-resolution image by taking its low-resolution image is known as super-resolution. The number of pixels is increased in single image super-resolution to get the better quality of images while recognition. To preserve the features in an image, super-resolution method is used so that it can be recognized accurately.

Super-resolution has been a point of research and still a progressing procedure in technology. For pixel-based images, many methods are introduced to achieve high-resolution enlargements. Single image super resolution is one of the deep learning methods. Surveillance, medical imaging, forensics, consumer photography and satellite imaging are some of its applications.

Super-resolution is challenging task. For the upscaling algorithms, the main problem is to reconstruct the texture details of the image. For optimization, mean squared error loss (MSE) is chosen. The loss is calculated between pixels of upscaled low-resolution image and original high-resolution image. The image quality measurements like mean squared error (MSE) and peak signal-to-noise ratio (PSNR) are used. In super-resolution algorithms, minimization of MSE and maximization of PSNR are used for evaluation and comparison of algorithms. The reconstructed image with higher measurement values may not be good as compared to the image with lower measurement values. Clarity of the image is good for the image that has low measurement values.

In this work, we focus on a GAN model which is referred as generative adversarial network. This model has a generator which consists of a network with skip-connections, which is called as deep residual network and a discriminator. Generator work is to fool the discriminator. Discriminator must identify the real image. A loss function is calculated using high feature maps of VGG network and with the combination of two losses namely content and adversarial loss. As the result, solution is difficult to distinguish from the original image. The image is upscaled with factor 4.

2 Related Work

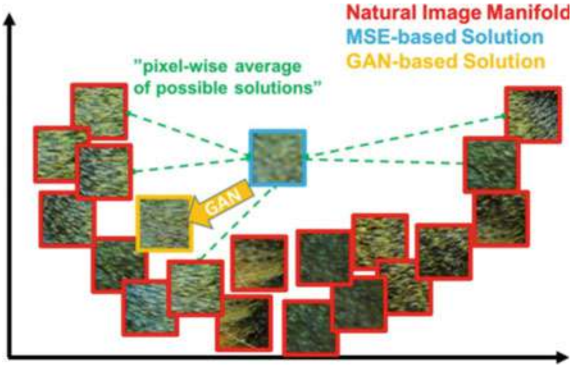
For super-resolution problem, the most convenient way is filtering approach, using linear or Lanczos, or bicubic filters. The idea of these algorithms is based on neighbourhood similarity and is very fast. But they over-simplify the problem and construct over smoothed images is one of the problems.

Recently, one of the methods used for image super-resolution is neural networks. The solutions for these neural networks involve big datasets, like ImageNet so that low-resolution and high-resolution images can be mapped. Some of the models are introduced in [1]. In this paper, they suggested different losses for optimizations and architectures of neural networks. For the purpose of super-resolution, the first step of image processing is upscaling of the image and used in the design of convolutional neural networks. Bicubic filtering is most common method used in this step. The goal of using this approach is to make the neural network to learn a way to "fix" bicubic filtering, and super-resolved image is produced with higher quality. An identical model proposed in [2]. This model uses novel sub-pixel layer so that suggested model is better than the previous models. Behind this design, the main goal is that, during the training process, only the image filtering better than bicubic filtering is learned, and all tasks are done systematically in low-resolution space so no need to perform in high-resolution space.

The final solution would be average plausible solutions if we minimize MSE. Therefore, over smoothed and not perceptually satisfying [1] solutions will be the final results. Illustration of this problem presented in Fig. 1.

Generative adversarial networks can be used to solve this problem. For the process like image generation [3], inpainting, style transfer the GAN's were efficiently used. Images generated using pixel-wise MSE loss along with use of an adversarial loss from discriminator are superior than the images generated by models which are trained to optimize MSE only.

Fig. 1 Averaging of the solutions to produce MSE smooth reconstruction



2.1 Convolutional Neural Networks Design

A specifically planned CNN structure is used for some PC vision issues which is the successful work done by Krizhevsky et al. [4].

The deeper network designs are very hard to train, but they can possibly expand the network's accuracy, and high complexity modelling mappings are allowed. Batch normalization is used to train these deeper network designs and also to balance internal co-variate shift. Another method that is used to train the deep CNNs is with skip-connections and residual blocks which are recently introduced.

2.2 Loss Functions

In order to recover all the lost high frequency details like texture in image is difficult, which is handled by Pixel-wise loss such as MSE. But by minimizing MSE, we can find pixel-wise average of solution that produces poor perceptual quality and smooth solution which is illustrated in Fig. 1 that represents the averaging of the solutions to produce MSE smooth reconstruction.

This issue was handled by the authors Mathieu et al. [5] and Denton et al. [6] by utilizing GANs for image generation applications. The pixel-wise MSE loss was discussed by Yu and Porikli [7]. Yu and Porikli used discriminator loss which is used to train a network and also used large upscaling factor(8x) to super resolve the face images. For unsupervised representation learning, GANs were used. For style transfer and for inpainting, the GAN model is used. In the feature spaces of VGG19 and scattering networks, Bruna et al. [8] minimized the squared error.

In the feature space of neural networks, a loss function which is based on Euclidean distances was calculated by Dosovitskiy and Brox [9] in combination with adversarial training. Superior image generation can be done with help of this proposed loss, and also, problem of decoding nonlinear feature representations is solved. Rather than using pixel-wise error measures, Bruna et al. [8] suggested to utilize the features extracted from a pretrained VGG network. Loss function which is specially designed by this author is on the bases of Euclidean distance. This author calculated loss function between feature maps which are extracted from the VGG19 [10] network. For both artistic style transfer and super-resolution, perceptually more strong outcomes are gained. The result of blending and comparing patches in pixel was explored by Li and Wand [11]. In [12–14] authors represented deep networks used for super-resolution.

2.3 Contribution

We used the GAN model which consists of generator and discriminator. This model produces the image that is very closer to the original images. We calculated loss named as perceptual loss and that involves two losses namely adversarial and content loss. The feature maps of VGG network help in calculation of content loss. This helps to generate more realistic images. Here, we used heritage images for training and testing. The low-resolution image is upscaled with factor 4. We compared images with image quality measurements like PSNR and MSE values. There is increase in PSNR and decrease in MSE value that produces image which is more similar to original image.

Compared the images with CNN model generated images and obtained good results by GAN model generated images.

3 Method

The estimation of high-resolution I^{HR} image from its low-resolution I^{LR} is one of the difficult tasks of single image super-resolution. The high-resolution images are accessible only at the time of training. To optimize network weights at training time, back propagation algorithm utilizes a pair of images (I^{LR}, I^{HR}). By applying a Gaussian filter to I^{HR}, low-resolution (I^{LR}) is achieved from its high-resolution images with factor r, and down sampling operation is performed. A tensor of size $H \times W \times C$ is used to define low-resolution image, and tensor of size $rH \times rW \times C$ is used to define high-resolution image, where C represents the colour channels number.

Estimation of I^{HR} from I^{LR} is carried out by generator function G. A feed-forward CNN G_{θ_G} is used to train the generator and parametrized by θ_G. The $\theta_G = W1 : L, b1 : L$ where W represent weights and b represent biases of L—layer deep neural network. We require specific loss function l^{SR} to produce good reconstruction of I^{HR} and can be reduced using back propagation. To train I_n^{HR}, $n = 1, \ldots, N$ and I_n^{LR}, $n = 1, \ldots N,$ we calculate

$$\theta_G = \mathrm{argmin}_{\theta_G} \frac{1}{N} \sum_{n=1}^{N} l^{\mathrm{SR}}(G_{\theta_G}\left(I_n^{\mathrm{LR}}\right), I_n^{\mathrm{HR}}) \tag{1}$$

We calculated a loss function which is entitled as perceptual loss, and it consists of other two losses like content and adversarial loss.

Discriminator network D_{θ_D} is optimized in different way, with G_{θ_G}. Adversarial min-max problem is solved by using this approach.

$$\min_{\theta_G} \max_{\theta_D} E_{I^{HR} \sim \mathrm{Ptrain}(I^{HR})}\left[\log D_{\theta_D}\left(I^{HR}\right)\right]$$

$$+ E_{I^{LR} \sim PG(I^{LR})}\left[\log\left(1 - D_{\theta_D}\left(G_{\theta_G}\left(I^{LR}\right)\right)\right)\right] \tag{2}$$

The main thought is to train generator model *G* in a such a way that it should fool the discriminator *D*. But the discriminator *D*'s work is to differentiate whether it is super-resolved image generated by G or the original image.

The fundamental thought of the discriminator is to figure out whether the image is real one or it is generated by G that is fake image.

The architecture of generator network involves B residual blocks which is illustrated in Fig. 2. The two convolutional layers which consist of 64 kernels of size 3 × 3 are present in each block. Batch normalization is used for feature maps, and for activation function, ParametricReLU is used. With the use of single or two sub-pixel layers, there is enlargement in resolution of image.

The discriminator network design is illustrated in Fig. 2, and it is trained to differentiate high-resolution images from generated SR images. This model involves eight layers, which has kernels of size 3 × 3. From beginning to an end of the network layer, depth increases with factor of 2 from 64 to 512 kernels which is similar to VGG architecture [10]. To get a probability for sample classification, sigmoid function is used. Two dense layers are present before this function.

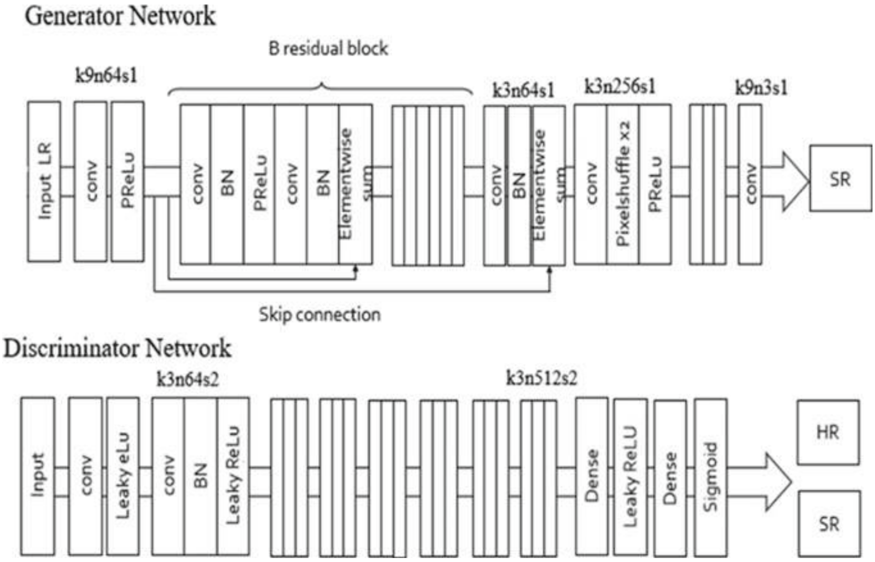


Fig. 2 Architecture of generator and discriminator

3.1 Perceptual Loss Function

Based on the MSE, perceptual loss function l^{SR} is generally modelled and used a loss function that is designed on the bases of perceptually relevant characteristics to evaluate a solution. The weighted sum of an adversarial and a content loss is defined as perceptual loss.

$$l^{SR} = \underbrace{\underbrace{l_X^{\text{SR}}}_{\text{content loss}} + \underbrace{10^{-3 l_{\text{Gen}}^{\text{SR}}}}_{\text{adversarial loss}}}_{\text{perceptual loss(for VGG based content losses)}} \tag{3}$$

3.2 Content Loss

The evaluation of pixel-wise MSE loss is as follows:

$$l_{\text{MSE}}^{\text{SR}} = \frac{1}{r^2WH}\sum_{x=1}^{rW}\sum_{y=1}^{rH}\left(I_{x,y}^{\text{HR}} - G_{\theta_G}\left(I^{\text{LR}}\right)x, y\right)^2 \tag{4}$$

Many methodologies depend on this loss which is used as optimization target for image SR. To achieve high PSNR, solutions of MSE optimization problem cannot provide high frequency details so that unsatisfying results are formed with overly smooth textures.

Rather than using this pixel-wise loss, we utilized a loss function which is most near to perceptual similarity. ReLU activation layers which is present in pretrained 19 layers VGG network is used to represent VGG loss function. In this network, the feature map gained by the j-th convolution is given as $\varphi_{i,j}$. The Euclidean distance between original image I^{HR} and constructed image $G_{\theta_G}\left(I^{\text{LR}}\right)$ is defined as VGG loss.

$$l_{\frac{\text{VGG}}{i.j}}^{\text{SR}} = \frac{1}{W_{i,j}H_{i,j}}\sum_{x=1}^{W_{i,j}}\sum_{y=1}^{H_{i,j}}\left(\emptyset_{i,j}\left(I^{\text{HR}}\right)x,y\right) - \emptyset_{i,j}\left(G_{\theta_G}\left(I^{LR}\right)x, y\right)^2 \tag{5}$$

The components of the respective feature maps are depicted with $W_{i,j}$ and $H_{i,j}$ in VGG network.

3.3 Adversarial Loss

$$l_{\mathrm{Gen}}^{\mathrm{SR}} = \sum_{n=1}^{N} -\log D_{\theta_D}\left(G_{\theta_G}\left(I^{\mathrm{LR}}\right)\right) \qquad (6)$$

The upscaled image is most similar to manifold of natural images by this loss. The probability that upscaled image, which is a natural image, is defined by $D_{\theta_D}\left(G_{\theta_G}\left(I^{LR}\right)\right)$. We will minimize $-\log D_{\theta_D}\left(G_{\theta_G}\left(I^{LR}\right)\right)$ rather than $\log[1 - D_{\theta_D}\left(G_{\theta_G}\left(I^{LR}\right)\right)]$ as discussed in [3] it improves gradient behaviour.

4 Comparison

The images generated with CNN model and the GAN model are compared in Fig. 3. The first image is bicubic, and second image generated using convolution neural network and last one is generated using GAN model. The quality of image generated by GAN model is good compared to other two images. CNN does not provide good results as compared to GAN, and it can be observed that the clarity of image is higher in image generated by GAN.

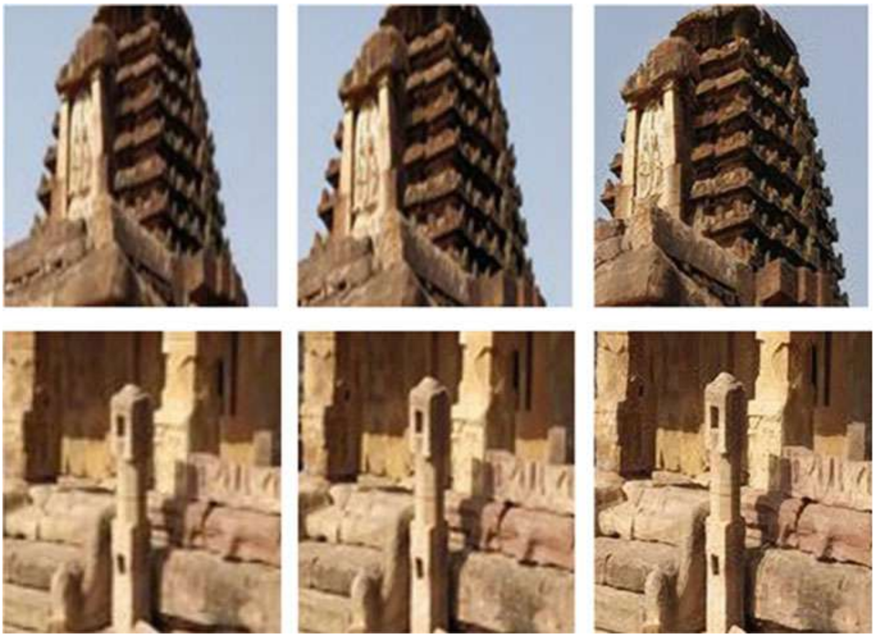


Fig. 3 Comparison with other method

5 Training and Results

Training process involves two models such as generative model *G* and discriminative model *D*. *G* objective is to fool *D*. *D* need to calculate the probability whether sample is from training data or from G. During the training period, both *G* and *D* run in competition with each other, and many times the steps are repeated so that they are improving in their individual jobs after each repetition.

During the training, the two types of images are given, one with high-resolution and another with low-resolution images. Figure 4 represents the high-resolution image, and Fig. 5 represents the low-resolution image. Fig. 6 is set of test images. Here, we have taken Aihole temple images for training and testing.

The training process is carried out with 50 epochs for heritage images. The number of learnable parameters in generator is 1,522,371 which has 16 residual blocks, and the number of learnable parameters in discriminator is 48,166,465 which has eight convolutional layers. The 800 heritage images take nearly 25 h to train, and 1600 heritage images take 50 h to train this model. As there is increase in number of training dataset, then the PSNR value increases, and MSE value decreases as described in Table 1. We have tested for the single image after training 800 images and 1600 images. After training the model, there is increase in PSNR value and decrease in MSE value where these values are used as quality measurement values, so we get the super-resolved image that is too closer to the original image. The values may change based on the given input image. The images obtained by using this model are with good quality and look like the original one.

Fig. 4 High-resolution images

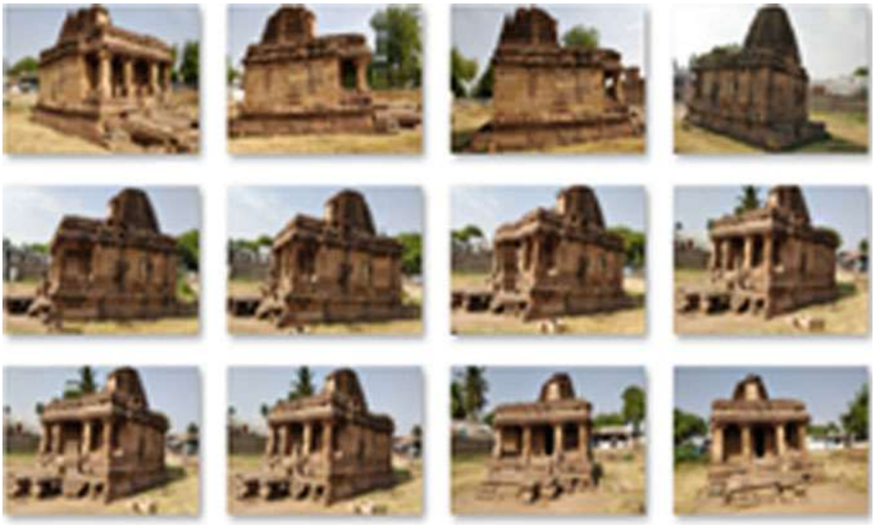


Fig. 5 Low-resolution images

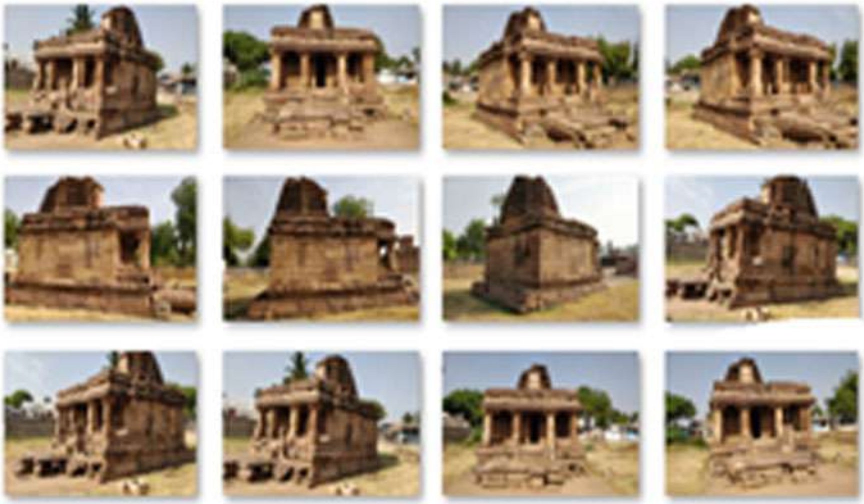


Fig. 6 Test images

The peak signal-to-noise ratio which is represented as PSNR in decibels between two images. Between the original and reconstructed image, this value is used as a quality measurement value. If the PSNR value is higher, then the image quality is better and lesser the value of MSE, then lesser the error.

In Fig. 7, the left image denotes the original high-resolution image, and the right

Table 1 Results obtained

	PSNR		MSE	
S. No.	Dataset (800 images)	Dataset (1600 images)	Dataset (800 images)	Dataset (1600 images)
1	19.17	29.67	511.57	167.64
2	21.04	25.22	399.85	187.36
3	17.5	25.8	551.25	195.40
4	22.11	25.40	786.00	170.04

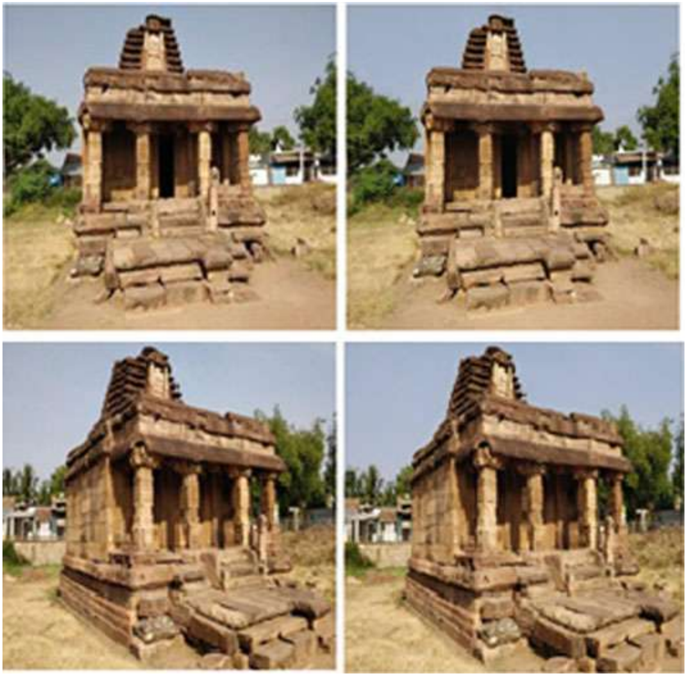


Fig. 7 Left image is the original, and right image is super-resolved [4 × upscaling]

image denotes the super-resolved image.

6 Conclusion and Future Work

To solve the problem of image super-resolution, we applied generative adversarial network model. This model involves two networks which are named as generator and discriminator. Generator generates the fake samples, and discriminator must differentiate between super-resolved image generated by generator and real image.

Generator main aim is to fool the discriminator. Discriminator needs to identify the fake one.

This model also optimizes the perceptual loss function. We calculated a loss function which is entitled as perceptual loss and that is combination of content and adversarial loss. The content loss is calculated with feature maps of VGG network. We have used the Aihole temple images for training. For estimation of super-resolution (SR) images, we used GAN model and obtained good results with high upscaling factors (4 ×).

The future work is to reconstruct structured scenes or text which is an attempt to obtain the clear text.

References

1. Johnson, J., Alahi, A., Li, F.: Perceptual losses for real-time style transfer and super-resolution. CoRR, abs/1603.08155 (2016)
2. Shi, W., Caballero, J., Huszr, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network (2016)
3. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D.: Generative adversarial nets
4. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advancesin Neural Information Processing Systems (NIPS), pp. 1097–1105 (2012)
5. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error. In: International Conference on Learning Representations (ICLR) (2016)
6. Denton, E., Chintala, S., Szlam, A., Fergus, R.: Deep generative image models using a laplacian pyramid of adversarial networks. In: Advances in Neural Information Processing Systems (NIPS), pp. 1486–1494 (2015)
7. Yu, X., Porikli, F.: Ultra-resolving face images by discriminative generative networks. In: European Conference on Computer Vision (ECCV), pp. 318–333 (2016)
8. Bruna, J., Sprechmann, P., LeCun, Y.: Super-resolution with deep convolutional sufficient statistics. In: International Conference on Learning Representations (ICLR) (2016)
9. Dosovitskiy, A., Brox, T.: Generating images with perceptual similarity metrics based on deep networks. In: Advances in Neural Information Processing Systems (NIPS), pp. 658–666 (2016)
10. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556 (2014)
11. Li, C., Wand, M.: Combining markov random fields and convolutional neural networks for image synthesis. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2479–2486 (2016)
12. Nemani, R., Michaelis, A., Li, S., Ganguly, A.: Progressively growing generative adversarial networks for high resolution semantic segmentation of satellite images. Louisiana State University, Northeastern University, NASA (2019)
13. Lim, B., Son, S., Kim, H., Nah, S., Lee, K,M.: Enhanced deep residual networks for single image super-resolution. Department of ECE, ASRI, Seoul National University (2017)
14. Kim, J., Lee, J.K., Lee, K.M.: Deeply-recursive convolutional network for image super-resolution. CoRR, abs/1511.04491 (2015)

Automated Detection of Liver Tumor Using Deep Learning

V. Abhijith, Mable Biju, Sachin Gopakumar, Sharon Andrea Gomez, and Tessy Mathew

Abstract Cancer has been recognized by the World Health Organization as the second leading reason for deaths around the world. With the rise in population, Hepatocellular Carcinoma (HCC) cases have increased due to a lack of early diagnosis and treatment. Conventionally, CT or MRI scans of affected livers undergo manual examination by trained professionals, which usually takes substantial time and effort. With the rising number of cases, this process needs to be sped up. Using deep learning models for medical image segmentation has proven to be an effective method. The proposed approach of deep learning model uses a 2D U-net architecture constructed on fully convolutional network (FCN). The U-net architecture consists of three layers; the contracting/down-sampling, the expanding/up-sampling, and the bottleneck layer which acts as a median between the other two layers. The dataset consists of computed tomography images for training and testing respectively where each scan is in a 3D image format called NIfTI (.nii) and is of variable sizes. Our proposed model is enveloped in application software, where the front end provides a minimalist and intuitive user experience. Using this approach, we received an accuracy of 0.71 using the dice similarity metric. The main benefit of having an application software approach is the ease of adoption in places where such a solution is required to save valuable time and effort.

Keywords Deep learning · U-net · Image segmentation · LiTS · Rectified linear unit (ReLU) · ADAM optimizer

V. Abhijith (✉) · M. Biju · S. Gopakumar · S. A. Gomez · T. Mathew
Mar Baselios College of Engineering and Technology, Trivandrum, Kerala, India
e-mail: mablebj15@gmail.com
URL: http://mbcet.ac.in

S. A. Gomez
e-mail: sharongomez570@gmail.com

T. Mathew
e-mail: tessy.mathew@mbcet.ac.in

APJ Abdul Kalam Technological University, CET Campus,
Thiruvananthapuram, Kerala 695016, India

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_35

1 Introduction

The last three decades saw a massive increase in new Hepatocellular Carcinoma (HCC) [1] cases as a result of population growth and changes in the distribution of ages in the population. The primary reason for this was the lack of early diagnosis and treatment. Conventionally, cancer detection is performed manually by trained professionals with the help of computed tomography (CT) or magnetic resonance imaging (MRI) scans of the liver. In situations where large amounts of liver scans have to be examined, the time and effort required are substantial.

Various cellular and stromal components are composed together to form a tumor. Traditional pathologists use a semi-quantitative grading system for residual tumor burden or report a portion of necrosis indirectly indicating viable tumor burden.

Medical image segmentation mainly focuses on extracting features and help doctors in making a more accurate decision. Figure 1 shows the medical image segmentation. MRI is a popularly used method in the category of radio imaging. The broad variety of image sequences is one of the exceptional characteristics of MRI imaging. In MRI, an image's contrast is based on the phase contrast pulse sequence. MRI also contains information different from that of CT. MRI scans generally take more time and are loud compared to CT scans. There are other techniques to detect liver cancer. We used computer-aided diagnosis to predict the accurate decision for an appropriate therapy.

If we are building a classification pipeline to classify whether a tumor is present or not, our model needs to know what the features of the CT scan to be classified are and it needs to be able to detect those features to generate this prediction. While considering liver tumor segmentation [2], the main focus is on extracting the volume of the liver from the CT scan. One way to solve this problem is to leverage expert knowledge to define those features manually and contour the liver on each slice of the CT scan and use the results of some detection algorithm to do the classification. The problem with this approach is that the contouring process requires more than an hour and the model used has to be invariant to the variations in size and location of the tumor while still being sensitive to the differences that occur in CT scans of

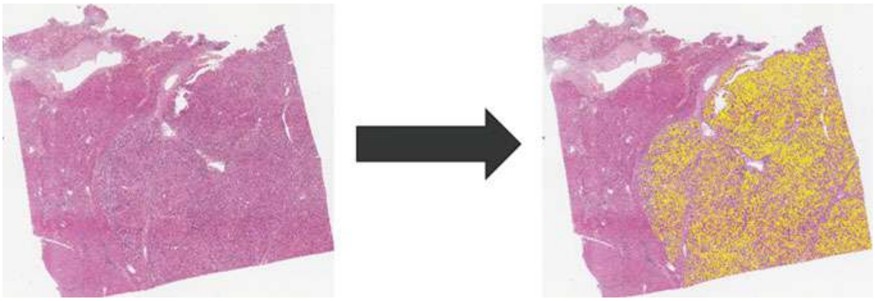


Fig. 1 Pictorial representation of image segmentation

different patients. Even though the pipeline could use these features that the experts define, where this manual extraction will break down is actually in the detection task and that is again due to the incredible variability in visual data, and because of this, the detection of these features is difficult in practice because the detection algorithm would need to withstand each of these different variations. The answer on how to extract features and detect their presence in images automatically in a hierarchical fashion is deep learning; a neural network-based approach can be used to learn visual features directly from data without any manual definition to learn a hierarchy of these features to construct a representation of the image that's internal to the network.

Deep learning techniques are used to build and train the models. They are found to be more accurate than feature-based methods and are beneficial when it comes to the challenging aspects of cancer segmentation as it can be irregular in shape and have ambiguity in terms of boundaries. This method has increased significantly in enhancing the precision of the segmentation of the medical image. So, there is a necessity for accurate segmentation methods and models. Convolution neural networks (CNNs) [3] have achieved great success in detection problems. To consider really how powerful convolution is, grasp that different weight filters can be used to produce distinct feature maps. By simply changing the weights of the filters, one can detect and extract different features present in the input image, this is powerful. Therefore, convolution allows us to capitalize on the spatial structure that is inherent in visual data and use sets of weights to extract local features. For learning features in the application of liver segmentation, CNN is considered to be one of the best approaches. Convolution is the main operation happening on CNN. It is the multiplication of two input functions to produce an output, which is the third function. It is mainly done to extract features from the input image. U-net [4] is a CNN model consisting of layers of convolution, down-sampling layers, and up-sampling layers. The number of down-sampling and up-sampling layers are the same. This work is trying to develop an application software where medical images (in the form of CT) can be fed, and the viable tumor burden cells are shown as output. The effectiveness of U-net in medical segmentation is why it is used in the proposed approach. The detection of cancer cells occurs in the background. The background process makes use of the deep learning technique.

The rest of the paper is divided as follows: Sect. 2 discusses the related work, Sect. 3 discusses the methodology used, followed by the experimental results obtained using the proposed methodology in Sect. 4. Section 5 provides some of the concluding remarks and the intended future work and applications.

2 Related Works

Deep neural network was first used for medical image segmentation wherein a sliding window, the classifier, is applied to each pixel of each slide by extracting a patch around the pixel to segment a whole stack. A disadvantage of this simplistic sliding window approach is that there is plenty of overlap input patches from adjacent pixels and the time-inefficiency of segmenting a whole stack with that approach. Ronneberger et al. [4] reformed the layers that are fully connected as convolutions and pursue the same function with better results. The researchers took a step further on the concept of FCNs [5] and suggested U-Net architecture, where the model had an encoder–decoder network. An input image placed in an encoder architecture will extract a large-level context and further flow into a decoder architecture to recover spatial information and pixel classification results. This is not the first work in the convolution neural network to use encoder–decoder architecture, the researchers added skip connections which connects expanding convolutional layers. The proposed U-Net network performed better in the domain of medical image segmentation. There have been many improvements made on it by several researchers. U-net was modified by Jégou et al. [6] by combining densely connected convolutional networks (DenseNet) with the U-shaped architecture of U-net. They did so by removing the convolutional layer and putting back dense block in the backbone architecture and then extending it to the segmentation of nature's image. This achieved great performance.

Christ et al. [7] used a cascaded fully convolutional neural network followed by dense 3D conditional random fields (CRF) to segment liver and lesions automatically from the abdomen CT. Since they required a high variety and complexity of livers, they used the 3DIRCADb dataset. Pre-processing was done using histogram equalization so that the contrast was increased. Firstly, the liver is segmented in the first FCN; further, its region of interest (ROI) is given to the second FCN which segments lesions, thus forming a cascaded architecture based on U-net which had twofold cross-validation. Post-processing was done by CRF to refine the predicted segmentation. The evaluation using dice was 93.1% by using cascaded architecture alone and it increased to 94.3% by using 3D CRF.

Hans et al. [8] used U-net fully convolutional where each layer uses a 3×3 filter size, followed by the ReLU activation function. Detection is based on liver tumors showing various characteristics like hypodensity and hyperdensity of the lesions. These characteristics show high variability from person to person making manual segmentation challenging. The training was done for 10 epochs using Adam optimizer with a $5e{-}5$ learning rate for updating parameters. To reduce false positives (FP filtering), 46 features were extracted for each tumor lesion using object-based image analysis based on its shape and image intensity statistics. Training datasets were from the LiTS challenge containing 131 contrast-enhanced abdominal CT scans. Their resolution ranged from 0.5 to 1.0 mm. The performance was compared with a medical technical radiology assistant with 10+ years of experience who manually segmented tumors. Evaluation metrics were dice/case, dice/correspondence, merge error, and split error. The network detected 77% of potentially measurable tumor lesions in the LiTS reference.

Li et al. [9] overcomes the disadvantages of using standalone 2D or 3D convolutional networks. This is because 2D convolution does not utilize the spatial information along the third dimension whereas 3D convolution is not computationally and memory efficient. In their proposed approach, the intra-slice features were computed using 2D DenseU-net and the inter-slice features were computed using 3D

DenseU-net. For better liver tumor segmentation, inter- and intra-slice features were computed. Hence, it jointly optimizes hybrid features in Hybrid Feature Fusion (HFF) layer. The LiTS dataset which contained 131 and 70 contrast-enhanced 3D abdominal CT scans and the 3DIRCADb dataset contains 20 venous phases of enhanced CT scans, where 15 volumes have hepatic tumors in the liver, were used for training and testing, respectively. Image pre-processing was done by truncating the image intensity values of all scans to the range of [−200, 250] HU to remove the irrelevant details. The result obtained was 72.2 dice segmentation of lesion, which is a similarity measure of the segmentations, obtained with actual segmentation, and 96.1 for segmentation of liver on the LiTS Dataset. On the dataset 3DIRCADb, a 0.93 + −0.02 dice score was obtained for segmentation of lesion and a dice score of 0.98 + −0.01 for the segmentation of the liver.

Yuan [10] developed a hierarchical model capable of segmenting liver and tumor and also estimating the tumor burden. This framework is based on a deep fully convolutional–deconvolutional neural network (CDNN). A CDNN has only layers that are either convolutional or deconvolutional and learning representations are done using local spatial information that is fed as input. Firstly, a simple CDNN with 19 layers is used to obtain a segmentation of the liver that is coarse quickly. The CT slice size was down sampled to 128 × 128 axially down-sampling and whole image volume was represented with a slice thickness of 3 mm. Since all slices of CT scans were not found to be useful, only slices that were superior or inferior to slice with the liver by a margin of 5 were included. For fine segmentation, CDNN containing 29 layers is applied to the region containing the liver. The region containing the liver was expanded in the x, y, and z-direction by 10 voxels and then the size was adjusted to 256×256. In this phase, all slices in the region of interest were included in the training. In the end, the liver that has been segmented is applied with histogram equalization to enhance it and this is used as an additional input to the second CDNN model boosting the segmentation of the tumor. The loss function is Jaccard distance which is the measure of dissimilarity between two sets when training the models. The LiTS challenge data of 130 scans were used for training. The evaluation was done on 70 test cases provided by the challenge. For the segmentation of the liver, the mean dice similarity coefficient (DSC), which is an index of spatial overlapping, of 0.963 was obtained, for segmentation of tumor 0.657 DSC and tumor burden estimation got 0.017 roots mean square error as results.

Bellever et al. [11] developed a fully automatic technique for segmenting liver and tumor tissues from CT scans using CNN. The network that performs the segmentation of the liver and the lesion contains two cascaded architectures. The main element for lesion and liver segmentation consists of a segmentation network with a detector to locate the lesions. The area of the liver is segmented first with a pre-trained VGG-16. Bounding boxes were placed in the liver; if it overlaps at least 25% with the liver, it was later labeled as positive or negative for training the detector. Dataset was from the LiTS challenge organized by ISBI and MICCAI 2017. They also added a 3D fully connected conditional random field (3D-CRF) as a post-processing step. The segmentation 3-i/o, BP in the liver, detector, 3D-CRF consists of three consecutive slices in the network and backpropagates through liver pixels when training the network to segment the lesion with the detector and 3D-CRF gave the best results among several combinations they tried. The dice score measured in this combination had a score of 0.59.

3 Methodology

3.1 Data

Our dataset is a part of the liver tumor segmentation (LiTS) challenge [12]. This challenge was organized by International Symposium on Biomedical Imaging (ISBI) 2017 and Medical Image Computing and Computer Assisted Interventions (MICCAI) 2017. Seven clinical institutions together provided these CT scans. Among these, 130 and 70 CT scans were used for training and testing respectively. The data of the scans, segmented mask of the image and the viable tumor areas were provided along with this dataset. The tumors have different contrast enhancements like hyper or hypo-dense contrast. The images represent a mixture of pre- and post-therapy abdomen CT scans. The image data is additionally very diverse with respect to the resolution and quality of images. The resolution of images differs from 0.56 to 1.0 mm in axial and 0.45 to 6.0 mm in the z-direction, whereas the quantity of slices in z differs from 42 to 1026. This dataset was provided with annotations referenced by a trained radiologist for the liver and tumor regions. These images were in NIFTI (.nii) format and were of variable sizes.

3.2 Deep Learning Model-U-net

Convolution neural networks [3] are the most frequently used neural network architecture that is used to solve problems with image segmentation. CNN is used for individual classification of each pixel in the image. However, keeping the dimensions of feature maps most of the time will cost a huge memory. After several convolution layers, a down-sampling layer is introduced to decrease the feature map dimension and refine the high-level context. Unfortunately, this can lead to a much-reduced resolution output than the input. Fully convolutional networks or FCNs are one of the many suggested techniques to avoid decreasing output image resolution. It was done by replacing fully connected layers as a sequence of up-sampling layers after convolutional layers. Classical CNNs primarily use a fully connected layer to acquire fixed-length vectors after the final convolution layer and place them in a classifier. FCNs, on the other hand, can accept any input image size. After the last convolution layer, the up-sampling layer can restore its input dimension to the same as the input image. In order to generate a prediction for each pixel, the structural content in the

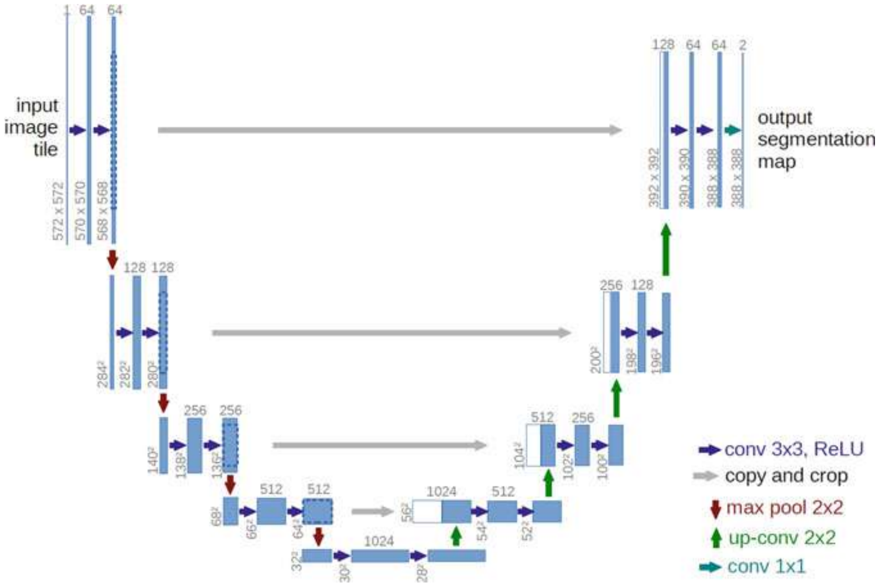


Fig. 2 U-net architecture

initial input image and the pixel-wise classification of the up-sampling feature maps to the anticipated segmentation are maintained.

The U-net architecture was first coined by Ronneberger et al. [4] and is constructed on the FCN and altered to provide better outcomes in medical imaging segmentation and cell detection. U-net has a symmetrical architecture. Figure 2 shows the U-net architecture. The layout is based on the structure of the network encoder–decoder. An input image is placed into an encoder architecture, to obtain the high-level context and then that context is passed to a decoder architecture to restore the outcomes of spatial information and pixel classification. A concatenation operator applies the skip connections between the down-sampling layers and the up-sampling layers. This is done due to the loss of information after each convolution. The network has a huge amount of feature maps in the up-sampling layer due to its symmetry, which allows information to be transferred.

U-Net architecture is divided into three layers.

Contracting/down-sampling Layer The contracting path is divided into blocks. Each block consists of two 3 × 3 convolution layer which is continued with a 2 × 2 max pooling. After each block, the number of feature maps doubles so that architecture can efficiently understand the complicated structures. These contracting layers are designed to extract the context of the input image so that segmentation can be done. Using skip connections, this helpful contextual data is transmitted to the up-sampling layers.

Bottleneck This portion of the network functions as a median between layers that contract and expand. It uses two 3 × 3 CNN layers followed by a 2 × 2 up convolution layer which converts back to original image resolution.

Expanding/up-sampling Layer This portion of the network functions as a median between layers that contract and expand. The up-sampling path is also divided into blocks. The number of blocks in the expanding layer is as same as the number of blocks in the contracting layer. Each of these blocks also consists of; two 3 × 3 convolution layers, deconvolution layer with stride 2 and concatenation from the contracting path with the respective cropped feature map. This expanding route aims to attain an accurate location from the contracting path associated with contextual information. The segmented image output will have two channels. One is taken as the foreground, the other as the background.

3.3 Implementation

The proposed approach employed a modified 2D-U-net fully convolutional network architecture. Even though the CT scans were provided in 3D format, using a 3D-U-net model would require high computational and hardware facilities. Therefore, a 2D-U-net model was leveraged for this work. Moreover, these CT scans were not converted 2D images as this would lead to severe loss of information.

Figure 3 depicts the workflow of the proposed approach. The dataset consists of CT scans, which includes abdominal CT scan along with the segmented ground truth. For the pre-processing phase, the NiBabel library CT scans were loaded easily and later images were converted to NumPy (.npy) array. Hence, the processing was faster compared to the previous approaches, so this was a necessary step in pre-processing and aided in obtaining better results. The resized images had a dimension of [256, 256, 1].

After pre-processing, the dataset is divided into train, test, and validation sets for further processing. The 2D-U-net model is trained with this training dataset where the initial creation of the model takes place. Model training was carried out by using

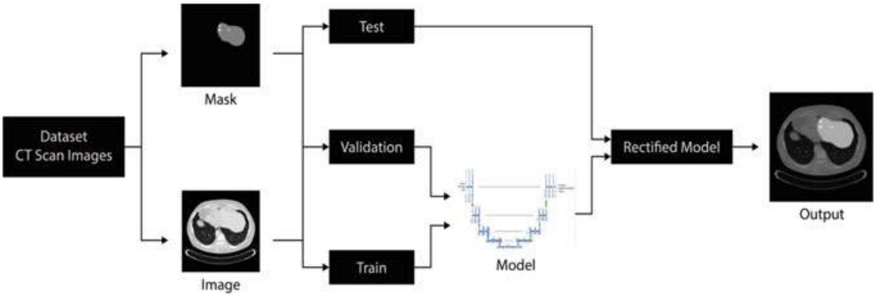


Fig. 3 Data flow graph

90 CT scans. Once the model was trained, it was then evaluated on the validation dataset of 10 CT scans to refine the model. Finally, this rectified model is used to detect tumors using the test dataset which consists of 30 CT scans and its performance was measured.

Firstly, these CT scans were resized to a format of [64 × 64 × 1] for faster processing and for learning the dataset. However, the dice score obtained was too low. For further training, scans were then resized to [128 × 128 × 1]. The training of these two methods was done in the Intel i7-7th gen processor along with 32 GB RAM and NVIDIA GTX 1080 GPU. To improve the results, training was further performed on a method that used [256 × 256 × 1] as the image size, which gave a decent result.

3.4 Training and Validation

The backend of this approach is a Python code (Python 3.7 with Keras 2.0 framework for deep learning). The model used for training the dataset was a 2D-U-net with nine convolutional layers. 90 scans were used for training the model and 10 scans for validation, followed by 30 scans for testing purposes as only 130 scans had ground truth. The training was performed in three batches for 30 epochs each for 6 h on an Intel i9-9900K processor along with 32 GB RAM and NVIDIA RTX 2070 Super GPU.

The front-end application was implemented as a Windows Presentation Foundation (WPF) application written in C# and XAML using Microsoft Visual Studio. The application allows the user to browse their local directories for a CT scan in NIfTI format and perform the prediction on it by clicking the “analyze” button, which runs the Python script for the prediction process in a new process thread. The application waits until the process terminates and retrieves the obtained result.

4 Experimental Results

The evaluation metrics used for this method were dice similarity coefficient [13], which is one of the most commonly used evaluation methods in medical image segmentation. It is a spatial overlap index that varies from 0 to 1. The value is 0 when there is absolutely no overlap and it is 1 when there is a complete overlap of the binary segmentation results. Hence, the average dice score per volume is used, and thus, it shows better results in the case of medical image segmentation compared to dice global which combines all datasets into one for evaluating. It is given by the equation:

$$\text{Dice} = \frac{2|X \cap Y|}{|X| + |Y|} \tag{1}$$

Table 1 Comparison of Dice scores of various existing works

Models	Dice score
FCN+3D CRF [7]	0.943
U-net+RF classifier [8]	0.96
HDense U-net [9]	0.961
CDNN [10]	0.963
Segmentation 3-i/o + BP in liver + detector + 3D-CRF [11]	0.59
2D U-net	**0.71**

Table 2 Dice score evaluated in various approaches

Models	Dice score
2D Unet-64	0.37
2D Unet-128	0.53
2D U-net-256	**0.71**

where X and Y are two sets and |X|and |Y|are the number of elements in each of the two sets.

The 2D U-net in this method was analyzed without using a pre-trained model. Table 1 shows the scores of various existing works. These results indicate that this proposed approach shows promising scores when compared with the existing works in this field. Therefore, this method can be used to extract liver segments from raw CT scans.

During the first phase of implementation, the processing was done on images after resizing it into [64 × 64 × 1]. However, this yielded a very low prediction score, since the images had a severe data loss due to the resizing. Table 2 shows the predictions made at each phase. Further, the processing was performed on images after preprocessing it to a size of [128 × 128 × 1]. This showed a slight improvement in the prediction score. The accuracy got improved from 0.37 to 0.53, which was better, but it was indeed lower than the existing works in this field. Lastly, the input image of size [256 × 256 × 1] was used and an accuracy of 0.71 was obtained. Figure 4 shows the predictions made by the 2D U-net 256 model. This can also be resized to [512 × 512 × 1] and to different 3D resolution, but it requires a lot of time and hardware resources.

Figure 5 shows a graph plotted using matplotlib between the number of epochs in the x-axis and the dice coefficient in the y-axis. The blue and orange lines indicate the train and validation respectively. It shows the dice score at a particular epoch. This model was trained for 30 epochs in three batches each.

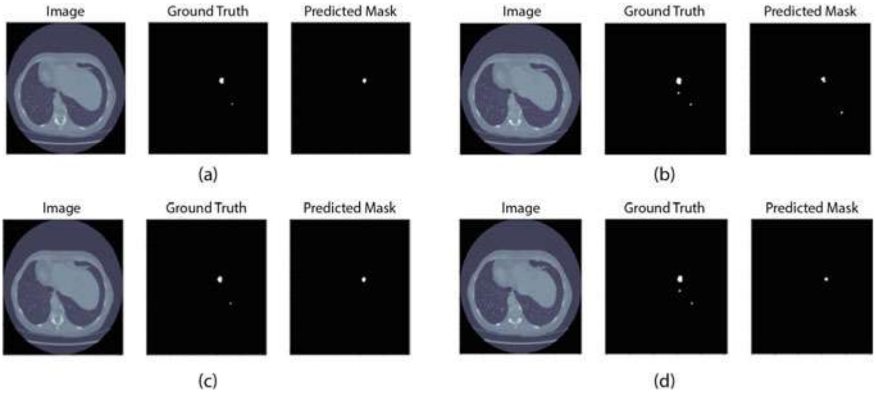


Fig. 4 Predictions made by the 2D U-net-256 model

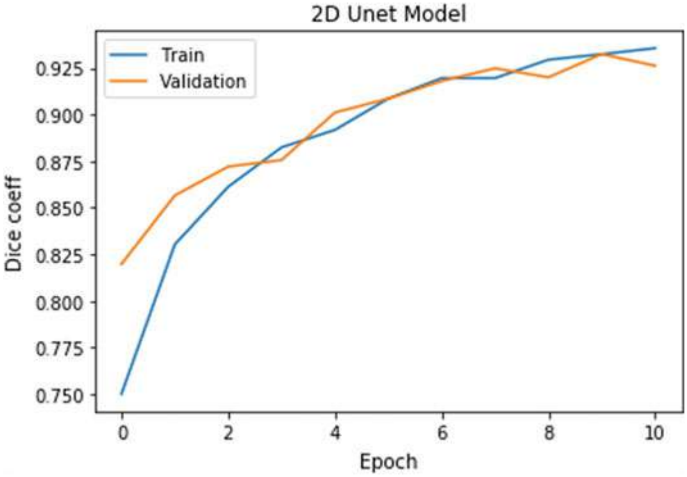


Fig. 5 Graph plotted between Dice and epochs

5 Conclusion

It is evident from the preceding discussions that deep learning is an effective means of automating and speeding up the detection of liver tumors from CT scans. The proposed deep learning model achieved this with an accuracy of 0.71. The prediction process performed by the model takes far less time and effort than a trained professional examining the scans manually. Using a U-net architecture for the model allowed the dataset to be relatively small. The application software approach encapsulates the model behind a simple user interface to obliviate the user from all that goes on in the prediction process. This approach makes it possible to perform the prediction easily with only the click of a button. The intuitive user experience will help in widespread adoption by institutions. In hospitals where it is needed, valuable time and effort that would have otherwise gone to the manual examination can be saved. It can be distributed commercially to hospitals where staffing may not be sufficient.

The dataset used was still quite large and required a higher-end device. With the hardware at our disposal, we were only able to train the 2D U-net model using CT scans from the LiTS dataset resized to a resolution of 256 × 256, which is a quarter of the original resolution of the scans in the dataset. Even though the 2D U-net model described in this paper is promising, there is further room for improvement by training the proposed model using higher resolutions and dimensions and/or a greater number of CT scans on hardware that is more powerful than what we used, to potentially improve the accuracy of prediction and also to decrease human errors. While the application software should work on most modern low-end PCs, it is not guaranteed to work perfectly without performance issues in heavily outdated PCs that may still be in use by some hospitals. Further optimization is needed to make this a truly accessible product, which is our next phase for working on this project.

Acknowledgements Our project received a financial assistance from Kerala State Council for State, Technology and Environment (KSCSTE), as a part of their Student Project scheme.

References

1. Forner, A., Llovet, J.M., Bruix, J.: Hepatocellular carcinoma. Lancet **379**(9822), 1245–1255 (2012)
2. Campadelli, P., et al.: Liver segmentation from computed tomography scans: a survey and a new algorithm. Artif. Intell. Med. **45**(2–3), 185–196 (2009). https://doi.org/10.1016/j.artmed.2008.07.020
3. O'Shea, K., Nash, R.: An Introduction to Convolutional Neural Networks. ArXiv e-prints (2015)
4. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), pp. 234–241 (2015)
5. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3431–3440 (2015)
6. Jégou, S., Drozdzal, M., Vazquez, D., Romero, A., Bengio, Y.: The one hundred layers tiramisu: fully convolutional DenseNets for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 11–19 (2017)
7. Christ, P.F., Elshaer, M.E.A., Ettlinger, F., Tatavarty, S., Bickel, M., Bilic, P., Rempfler, M., Armbruster, M., Hofmann, F., D'Anastasi, M., Sommer, W.H.: Automatic liver and lesion segmentation in CT using cascaded fully convolutional neural networks and 3D conditional random fields. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 415–423. Springer, Cham (2016)
8. Chlebus, G., Schenk, A., Moltz, J.H., van Ginneken, B., Hahn, H.K., Meine, H.: Deep learning based automatic liver tumor segmentation in CT with shape-based post-processing. In: 1st Conference on Medical Imaging with Deep Learning (MIDL 2018)

9. Li, X., Chen, H., Qi, X., Dou, Q., Chi-Wing, F., Heng, P.-A.: HDenseUNet: hybrid densely connected UNet for liver and tumor segmentation from CT volumes. IEEE Trans. Med. Imaging **37**(12), 2663–2674 (2018)
10. Yuan, Y.: Hierarchical convolutional-deconvolutional neural networks for automatic liver and tumor segmentation. arXiv preprint arXiv:1710.04540 (2017)
11. Bellver, M., Maninis, K.K., Pont-Tuset, J., Giró-i-Nieto, X., Torres, J., Van Gool, L.: Detection-aided liver lesion segmentation using deep learning. arXiv preprint arXiv:1711.11069 (2017)
12. Bilic, P., Christ, P.F., Vorontsov, E., Chlebus, G., Chen, H., Dou, Q., Fu, C.W., Han, X., Heng, P.A., Hesser, J., Kadoury, S.: The liver tumor segmentation benchmark (liTS). arXiv preprint arXiv:1901.04056 (2019)
13. Yeghiazaryan, V., Voiculescu, I.: An Overview of Current Evaluation Methods Used in Medical Image Segmentation. Department of Computer Science, University of Oxford (2015)

Breast Mass Classification Using Classic Neural Network Architecture and Support Vector Machine

R. Priya, V. Sreelekshmi, Jyothisha J. Nair, and G. Gopakumar

Abstract According to WHO, the most dangerous disease prevailing among women is breast cancer. It is among one of the diseases that is untraceable in the beginning. About 1 in 8 women suffer breast cancer and even results in the removal of their breast. In this domain, a novel experiment to classify breast cancer using convolutional neural network and fuzzy system is introduced. A combination of convolution neural network and fuzzy system has been devised for grouping similar masses of benign and malignant in mammography database based on the mass area in breast. The mammography images are taken for image enhancement and image segmentation for identifying the mass area and the classic neural network architecture (Alexnet) performs the feature extraction. After that it is followed by fuzzy system for finding how much denser the malignant or benign cancer is. A well-known classic neural network architecture AlexNet is employed and is fine tuned to group similar classes. The fully connected (fc) layer is replaced with support vector machine (SVM) to improve the classification effectiveness. The results are derived using the following publicly available datasets: (1) digital database for screening mammography (DDSM), (2) curated breast imaging subset of DDSM (CBIS-DDSM) and (3) mammography image analysis society (MIAS). Data augmentation is also performed to increase the training samples and to achieve better accuracy.

Keywords Breast cancer · Mammogram · Deep learning · Fuzzy membership functions

R. Priya · V. Sreelekshmi · J. J. Nair (✉) · G. Gopakumar
Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Amritapuri, India
e-mail: jyothishaj@am.amrita.edu

R. Priya
e-mail: priyar@am.students.amrita.edu

V. Sreelekshmi
e-mail: sreelekshmiv@am.amrita.edu

G. Gopakumar
e-mail: gopakumarg@am.amrita.edu

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_36

1 Introduction

Breast cancer is one of the dangerous diseases prevailing among women today. Breast cancer is mainly due to the growth of tissues in the breast. Around 55,176 breast cancer cases are reported in UK between 2015–2017, 1199 deaths due to breast cancer, 78% women survived breast cancer in England and wales between 2010 and 2011 and around 23% of the preventable cases are reported in UK in 2015 [1]. The rate of breast cancer is found to be increasing and the only way to prevent the breast cancer is to detect the breast cancer as early as possible and undergo treatment.

There are many symptoms for breast cancer. The main symptoms include lumps, nipple discharge, dimbling, breast or nipple pain, nipple retraction or invasion, redness, changes to the skin texture, lymph node changes and swelling [2]. For detecting malignant cancer, main symptoms are excessive spreading and irregular in shape. The rate of growth is often rapid, invasive, infiltrating and destroying the surrounding normal tissue. The rate of spreading is also very fast. Benign cancers are mainly oval or circular in shape and it would not spread very fast.

The mammography images are the main sources for detecting breast cancer. Mammography images give us an overview about the mass area, how much is the cancer spreading and visualize the cancer very easily. It helps the doctors to understand how the cancer is spreading, its type and how to cure it. Mass area in the breast cancer is the main source of information. Based on the mass area, we can predict which type of cancer effected the patient and how much it is spread and how serious the cancer is.

In some cases, we are unable to detect the relevance of the cancer, for that fuzzy logic employed to detect the degree of the cancer or the measure of the density of the cancer. The fuzzy logic is mainly applied to give an output as yes or no. Here in this case, it is employed to detect the measure of the dense area in the breast if the mass area corresponds to any of the described class of cancer. Fuzzy logic is often used to give answer in uncertain situation.

In this paper, three main powerful concepts are employed. The deep convolution neural network is used for feature extraction, the support vector machines (SVM) is used for classification and the fuzzy logic is used to give the measure of the density of the mass area in the breast.

The rest of the paper is mainly organized as follows: Sect. 2 presents the related work. Section 3 includes the architecture of the proposed work. Section 4 discusses the experiment and result and finally Sect. 5 concludes the proposed work.

2 Related Works

A breast CAD method based on fusion deep features is proposed in the work [3]. The main idea of this fusion is to extract the deep features from the convolutional neural network (CNN) architecture to apply that in mass area detection and mass area diagnosis. In the mass detection, method based on sub-domain CNN deep features and US-Extreme Learning Machine (US-ELM) clustering is developed. The ELM classifier is utilized to classify the malignant and benign masses using a feature set, fusing deep features, morphological features, texture features and density features.

The Wisconsin breast cancer database is analyzed and various machine learning algorithms like naive Bayes, SVM, logistic regression, K-nearest neighbours (KNN) , random forest neural network and CNN classifiers [4] are applied. They deeply pointed out the performance of different deep neural networks on the database. The main observation they pointed out in deep neural network is that they found that the convergence time increases significantly and its getting harder to optimize the network. With three hundred feature map, they got the best result.

The work [5] incorporated ResNet-50 architecture, Class Activation Map techniques to classify and detect malignant and benign tumor masses. Two main steps it includes are: one is classify into normal or non cancerous and second is: localize via CAM depending on first stage.

The work focuses [6] on classifying malignant and benign masses. The method is CAD. The segmentation was based on two types. One is ROI region of interest (ROI) segmented manually and ROI extracted using threshold techniques. The feature extraction is performed by the deep convolutional neural network (DCNN). Alexnet architecture is employed here. Transfer learning is applied to the alexnet and it classifies the images into two classes, one is malignant and the other is benign. The fully connected layer is replaced with SVM classifier to achieve better accuracy. The accuracy of alexnet on ROI extracted using manually is higher by 1.8% compared to second using the DDSM dataset.

Using slide images of blood smear, the work [7] focuses on detecting malaria and classifying leukemia cell lines are addressed. The work follows the pattern of starting with theory behind multilayer neural networks with backpropagation followed by motivation, theory and concluding with CNN model.

The manuscript [8] investigates the effectiveness of transfer learning utilizing certain ensemble methods and fine tuned features from deep neural networks for grouping similar leukemia cell lines. Various deep learning methods are under investigation on this domain of this work. The inference drawn out is that computerized techniques plays a crucial role on cyto-pathological testing under restricted conditions.

In the paper [9], the work incorporated supervised machine learning methods added with ensemble techniques to introduce a model for breast tumor prediction. Various other predominant techniques like support vector machines, Bayesian classifier, KNN, etc. are taken into consideration.

There are many factors contributing to breast cancer occurrence. The various key variables are very difficult to trace out, yet the factors have high significance in detecting the breast masses. The field machine learning added with regular diagnosis can help to decide the occurrence of breast cancer. All the features are not taken into consideration only relevant ones are. So feature selection procedures are added to it. The impact of various feature selection techniques on accuracy of available machine learning algorithms [10] was also discussed by many researchers.

3 Proposed Method

3.1 Dataset Collection

In this paper, the first dataset used is mammogram images obtained from the MIAS [11] dataset. MIAS dataset is developed by a group of interested researchers for understanding and observing breast cancer among women. MIAS dataset contains 312 images from 161 patients including both right and left breast images. The dataset contains mammography images carefully identified by the radiologists. In the MIAS dataset, the mammogram images originally were of the size of 1024 × 1024 pixels.

The second dataset used is mammogram images obtained from the DDSM [12] dataset. The digital database for screening mammography (DDSM) is widely used by the research community. The DDSM is a database of 3760 grayscale mammography images. It contains benign and malignant cases with verified medical deviation information. It contains 1257 full mammography images, 1246 region of interest extracted (manually) images, 1257 region of interest extracted (threshold) images.

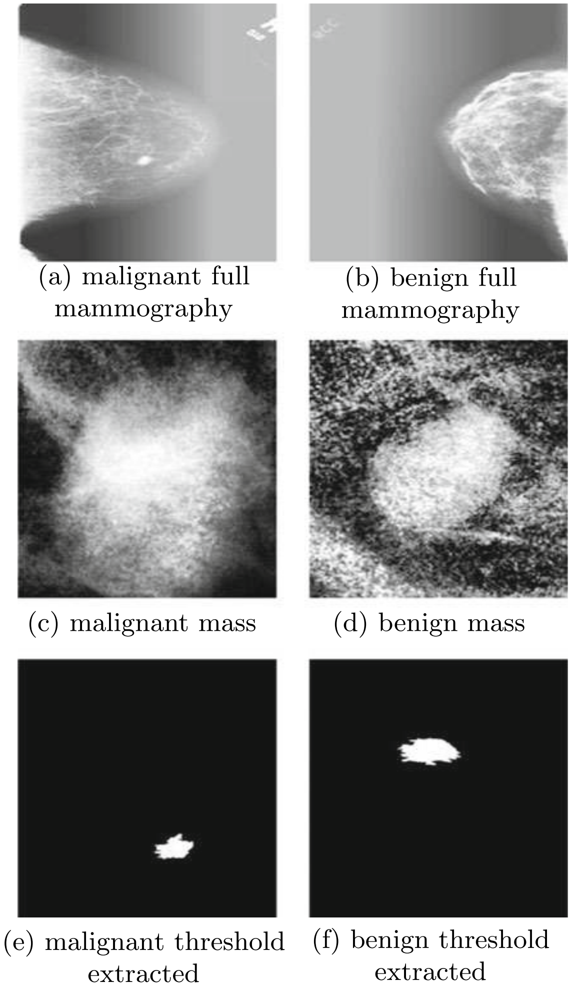
The third dataset used is the curated breast image dataset (CBIS-DDSM). It is the advanced version of DDSM dataset where the cancerous areas are extracted by the professionals for the future research. It contains 3760 grayscale images. The malignant cases full mammogram images, malignant mass of it and the region of interest (extracted by threshold based technique) shown in Fig. 1a, c, e respectively, and also, the benign cases for the same is shown in Fig. 1b, d, f, respectively.

3.2 Dataset Preprocessing

Image enhancement is a process of enhancing the images for future image analysis. Several computer vision techniques have been used in this paper. The mammography images are preprocessed using:

- Image adjustment: The mammogram images are adjusted using image adjustment technique. Some contrast adjustment is done to enhance the image.
- Histogram equalization: The histogram equalization is used for contrast adjustment in images. The contrast adjustment is done using the image histogram.
- Contrast limited adaptive histogram equalization (CLAHE) algorithm : It is similar to histogram equalization but with some enhancement processing in histogram. Most of the images obtained for image analysis are of washed out or images with contrast are very low. So to enhance those images, we use CLAHE by spanning the histogram in its entire range.

The proposed approach is summarized in Fig. 2.


(a) malignant full mammography

(b) benign full mammography

(c) malignant mass

(d) benign mass

(e) malignant threshold extracted

(f) benign threshold extracted

Fig. 1 CBIS-DDSM dataset

3.3 Feature Extraction

Feature extraction is the main tool used for extracting important features from an image. In recent years, deep convolution neural network (DCNN) emerged as an effective tool for extracting features from an image. The main components of a deep convolution network are the (i) convolution layers, (ii) pooling layers and (iii) fully connected layers. There are two categories of architectures for convolution neural network, one is classic network architectures and other is the modern network architectures. The classic neural network architectures include LeNet-5 architecture, AlexNet architecture and VGG-16 architecture. The modern neural network archi-

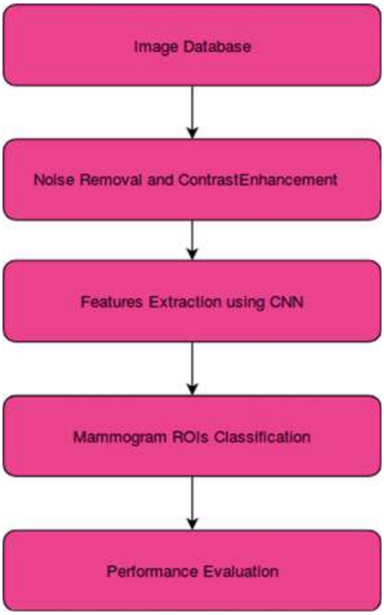


Image Database
Noise Removal and ContrastEnhancement
Features Extraction using CNN
Mammogram ROIs Classification
Performance Evaluation

Fig. 2 Proposed method

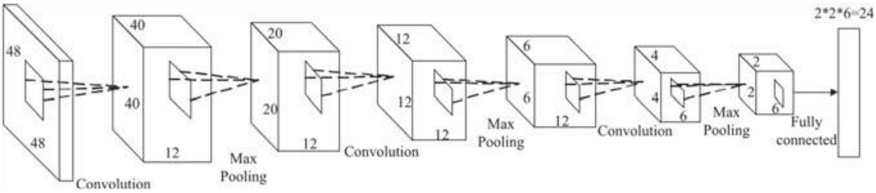


Fig. 3 CNN architecture for mass detection

tecture includes Inception network, ResNet architecture, ResNeXt architecture and DenseNet architecture. The network used here is AlexNet architecture.

In this paper, CNN is used to extract deep features from the preprocessed images. Figure 3 presents an AlexNet architecture. Table 1 shows the layers, kernel size, strides, padding, activations, etc. used by the AlexNet architecture.

3.4 Classification

In this work, the classification is done by support vector machines (SVMs). The support vector machines are effective classifiers. SVMs are supervised learning models which separates the data points using suitable hyperplanes. The SVM is added in

Table 1 Model architecture

Input image (224, 224, 3)

Layers	Kernels size	Strides	Padding
Convolution layer 1	(11, 11)	A(4, 4)	‘valid’
Max pooling	(2, 2)	A(2, 2)	‘valid’
Convolution layer 2	(11, 11)	(1, 1)	‘valid’
Max pooling	(2, 2)	A(2, 2)	‘valid’
Convolution layer 3	(3, 3)	(1, 1)	‘valid’
Convolution layer 4	(3, 3)	(1, 1)	‘valid’
Convolution layer 5	(3, 3)	(1, 1)	‘valid’
Flattening			
1st fully connected layer	Dense (4096)	‘relu’	‘null’
2nd fully connected layer	Dense (4096)	‘relu’	‘null’
Dropout(0.4)			
3rd fully connected layer	Dense (1000)	‘relu’	‘null’
Dropout(0.4)			
Output layer	Dense (3)	‘softmax’	‘null’

the alexnet model using the hinge loss function and l2 regularization. The activation function used in the fully connected layer is replaced with activation function suitable for SVM classification. In this paper, we use SVM with different kernels.

3.5 Transfer Learning

Transfer learning is a technique of using only last fully connected layers and keeping upper layers untrained. The ImageNet pretrained alexnet model is used in this paper. So here, the last three layers of alexnet are trained to classify two classes of images mainly malignant and benign. The upper layers are kept untrained.

3.6 Fuzzy System

In this paper, fuzzy system is used to estimate the density of the cancerous region. Membership functions are used to estimate the severity and density. Density is estimated using density membership functions. The severity of the cancer is estimated using severity membership function. The severity of the cancerous area is estimated by counting the number of white pixels (in roi extracted using threshold), and using the severity membership function, we can find out how severe the cancer is. The rules can be defined as:

rule 1: if density is poor, then severity is low
rule 2: if density is average, then severity is medium
rule 3: if density is good, then severity is high.

4 Experiment and Results

The performance evaluation of the newly developed medical imaging CAD is a significant task which tells us whether the developed system is an improvement over existing systems or not. To evaluate our experiment, ROC, confusion matrix, density membership function and cancer severity results are used.

In this paper, the convolutional neural network(CNN) used for feature extraction is AlexNet architecture. The observed accuracy for AlexNet with SVM is 85%. We have used the precision, recall and accuracy measures as the evaluation metric for mammogram classification shown in Table 2.

Precision or specificity is the fraction of the number of true positive predictions divided by the total number of true and false positives in the set shown in Eq. 1. Recall also called sensitivity for binary classification is the total number of true positive predictions divided by the total number of true positive and false negative in the set shown in Eq. 2. The accuracy is also calculated here as the proportion of correct predictions (both true positives and true negatives) among the total number of breast cancer cases examined shown in Eq. 3. In these equations, the TP is true positive , FP is false positive, TN is true negative and FN is false negative.

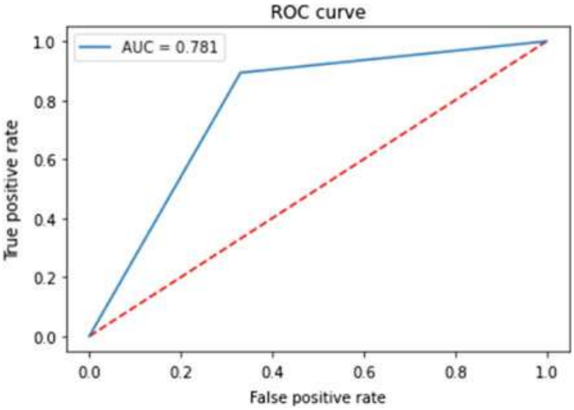
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

Table 2 Precision, recall and accuracy

Metrics	Alexnet	Resnet
Sensitivity	89	67
Specificity	81	87
AUC	52	62
Accuracy	85	78

Fig. 4 ROC curve



$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{TN} + \text{FP}} \tag{3}$$

By setting various cutoffs, the performance measurement of a classification task is measured using the area under the curve-receiver operating characteristic (AUC-ROC) curve. A probabilistic curve is given by ROC and a separability measure is given by AUC. The model has high AUC means its classification of classes is efficient. The AUC measures the entire two-dimensional area underneath the entire ROC curve which is shown in Fig. 4. The x-axis is plotted with FPR and TPR on y-axis. AUC = 1 means the model is perfect. Our model gives an AUC of 0.781.

The confusion matrix has been obtained from the testing part. In this case, for example, out of 100 actual malignant images, 10 images was classified as normal. The confusion matrix without normalization and with normalization is shown in Figs. 5 and 6, respectively.

From Table 3, we observed that SVM with linear kernal outperforms the gaussian kernel. The linear kernel gives better results as compared to that of gaussian kernal and clearly explains the accuracy changes when the two kernals are applied for all the four different datasets. Table 4 we observed that Alexnet with SVM gives high accuracy on combining mias and ddsm dataset. From this, we can understand that when we combine two different datasets the model can perfectly learn the various features and give high accuracy.

Figure 7 corresponds to the input density. The scale chosen here is 0–10. It shows the density membership function. The density contains values poor, average and good.

Figure 8 corresponds to the input severity. The scale chosen here is 0–26. The severity contain values low, medium and high. Rules defined here are:

Fig. 5 Confusion matrix without normalization

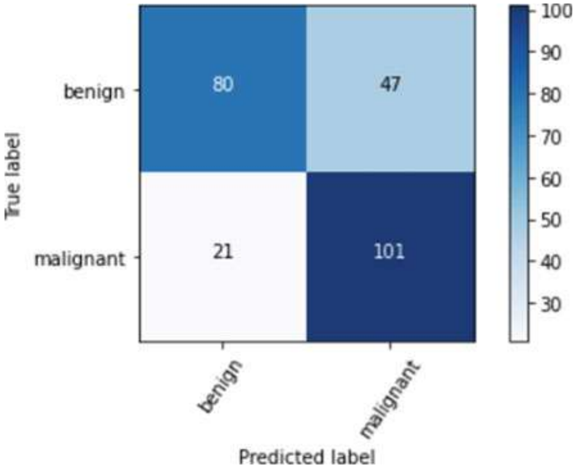
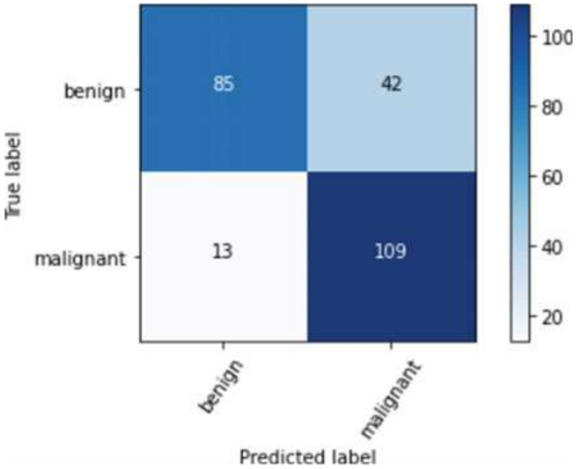


Fig. 6 Confusion matrix with normalization



- density 'poor' and severity 'low' then cancer severity will be 'low'.
- density 'average' and severity 'medium' then cancer severity will be 'medium'.
- density 'good' and severity 'high' then cancer severity will be 'high'.

Finally, Fig. 9 represents the cancer severity result obtained from the rules above mentioned.

The predicted result after classification using the deep learning architecture and mass detection using fuzzy membership function is shown in Fig. 10.

Table 3 AlexNet with SVM

AlexNet with SVM observed results		
Dataset	SVM kernel	Accuracy
Full mammography	Linear kernel	55
Full mammography	Gaussian kernel	55
Roi-extracted manually	Linear kernel	64
Roi-extracted manually	Gaussian kernel	59
CBIS-DDSM (without threshold)	Linear kernal	61
CBIS-DDSM (without threshold)	Gaussian kernal	58
Roi-cropped-threshold	Linear kernal	56
Roi-cropped-threshold	Gaussian kernal	55
cbis-ddsm	Linear kernal	60
cbis-ddsm	Gaussian kernal	57

Table 4 Observed accuracy

Models	Dataset	Accuracy
Resnet without svm	Roi extracted manually	78
Alexnet with svm	mias+ddsm	85
Alexnet without svm	cbis-ddsm	63
Alexnet with svm	cbis-ddsm	53.20
Alexnet with svm	Roi extracted manually	64
Alexnet with svm	full mammography	61.35

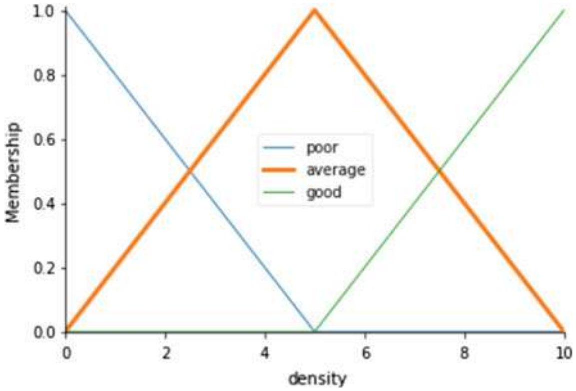


Fig. 7 Density membership function

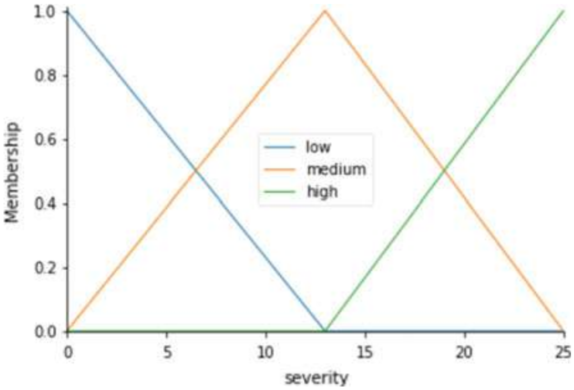


Fig. 8 Severity membership function

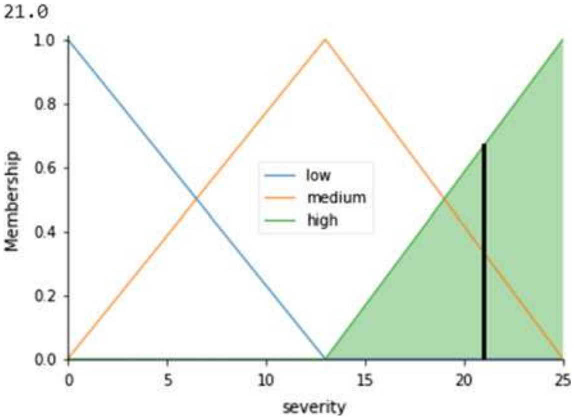


Fig. 9 Cancer severity result

5 Conclusion and Future Work

In this work, we found out that Alexnet with SVM performs much better than resnet model. The linear kernel also gives better results. Fuzzy system estimates the density of the malignant region. It gives about how denser the cancer is from the malignant images. For the clinical data, the accuracy is not considered as the main metric since the accuracy of the model depends on various factors. But for patient data metrics such as precision, recall, AUC etc. play an important role in medical imaging. This paper mainly focused on alexnet and resnet for feature extraction and fuzzy rules

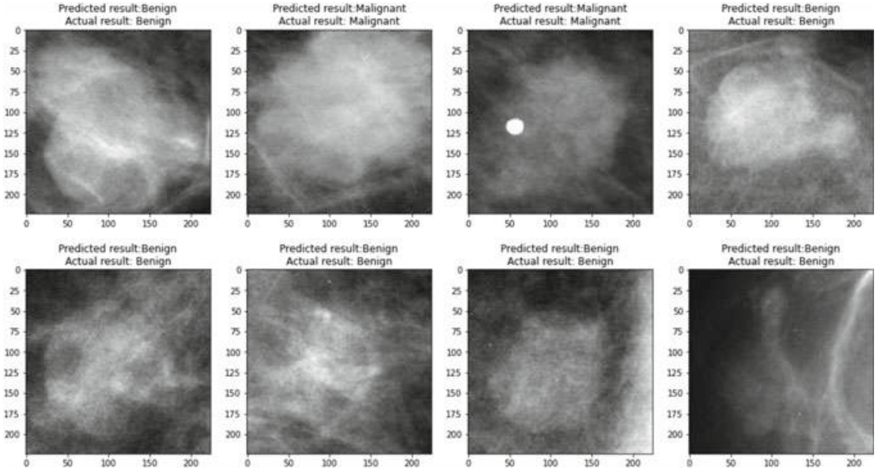


Fig. 10 Prediction results

to estimate the density of the mass area in the breast. So as a future work, it can include Inception, GoogleNet, LeNet etc. and can incorporate various computer vision techniques for estimating the mass area.

References

1. Breast Cancer India. Statistics of breast cancer in India [online]. Available at: http://www.breastcancerindia.net/statistics/trends.html
2. Breastcancer.org is a nonprofit organization and supportive online community [online]. Available at: https://www.breastcancer.org/symptoms
3. Wang, Z., Li, M., Wang, H., Jiang, H., Yao, Y., Zhang, H., Xin, J.: Breast cancer detection using extreme learning machine based on feature fusion with CNN deep features. IEEE Access **7**, 105146–105158 (2019)
4. Shahnaz, C., Hossain, J., Fattah, S.A., Ghosh, S., Khan, A.I.: Efficient approaches for accuracy improvement of breast cancer classification using Wisconsin database. In: IEEE Region 10 Humanitarian Technology Conference (R10-HTC), pp. 792–797. IEEE (2017)
5. Ragab, D.A., Sharkas, M., Marshall, S., Ren, J.: Breast cancer detection using deep convolutional neural networks and support vector machines. PeerJ **7** (2019)
6. Fathy, W.E., Ghoneim, A.S.: A deep learning approach for breast cancer mass detection. Int. J. Adv. Comput. Sci. Appl. **10**(1), 175–182 (2019)
7. Gopakumar, G., Sai Subrahmanyam, G.R.K.: Deep learning applications to cytopathology: a study on the detection of malaria and on the classification of leukaemia cell-lines. Handbook of Deep Learning Applications, pp. 219–257. Springer, Cham (2019)
8. Kalmady, K.S., Kamath, A.S., Gopakumar, G., Subrahmanyam, G.R.S., Gorthi, S.S.: Improved transfer learning through shallow network embedding for classification of leukemia cells. In: 2017 Ninth International Conference on Advances in Pattern Recognition (ICAPR), pp. 1–6. IEEE (2017)
9. Dhanya, R., Paul, I.R., Akula, S.S., Sivakumar, M., Nair, J.J.: F-test feature selection in stacking ensemble model for breast cancer prediction. Procedia Comput. Sci. **171**, 1561–1570 (2020)

10. Dhanya, R., Paul, I.R., Akula, S.S., Sivakumar, M., Nair, J.J.: A comparative study for breast cancer prediction using machine learning and feature selection. In: 2019 International Conference on Intelligent Computing and Control Systems (ICCS), pp. 1049–1055. IEEE (2019)
11. Suckling, S.A.J., Betal, D., Cerneaz, N., Dance, D.R., Kok, S.-L., Parker, J., Ricketts, I., Savage, J., Stamatakis, E., Taylor, P.: The mammographic image analysis society digital mammogram databaseeExerpta medica. In: International Congress Series 1069, pp. 375–378 (1994). https://www.mammoimage.org/databases/
12. Zuiderveld, K.: Contrast Limited Adaptive Histogram Equalization. Academic (1994)

Symposium on Emerging Topics in Computing and Communications (SETCAC'20)

Providing Software Asset Management Compliance in Green Deployment Algorithm

Noëlle Baillon-Bachoc, Eddy Caron, Arthur Chevalier, and Anne-Lucie Vion

Abstract Today, the use of software is generally regulated by licenses, whether they are free or paid and with or without access to their sources. The world of licenses is very vast and unknown. Often only the public version is known (a software purchase corresponds to a license). For enterprises, the reality is much more complex, especially for main software publishers. Very few, if any, deployment algorithm takes software asset management (SAM) considerations into account when placing software on Cloud architecture. This could have huge financial impact on the company using theses software. In this article, we present the SAM problem more deeply; then, after expressing our problem mathematically, we present *GreenSAM*, our multiparametric heuristic handling performance and energy parameters as well as SAM considerations. We will then show the use of this heuristic on two realistic situations, first with an Oracle Database deployment and second with a larger scenario of managing a small OpenStack platform deployment. In both cases, we will compare *GreenSAM* with other heuristics to show how it handles the performance/energy criteria and the SAM compliance.

Keywords Licensing · Cloud · Software asset management · Resources management · Deployment

N. Baillon-Bachoc · A. Chevalier · A.-L. Vion
Orange S.A., Paris, France
e-mail: noelle.baillon@orange.com

A.-L. Vion
e-mail: annelucie.cosse@orange.com

E. Caron · A. Chevalier (✉)
Univ Lyon, EnsL, UCBL, CNRS, Inria, LIP, 69342 Lyon Cedex 07, France
e-mail: chevalier.arthur@ens-lyon.fr

E. Caron
e-mail: caron.eddy@ens-lyon.fr

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_37

1 Introduction

In contrast with public licensing, we have to license software by fulfilling what we call a metric. A metric defines a way to calculate the number of licenses required for software, so we can license it with x licenses under metric A or y licenses under metric B. The price of licenses depends on the used metric. The metrics of software licenses are defined contractually either by the general conditions of sale found on the publisher's website or by a contract between the customer and the publisher. The general terms and conditions of sale available online can be updated, but a license purchased before a change in the legal text must follow the old version of the metric definition, so it is necessary to have a continuous monitoring of these metrics and the ability to identify and retrieve the legal documents of each license when needed.

One problem stems from the fact that we can interpret the metrics definitions in contracts in different ways, and thus, the customer is not in conformity because of an unintended misunderstanding. A concrete and recent example of this is the trial between SAP and Diageo [1], a company that uses SAP products. Because of a lack of usage rights understanding due to legal uncertainty about licenses, the company was fined £55 million ($75 million) for non-compliancy and more recently a disagreement between SAP and AB InBev [2] where the latter was facing a $600 million penalty. Most companies cannot afford or face such penalties. Another telling number is the fact that 85% [3] of companies sanctioned for counterfeiting were unknowingly non-compliant which shows the need for tools to verify and ensure compliance automatically.

The Cloud also brings issues, there is no more obvious relation between software and hardware. Without a border, every instance of an application may run on every server and this dynamicity impact metrics. To solve this case, it is necessary to recreate links between the resources used by software and the hardware layer by any means and at the deepest levels. The problem is that monitoring software usage in the Cloud is a difficult task, made easier by software inventory tools provided by publishers. The problem arises when it is necessary to link physical and software inventories while keeping track of when they were made because a virtual machine or container can move much more easily in the Cloud than in traditional architectures. These problems are critical as compliance is expressed as 'yes' or 'no.' A simple mistake can cause serious damages as seen before. Then comes the problem of identifying the software itself. Indeed, the discovery tools go back to the names of the executables and try to recognize the associated products. Several techniques exist to improve this identification as mentioned in the state of the art [4, 5] but it remains generally difficult. We will focus on the management of these products here under the assumption that the identification is total and accurate.

We can see that the larger a Cloud is, the greater is the need for tools that can track software usage, identify licenses used, verify compliance and manage effective placement. This tool is part of a process called 'software asset management' (SAM) which must be able to carry out the actions described above but also have some industrial objectives allowing to have a return on use and therefore to conduct a supplier strategy or even carry out portfolio consolidations but we will not focus on these aspects.

In addition, the functioning of metrics forces the user to measure the usage of the Cloud in a very extensive and very precise way. The metric function takes in resources ranging from simple physical resources (number of cores, number of users) to much more complicated resources (number of indirect accesses). Each metric will target one to several of these, and to be able to manage all the metrics of each application in the Cloud, we need to know exactly how much material or immaterial resources are being consumed as mentioned above.

One extra problem is that the compliance verification is done retrospectively (after the use so it is too late to take action) and takes a lot of time: for a large Cloud, it can take several months to check the compliance for all installed software without appropriate tools. Besides, to reduce and optimize the license consumption according to the needs, software asset managers should have information from all sides (machine management, human resources, accounting) because of the lack of standardization of the metrics. Indeed, editors can use in their metrics everything they want from physical attributes of servers to number of employees in different teams.

Finally, there is a total lack of deployment algorithm considering compliance and license cost. Only few papers showed an interest in SAM and often with the wrong assumption of one license per software. Also, only considering SAM during the deployment is not possible, as reducing the number of licenses with a processor number-based metric could lead to placing all the software on a single processor server. Therefore, we need to use a multi-parametric heuristic. In 2014, data centers in the USA consumed an estimated 70 billion kWh, representing about 1.8% of total US electricity consumption. Current study results show data center electricity consumption increased by about 4% from 2010–2014 and is expected to continue increasing in the near future by with the same rate. Based on current trend estimates, US data centers are projected to consume approximately 73 billion kWh late 2020. To answer this trend and software asset management problem, we introduce *GreenSAM* which is a multi-parametric deployment heuristic taking into account software asset management considerations as well as performance and energy. Performance and energy are contradictory—forming the basis for multi-objective optimization.

This paper is organized as follows: We start by presenting the state of the art of software asset management and the very few papers talking about deployment algorithm with license considerations in Sect. 2. Then, in Sect. 3, we describe the major advancement of this paper, the *GreenSAM* heuristic before introducing two use cases for it in Sects. 4 and 5. Finally, in Sect. 6, we conclude and discuss future directions to enhance the *GreenSAM* heuristic.

2 Related Work

Genesis of software asset management was in 1999 when Holsing and Yen [6] proposed a study leading to the first considerations on the model and identification of software. In 2004, Ben-Menachem and Marliss [7] underscored the need for investment and the creation of tools for such processes to ensure long-term management of software assets. In 2011, McCarthy and Herger [8] offered a solution to combine information technology (IT), processes and software asset management: This requires the ability to scan all the infrastructure, make a license inventory, implement contract management and to produce reports on the state of readiness for compliance and verification. In 2014, Gocek et al. [9] described the SAM as tools for discovering and collecting information on instances of software used in monitored environments. Recently in 2017, Vion et al. [10] made a brief survey of the existing SAM tools, their benefits and proposed a SAM model for a Cloud architecture. Moreover, in 2018, Chevalier et al. [11] proposed an efficient and economical way of handling metrics in Cloud environments for Oracle Database and showed that a deployment algorithm focused on software licenses could save money. During the same year [12] proposed optimization of the placement of virtual machines with many other parameters including license costs and showed that handling both problems of mapping virtual machines to physical machines and mapping applications to virtual machines leads to better results than considering the two problems in isolation. Even so the problem is well formulated, it uses the fact that an application uses one license at most as we see in his UML model and that the number of licenses does not rely on underlying architecture which is an unrealistic view of the licenses. We need to tackle this problem.

On the other hand, to identify software, several works were first carried out in 2014, Han et al. [5] proposed a way to identify open-source software. Cho et al. [4] proposed a technique on proprietary software to reduce counterfeiting through a birthmark technique located in the executable. A data discovery agent could use this birthmark to correctly identify the software. Then in 2018, Vion [13] proposed another way to identify the software following a purchase by relying on ISO 19770-3 and -4 [14] standards to correctly recognize the software and the associated rights.

As many IT services and analyses use the Cloud and are dependent on large infrastructures that can be local or remote [15], more and more questions are being asked about the huge energy consumption of these infrastructures to supply and cool computing machines [16]. Research has been conducted to reduce these consumptions [17] and heuristics have been proposed such as *GreenPerf* [18] for example which introduces a performance and power consumption ratio to improve energy efficiency or energy-efficient framework dedicated to Cloud [19]. In the context of 5G, very recent research shows that it is possible to reduce consumption [20, 21]. The field of multi-parameter optimizations is very wide and many researches are based on Pareto fronts in fields ranging from sensor networks [22, 23] to virtual machine deployment [24].

3 GreenSAM: Energy and Software Asset Management

The *GreenSAM* heuristic aims at ensuring compliance during deployment of applications while saving energy and keeping a sufficient level of performance. This heuristic takes as input the set of servers we can deploy on and for each servers, its attributes which we describe later. First, *GreenSAM* will remove servers that will causes loss of compliance because of contractual rules. Then, it will compute the Pareto front based on the servers' attributes (here energy, performance, and license number) to eliminate non Pareto efficient ones giving us a subset. Then we compute scores for each server in this subset based on a formula optimizing the parameters described in Sect. 3.2. We finally deploy the software on the server with the best score. If two server get the same score, then we choose at random. We can see in Fig. 1 the flow of the deployment algorithm using the *GreenSAM* heuristic.

These different steps are described in the following subsections.

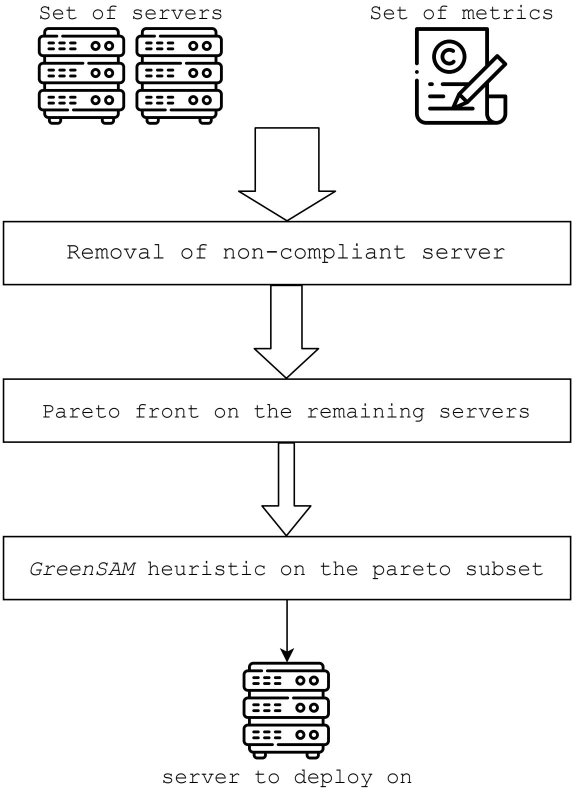

Set of servers
Set of metrics
Removal of non-compliant server
Pareto front on the remaining servers
GreenSAM heuristic on the pareto subset
server to deploy on

Fig. 1 Flow of deployment algorithm using *GreenSAM* heuristic. The first step removes the servers that cause a non-compliance situation if the software is deployed on them. Then, in second step, we apply a Pareto front to the remaining servers to keep only the Pareto optimal servers. Finally, to choose one in the Pareto subset, we apply the *GreenSAM* heuristic and take the best server

3.1 Servers Attributes

As performance and energy as huge research domains and evolve quickly, *GreenSAM* is based on agnostic parameters. It means that every parameter we input into *GreenSAM* is scores given by the user. The goal is to be able to compare servers based on these scores. If the user wants to use cutting-edge computing method for energy consumption, he will not have to use another heuristic each time and *GreenSAM* will not have to understand this cutting edge method. This way, we can enhance the accuracy of *GreenSAM* by enhancing the method to compute scores we give to it. Therefore, *GreenSAM* manipulate parameters without units but is able to compare servers by normalizing the attributes. Having two servers with 50 and 30 energy consumption will have the same order relation than two servers having 100 and 60 energy consumption.

In our case, the energy consumption is computed with the number of active cores on the server. The SAM score is given by the price of the licenses we would have to possess if we place the software on one particular server. Finally, we used a modular performance indicator. For example, the performance of a distributed high-performance computing application will be affected by the network speed but also by the parallel power of the machine (heterogeneous architecture or not, for example). Storage service in the Cloud will be efficient if many people can access it at the same time and if it has enough storage (more or less fast). We define different performance classes in *GreenSAM* to describe the performance calculation of each product if we want to have a precise index of whether or not a product is performing in a given environment. A product is therefore defined by its metrics and its performance calculation function (performance class).

3.2 Multi-objective Optimization

Any optimization problem will have design parameters whose best possible values from the viewpoint of the objectives are sought to be attained in the optimization process. The optimization task here is to map a set of software onto available resources, here servers. The three objective functions are defined with the following variables:

n The number of servers noted s

m The number of applications noted a

A_s The attributes of a server s

f_a The formula for the metric of the application a that takes into account attributes of servers

E_s The energy consumed by the server s for the first installation. When we deploy an application a on a server already containing one then $E_s = 0$.

PI_a The performance class of the application a. It is a function waiting for attributes of a server to give performance score. If the application a is not installed on the server s then $PI_a(A_s) = 0$.

3.2.1 Minimize Energy Consumption

The total energy consumption 'E_T' of our deployment is then expressed as:

$$E_T = \sum_{s=1}^{n} E_s \times (\neg(\exists\, a \in [1 \ldots m]/a \in s)) \tag{1}$$

3.2.2 Maximizing Performance

To avoid putting all applications on the same server, we added operational constraints. Each application will require at least 2 cores not used so $PI_a(A_s) = 0$ if a does not fit on s. The overall performance P_T is expressed as:

$$P_T = \sum_{s=1}^{n} \sum_{a=1}^{m} PI_a(A_s) \tag{2}$$

3.2.3 Minimizing Software Cost

This objective function will stop the process if the metric computation brings a non-compliant state. The total license consumption L_T is expressed as:

$$L_T = \sum_{s=1}^{n} \sum_{a=1}^{m} f_a(A_s) \tag{3}$$

In most cases, machines that bring performance will have higher energy consumption, implying that objectives P_T and E_T are contradictory—forming the basis for multi-objective optimization.

We first compute the three criteria each time we deploy a product because the results per server may vary between each deployment, indeed the performance indicator can vary because of the past deployment as well as the license cost (e.g., the metric depends on the number of instance of the product on the server) which provide the Cloud dynamic state. Then we filter servers that are not compliant with the current product metrics (e.g., there can be constraints with country or language). Afterwards, we apply a Pareto front to this dataset to have a subset of potentially good servers. We still have to choose one of the 'best servers' given by the Pareto front and so we implemented a function to give a score to each server of the subset. This score function will simply divide normalized performance score by the sum of normalized license and energy points. We add one to the denominator to avoid being in a case of a server having no energy and license consumption breaking the division. Finally, *GreenSAM* will return the first server of the sorted subset to deploy the software on it before starting again for the next product to deploy.

Note that in the case of the deployment of multiple products, we may not have the optimal deployment because of local optimization. *GreenSAM* goal is to optimize a unique deployment but could be enhanced later to handle multiple deployment at once.

4 Oracle Database Enterprise Edition Use Case

For both use cases, we use two datasets. The first one is a set of about 5000 servers coming from Orange™ Cloud (Orange™ is the first historical French multinational telecommunications corporation). This dataset allows us to express the efficiency of the Pareto front reducing these 5000 servers to subsets of tens of potential servers. The next datasets if used to compare *GreenSAM* with other heuristics described later in this section. For each comparison between the four heuristics, we generated hundred datasets of 200 servers with random but realistic attributes. We then compute the average results of the four heuristics on these hundred datasets to make a fair comparison on many Cloud architectures.

In this use case, we deploy 10 Oracle databases in a new Cloud to allow development projects to use them. From the benefit of [11], we can prove that in our use case; it is better to focus on the processor metric which is defined as follows: To deploy a database d on a server s with c_s cores, a corefactor co_s and inside a cluster Cl_s then the number of licenses you need to have to install the database on t is the following:

$$L = \sum_{s=1}^{n} c_s \times co_s \times (s \in Cl_t) \times (\neg(\exists\, a \in Cl_t)) \tag{4}$$

where a is a previously installed database meaning you have to sum all the cores of the cluster in which the server t is, except if there is already a database somewhere in this cluster.

This upstream calculation avoids the deployment algorithm having to compute each metric of each product before making the Pareto front.

As we have only one product to deploy ten times, the only performance indicator is defined as follows: We must have a minimum of 2 cores to be eligible and the performance score is the number of cores divided by the number of already installed databases on it.

We can see in Fig. 2 the Pareto front on the first deployment. In subsequent deployments, it is always interesting to deploy databases on the same server until this server no longer has enough cores available. Finally, in Fig. 3, we can see that the front of the Pareto only has a few servers left until the end of the deployment. In Fig. 4, we compared the result of this algorithm with three others on different sizes of Cloud:

- **PerfEnergy**: The first is an algorithm that promotes performance first, followed by energy.

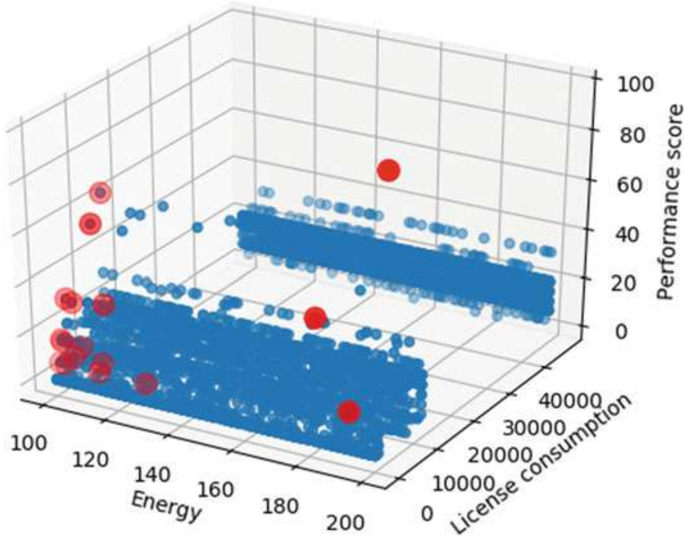


Fig. 2 17 servers (in red) as a part of the Pareto front on the 5000 servers used for this simulation

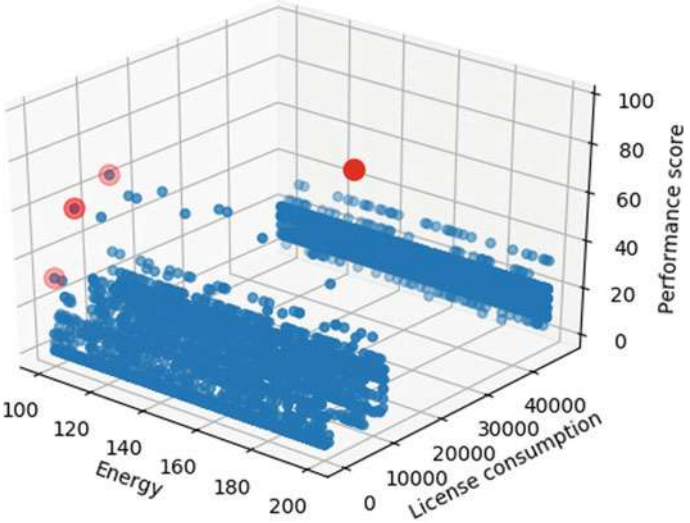


Fig. 3 Only four servers are part of the Pareto front at the end of the 5th deployment. Then until the end, there will be as many. In the end, eight different servers will be used

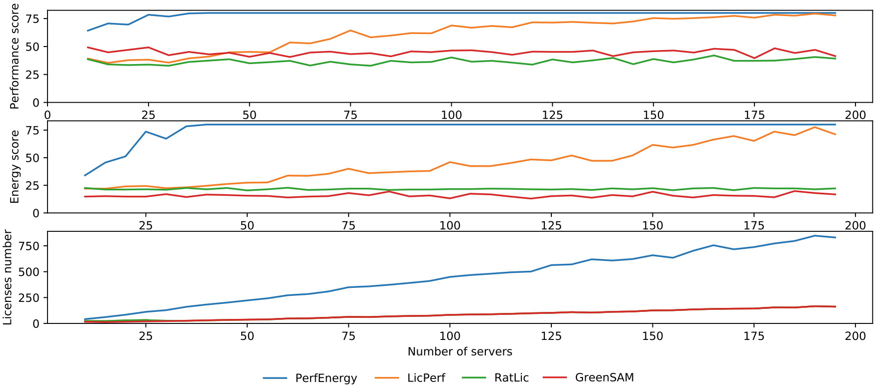


Fig. 4 *GreenSAM* is a lot better in performance and license criteria even though it loses around 30% of performance. RatLic is close but does not achieve the same performance and LicPerf is terrible in terms of energy as PerfEnergy

- **LicPerf**: The second one focuses on the number of licenses and then on performance.
- **RatLic**: The last one focuses on the ratio performance by energy and then license consumption.

We can see in the results that *GreenSAM* is pretty good and succeeds in minimizing energy but at the price of performance. In the next section, we use *GreenSAM* in a more complex deployment that is an OpenStack environment.

5 OpenStack Use Case

In this case, we will deploy a minimal OpenStack platform with the metrics of the RedHat editor. The purpose of this deployment is to have a lead node (called director), ten compute nodes and twenty CEPH nodes. Each node has its metrics and performance class defined as follows:

Director It must be located in a cluster with as many servers as possible to be able to deploy as many compute and CEPH nodes as desired. The number of machines in the parent cluster will, therefore, be the performance index.

Compute It needs a lot of cores but must be deployed on single servers. A machine that has too many VMs cannot be reused. The performance index is the number of cores.

CEPH It requires a lot of memory. The performance index is the amount of memory available. We cannot reuse a server already used for another application.

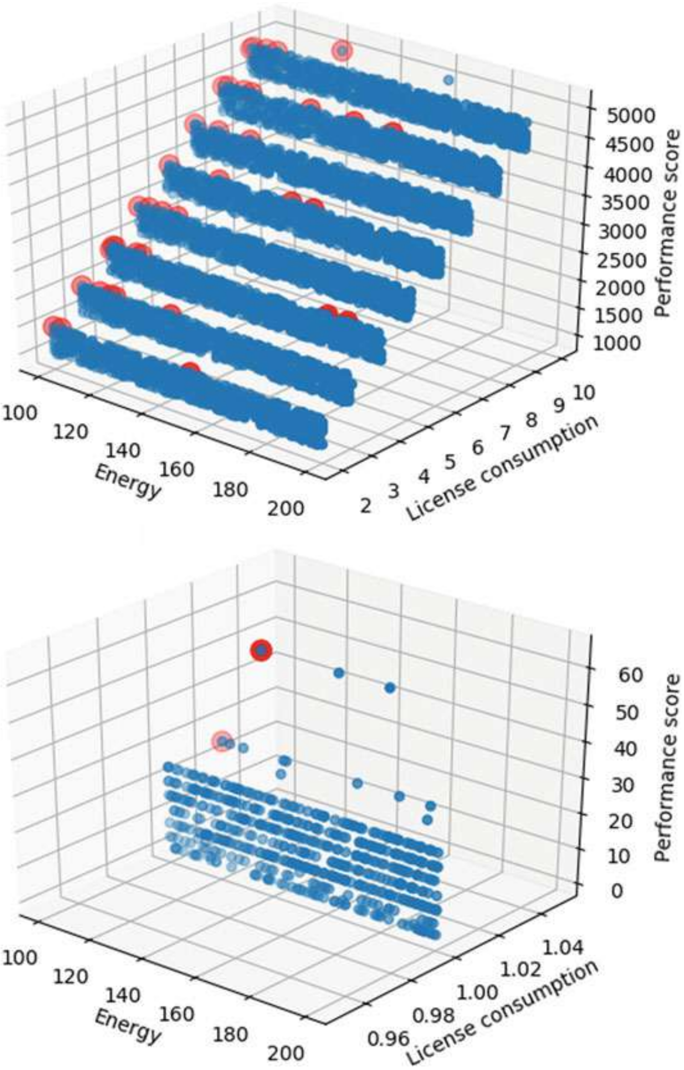


Fig. 5 Top: second deployment of the CEPH product. Bottom: second deployment of compute product. The license dimension has no impact here as the metric demands one license per installation

For the volume of licenses, you need one license for the lead node, one license per compute node and one license per 500 memory slots for CEPH nodes. We can see in Fig. 5 the Pareto front of the CEPH and compute products.

Finally, in Fig. 6, we can see the comparison between *GreenSAM* and three other algorithms. *GreenSAM* still saves energy while keeping the number of licenses required low at the price of performance, which remains acceptable unlike the LicPerf

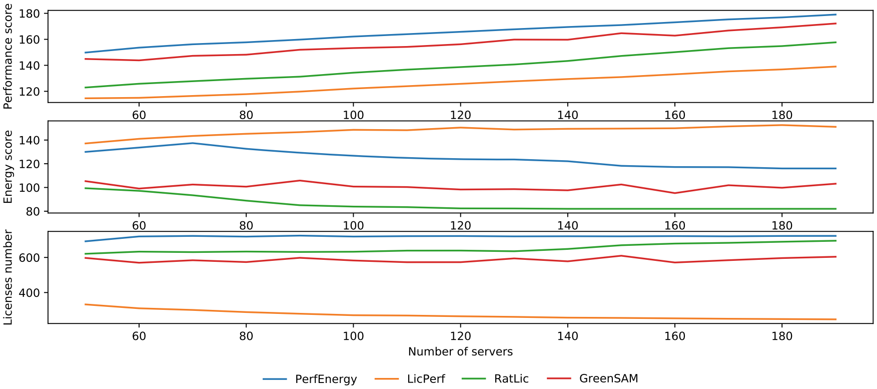


Fig. 6 This shows that *GreenSAM* still manages to get good results on a multi-products deployment. RatLic is saving more energy but fails at reducing license consumption and saving performance. LicPerf is very good at license criteria but still terrible at saving energy and performance

or RatLic algorithm. It demonstrates that going only for the performance in the Cloud can lead to spending more money on software or disastrous scenarios like Diageo [1] and shows the need to have algorithms like *GreenSAM* that take into account software licensing.

6 Conclusion

This paper introduces *GreenSAM*, a multi-objective heuristic for deploying services in the Cloud that guarantees license compliance while reducing energy consumption but maintaining reasonable performance. This heuristic gets very good results on two use cases with two significant datasets (one simulated and one real) and compared to three others heuristics. In both experiments, *GreenSAM* achieved reducing energy and license consumption while maintaining acceptable performance and keeping full compliance by filtering bad servers. Compared to the prior state of the art, we are now able to deploy automatically and while respecting compliance different products on a Cloud architecture all the while optimizing energy and performance.

This heuristic allows in the case of a 5G tool deployment to ensure good power consumption and software compliance and therefore could be used in network orchestrators such as ONAP [25] for 5G to manage the SAM part of the deployment while allowing energy savings in a technology that will be massively used. This use would remain very dependent on the execution time of the deployment calculation because they are done at high speed in the context of the virtualized network. To conclude, this heuristic offers a solution to manage an IT environment and the metrics associated with the products, even with complex licensing rules.

For the enhancement of *GreenSAM*, new approaches can be considered. Indeed, the implementation of a better heuristic function for sorting could allow choosing a better server from the subset of servers on the Pareto front. This cost function would be composed of the attributes that we want to optimize and would give a better score for each server to obtain a unique one. Another way would be to specify the objectives to have a finer deployment in our Cloud and add parameters to allow us to have a performance index closer to reality. It would be equally interesting to compare performance drop on each performance category to see the impact of software classes and improving the algorithm.

In addition, we could add more parameters to this heuristic to have a more realistic deployment. With the concept of agnostic parameters, we could modify the score function of *GreenSAM* to add networking or security considerations. Pushing further, it would be interesting to see if it is possible and profitable to make parameters fully generic. Allowing the user to add any parameter only giving attributes scores. Finally, we could study the problem of multiple deployment at once with the *GreenSAM* heuristic to avoid making only local optimizations.

References

1. England (Technology WHC, Court) C (2017) SAP UK Ltd v. Diageo Great Britain Ltd [2017] ewhc 189 (tcc). http://www.bailii.org/ew/cases/EWHC/TCC/2017/189.html
2. Sayer P (2018) SAP settles licensing dispute with AB InBev. https://www.itworld.com/article/3264435/sap-settles-licensing-dispute-with-ab-inbev.html
3. Flexera (2014) Flexera software and IDC research survey report: software license audits and costs & risks to enterprises. http://learn.flexerasoftware.com/slo-wp-key-trends-audits-cost-risk
4. Kim, D., et al.: A birthmark-based method for intellectual software asset management. In: Lee, S., et al. (eds.) The 8th International Conference on Ubiquitous Information Management and Communication, ICUIMC '14, Siem Reap, Cambodia— 09-11 Jan 2014. pp 39:1–39:6, ACM (2014). https://doi.org/10.1145/2557977.2558062
5. Han, Y., et al.: A new detection scheme of software copyright infringement using software birthmark on windows systems. Comput. Sci. Inf. Syst. **11**(3), 1055–1069 (2014). https://doi.org/10.2298/CSIS130918064H
6. Holsing, N.F., et al.: Software Asset Management: analysis, development and implementation. Inf. Resources Manag. J. **12**(3), 14 (1999). https://doi.org/10.4018/irmj.1999070102
7. Ben-Menachem, M., et al.: Inventorying information technology systems: supporting the "paradigm of change". IEEE Softw. **21**(5), 34–43 (2004). https://doi.org/10.1109/MS.2004.1331300
8. McCarthy, M.A., et al.: Managing software assets in a global enterprise. In: 2011 IEEE International Conference on Services Computing, pp. 560–567 (2011). https://doi.org/10.1109/SCC.2011.119
9. Gocek, P., et al.: Obtaining software asset insight by analyzing collected metrics using analytic services. US Patent 9,652,812 (2017). https://www.google.com/patents/US9652812
10. Vion, A., et al.: Software license optimization and cloud computing. Cloud Comput. **2017**, 125 (2017)
11. Baillon, N., et al.: Towards economic and compliant deployment of licenses in a Cloud architecture. In: Workshop: Cloud Management and Operations, in Conjunction with IEEE Inter-

national Conference on Cloud Computing (IEEE CLOUD 2018), San Francisco, USA (2018). https://hal.inria.fr/hal-01808751. hal-01808751

12. Mann, Z.A.: Resource optimization across the cloud stack. IEEE Trans. Parallel Distrib. Syst. **29**(1), 169–182 (2018). https://doi.org/10.1109/TPDS.2017.2744627
13. Vion, A.: Software asset management and cloud computing. PhD, Université Grenoble Alpes (2018). https://tel.archives-ouvertes.fr/tel-01901991
14. Wikipedia (2019) ISO 19770 Wikipedia. https://www.iso.org/standard/68531.html
15. Foster, I., et al.: Computational grids. In: Palma, J.M.L.M., Dongarra, J., Hernández, V. (eds.) Vector and Parallel Processing–VECPAR 2000, pp. 3–37. Springer, Berlin, Heidelberg (2001)
16. Dongarra, J., et al.: The international exascale software project roadmap. Int. J. High Perform. Comput. Appl. **25**(1), 3–60 (2011). https://doi.org/10.1177/1094342010391989
17. Berl, A., et al.: Energy-efficient cloud computing. Comput. J. **53**(7), 1045–1051 (2010). https://doi.org/10.1093/comjnl/bxp080
18. Balouek-Thomert, D., et al.: Energy-aware server provisioning by introducing middleware-level dynamic green scheduling. In: 2015 IEEE International Parallel and Distributed Processing Symposium Workshop, pp. 855–862 (2015). https://doi.org/10.1109/IPDPSW.2015.121
19. Orgerie, A.C., et al.: When clouds become green: the green open cloud architecture. In: PARCO (2009). https://hal.inria.fr/ensl-00484321v1
20. Zhang, K., et al.: Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks. IEEE Access **4**, 5896–5907 (2016). https://doi.org/10.1109/ACCESS.2016.2597169
21. Zappone, A., et al.: Energy-efficient power control: a look at 5G wireless technologies. CoRR abs/1503.04609 (2015). http://arxiv.org/abs/1503.04609
22. Sengupta, S., et al.: Multi-objective node deployment in WSNs: in search of an optimal trade-off among coverage, lifetime, energy consumption, and connectivity. Eng. Appl. Artif. Intell. **26**(1), 405–416 (2013). https://doi.org/10.1016/j.engappai.2012.05.018
23. Khalesian, M., et al.: Wireless sensors deployment optimization using a constrained Pareto-based multi-objective evolutionary approach. Eng. Appl. Artif. Intell. **53**, 126–139 (2016). https://doi.org/10.1016/j.engappai.2016.03.004
24. Xu, B., et al.: Dynamic deployment of virtual machines in cloud computing using multi-objective optimization. Soft Comput. **19**(8), 2265–2273 (2015). https://doi.org/10.1007/s00500-014-1406-6
25. Linux Foundation (2019) ONAP: Open Network Automation Platform. https://onap.org

An Analysis of Rainstreak Modeling as a Noise Parameter Using Deep Learning Techniques

B. Akaash and R. Aarthi

Abstract Outdoor vision systems (OVS) play a vital role in the surveillance of the environment. However, the images and videos captured by these systems could be severely tampered by the sharp intensity changes brought about by adverse weather and climatic conditions. In this work, synthetically prepared rain images are modeled to visualize the randomly distributed rainstreak patterns as noise. The analysis has been performed using various deep learning networks such as auto-encoders with and without skip connections and denoising convolutional neural networks (DnCNN). The best model for this process has been suggested based on mean squared error (MSE), peak signal-to-noise ratio (PSNR), and structural similarity index (SSIM) obtained by comparing the original and the reconstructed image.

Keywords Outdoor vision systems · Denoising convolutional neural networks · Auto-encoders · Skip-connections · Synthetic dataset · PSNR · SSIM

1 Introduction

In the current day scenario, outdoor vision systems (OVS) play an imperative role in the surveillance and monitoring of the exterior environment and perpetually aim in providing security and protection to the civilians habituated in that environment. The hardware constituents may include CCTV cameras, drones, go-pros, cameras in self driving cars, etc., which record real-time images or videos of the area that they vividly cover depending on their coverage range. However, these outdoor scenes are vulnerably affected by the adversities of weather and climatic changes.

B. Akaash · R. Aarthi (✉)
Department of Computer Science and Engineering, Amrita School of Engineering,
Coimbatore, India
e-mail: r_aarthi@cb.amrita.edu

B. Akaash
e-mail: cb.en.p2aid19005@cb.students.amrita.edu

Amrita Vishwa Vidyapeetham, Coimbatore, India

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_38

These weather changes can be broadly classified into steady and dynamic conditions. The atmosphere is prone to mist, haze and fog, which are grouped as steady conditions. The dew droplets formed in such a scenario are too small to be explicitly detected by a camera and hence, their individual intensities are summed to produce the aggregate effect of these drops. On the contrary, raindrops and snowflakes are categorized as dynamic conditions, where these dewdrops are significant enough to be individually recognized. Thus, rain projects itself to be an ill-posed problem because of the low-probability of prediction of various unforeseen factors. These include the nature of the rain streaks, the intensity with which it falls, the density of the rain drops, sudden atmospheric changes, impact of sunlight, etc.

Object detection being a high level vision task has always been very important when it comes to surveillance systems because of the advancement in smart traffic control systems and autonomous cars. In the previous work, the focus on object detection and the factors affecting the constituency of traffic monitoring and OVS has been discussed in [1, 2]. Conventional image processing techniques have worked well, but in the case of OVS, applying traditional techniques has adverse effects on the background information of the image. There has been significant work done using weighted image filtering techniques, i.e., by using guided filters to preserve geometrical details in rain-removed image [3]. Also, the authors in [4] have thus taken the liberty to explore the Res-Net architecture, which constructively work on preserving the background whilst removing the rain component.

Using semi-supervised method, the work in [5] has trained the network using both unlabeled real-world data and labeled synthetic data to induce better generalization in the model. Post the process of deraining, proper reconstruction of clean image is an important step. In [6], treating each of the RGB color channels separately and combining them with variational auto-encoders has exhibited a better reconstructed image. To understand and inspect the effect of rain in an image, [7] presents the need to understand the photometric and geometric properties of rain. Temporal correlation and similarity have also been used to model deep learning networks to understand the rain patterns [8]. Other related work in this domain can be found in [9–12].

In the present work, two methods have been proposed for image deraining, considering the rain streaks as a noise component. In the first method, stacked auto-encoders have been analyzed with and without skip connections and in the second method, residual networks have been analyzed. The dataset used is Rain800 [9], where 700 images were used to train and 100 images were used to test the model. The models are evaluated based on metrics, i.e., mean squared error (MSE), peak signal-to-noise ratio (PSNR), and structural similarity index (SSIM).

The following sections elucidate on the methodology and further discuss the analysis and results. The paper culminates with conclusion and future scope.

2 Methodology

In this section, the flow diagram adapted is exhibited along with an elucidation about the dataset. The latter segment deals with a detailed description of the deep learning models.

2.1 Proposed Work and Dataset

The flow diagram of the proposed work is given in Fig. 1. The dataset used here is Rain800 [9]. Post-performing and pre-processing techniques, the work constructively caters to two vigilantly designed techniques, i.e., one, by mapping a function to implicitly learn the constituent of a clean image, from that of an image tampered with rain streaks and the other, which aims to efficiently learn the distribution of these rain streaks and hence, successfully subtract it to obtain the clean image. The former technique projects rain as an inverse problem and elucidates the powerful exploitation of the convolutional auto-encoders with and without skip connections while the latter explores the concept of residual learning built using denoising convolutional neural networks (DnCNN).

The Rain800 dataset is synthetically developed on a particular set of clean images, where the rain streaks are photoshopped in such a way that various streak patterns are embedded onto clean images. Thus, a single image conjointly presents both these images such that the right portion consists of the clean image. These streak patterns are varied based on the direction of the rainfall, their scaled patterns, intensity of the rain blotches, etc.

Figure 2 shows a sample of the used dataset. Figure 2a presents the clean image whilst Fig. 2b explicitly showcases the synthetically obtained rainy image. The images portray to have a varied nature and size, thereby emphasising on the necessity of preprocessing. The raw image, which contains both the clean as well the rainy image is halved widthwise, and stored in separate folders. Following this step, all the newly obtained images are resized to a uniform size of 256 × 256 pixels. These images are subsequently fed to various deep learning models for training in order perform the required objective. The design of each of these models has been discussed in the following subsection.

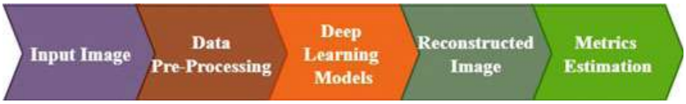


Fig. 1 Flow diagram of the proposed work

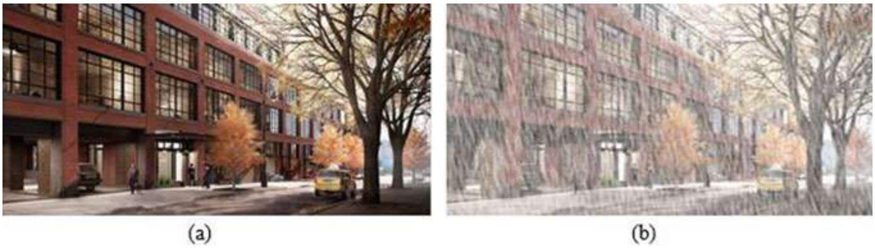


Fig. 2 Dataset sample

2.2 *Model Design*

2.2.1 Neural Networks with Stacked Convolutional Autoencoders

1. Autoencoders without Skip Connections
Convolutional deep networks present state-of-the-art results in various computer vision problems such as image denoising, impainting, and super-resolution. The convolutional operator keenly aims to extract the salient features by performing filtering operations on the original input image. Convolutional auto-encoders are developed to capitalize this operation. This in turn is used to encode the input and furthermore, the auto-encoder tries to decode the image. The architecture is presented in Fig. 3.

Internally, it consists of three constituents, i.e., an encoder, latent space, and a decoder. The encoder is designed to read the input image and filter it in such a way that the notable features are "learned" by the model itself. This thus becomes a classic example of unsupervised learning as the model simply is trained to learn a function $f(x)$ such that it understands the features of the image and further aid in reconstructing an output image given in Eq. 1, where x is the original image, $f(x)$ is the encoder function, $g(f(x))$ is the decoder function and y is the reconstructed image.

$$y = f(x) \tag{1}$$

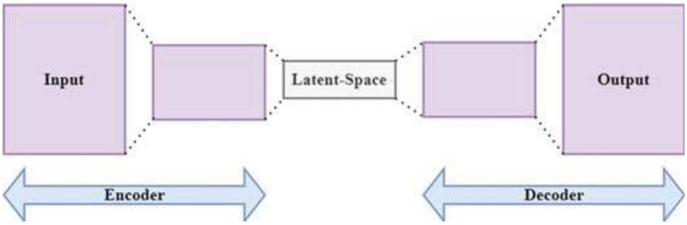


Fig. 3 Architecture of regular autoencoder network

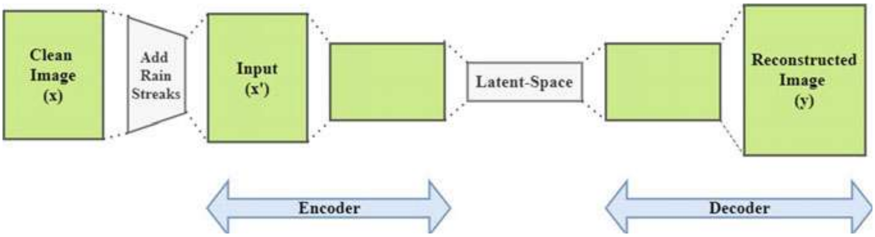


Fig. 4 Architecture of denoising autoencoder network without skip connection

Latent space showcases the compressed representation of the input image. Since all the features are now in a condensed form leading to a bottleneck, it should be ensured that overfitting of the model does not happen and hence, the concept of regularization is effectively induced here. The decoder, whose working is quite contrary to that of an encoder, learns to formulate $g(f(x))$ in order to obtain y, such that the loss encountered between x and y is as minimal as possible. Thus, in a regular auto-encoder, the procured loss is verily, $L(x, y)$, where L could be mean squared error (MSE).

$$L(x, y) = \frac{1}{MN}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}||x_{ij} - y_{ij}||^2 \tag{2}$$

However, as this work focuses on the process of deraining a rainy image to retrieve a clean image, the block diagram represented in Fig. 3 can be modified to the one exhibited in Fig. 4.

Hence, in the case of such images, the so-called denoising process is evidently an analogy to “deraining”. That is, the original image x, is induced synthetically with some rain (noise) component, creating x'. Now, x' acts as the input to the auto-encoders, now nomenclatured as denoising auto-encoders, which aim in learning a function $f(x')$ and during the decoding process, it learns a corresponding function, represented in Eq. 3, where y represents the reconstructed image.

$$y = g(f(x')) \tag{3}$$

The loss function is given in Eq. 4, where $L(x, y)$ is a statistical parameter mean squared error, which measures the average squared difference between the reconstructed image y and the original image x.

$$L(x, y) = L(x, g(f(x'))) \tag{4}$$

The objective of such a model is to reduce the loss between these two images. It is to be noted that in the auto-encoder network, the loss function always introduces a penalty when the input x is dissimilar from the reconstructed y.

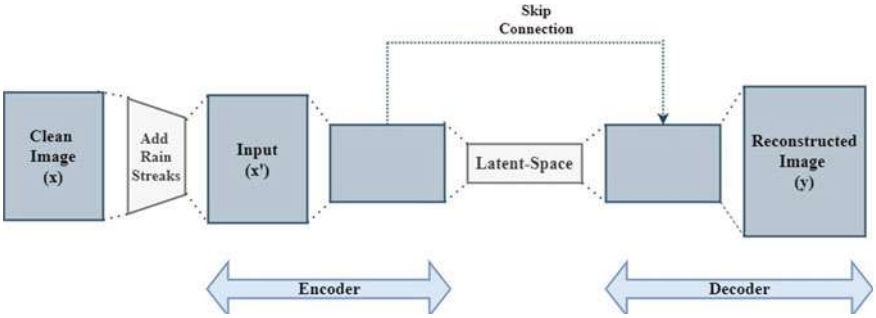


Fig. 5 Architecture of denoising autoencoder network with skip connection

2. Auto-encoders with Skip Connections

Deep neural networks tend to suffer from a degradation problem if there are numerous convolutional and deconvolutional layers. In most cases, a problem of information loss surfaces during and after these layers perform the reconstruction of the original image. This loss points to gradient decrease in model learning. Therefore, for better learning of the networks, skip connections are added amidst correspondingly mirroring convolutional and deconvolutional networks. These connections efficiently aid in the process of back-propagation of the gradients to the encoder, and effectively pass the nuances of the image to the decoder, thereby attaining a noticeable performance improvement (Fig. 5).

The passed convolutional feature maps are aggregated element-wise to the deconvolutional feature maps and are passed to the next layer after rectification, thus preserving the main objective of the convolutional layers, that is, to behave as a feature extractor, which preserve the primary components of image properties as it simultaneously eliminates the noise. Such networks are called residual networks, whose objective is to figure out whether to use all the layers of the design or to “skip” a couple of layers in order to preserve or improve the performance of the model. These skip layers are added across the bottlenecks and the obtained reconstructed image portrays an image quality better than the ones constructed using regular auto-encoders. Here, the equation of the loss function is retained which is same as Eq. 4.

2.2.2 Denoising Convolutional Neural Networks (DnCNN)

In this subsection, end-to-end CNN is designed, which adopts a residual learning strategy to remove the noisy observation and hence offer a clean image, unlike the previous case where the estimation of the clean image was done directly by demonstrating deraining as an inverse problem. With this approach, DnCNN strives to discard the noise component to produce a clean image by passing through multiple hidden layers. The major difference here is that pooling is not incorporated, as the intention is to retain the size of the original image. The convolutional filters are set

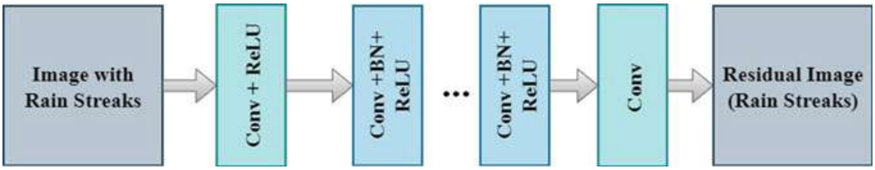


Fig. 6 Architecture of a residual network

to a size of (3 × 3) and the receptive field of DnCNN is given as $(2d + 1)x(2d + 1)$, where d is the depth. Let the original image be x and x' represents the obtained image post inducing x with the rainstreaks. The model is trained to obtain $R(x')$, which depicts the residual mapping of rainstreaks. The model obtains a function g, as an output of the subtract layer as represented in Eq. 5.

$$g = x - R(x') \tag{5}$$

The objective function here again, is a loss function, such that

$$L = \frac{1}{2N}\sum_{i=1}^{N} ||x_i - g_i||^2 \tag{6}$$

To summarize, the DnCNN model targets to formulate $R(x')$ and notably, batch normalization is adapted in order to boost and speedup the model's performance. ReLU function is used to separate the image's features from its rainstreaks, as it traverses through these hidden layers. The layers used are presented in Fig. 6.

3 Analysis and Results

This section elucidates on the network architecture and the metrics used to analyze the performance of each of the models.

3.1 Network Architecture

3.1.1 Autoencoders Without Skip Connections

The encoder network has three layers of convolution networks each of 1,286,432 filters with a filter size of (3 × 3) and after each layer, a subsequent max pooling layer is used to extract the important and relevant information of the image, which in-turn leads to reducing its size. The image which was originally 256 × 256 × 3 is now reduced to a compressed representation. Deconvolution layers with filters 3,264,128 for decoder network are designed to reconstruct back the original from reduced representation and subsequent upsampling layers to bring this back to the original size.

3.1.2 Auto-Encoders with Skip Connections

This network is pretty similar to the previous model, but the addition is the skip networks across the bottlenecks, i.e., from the corresponding encoder and decoder layer brings a significant difference. The encoder network has two consecutive convolutional layers of the same number of filters with size (3 × 3), i.e., 64 and 128, respectively, followed by a subsequent max pooling layer. Post that, a convolutional layer of 256 filters is embedded into the network. The decoder network starts with an upsampling layer which is followed by two deconvolutional layers with 128 and 64, respectively. Finally, the skip connection is added from the encoder layer.

3.1.3 Denoising Convolutional Neural Networks

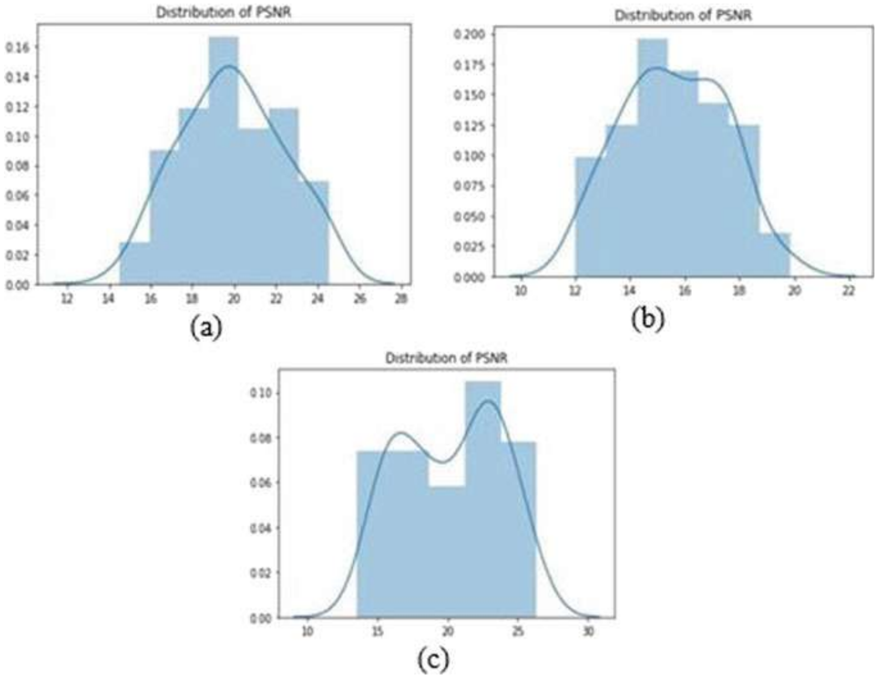
The CNN representation is without max pooling because the agenda is to retain the original size of the image. The first layer is a convolution layer with specification as (64, (3, 3)) added along with ReLU function. This is followed by ten convolution layers with the same specifications and activation function, added along with batch normalization. It is to be noted that the depth has been limited to 10 because of the computational constraint. However, more depth would produce stupendous results. Additionally, another convolution layer is added to obtain the RGB channels of the image. Finally, a subtract layer is utilized to find the clean image from the rainy image, which has been subtracted with the modeled noise residual.

3.2 Metrics

As mentioned in the previous sections, the standards of measurement used to analyze and evaluate each of these models are described below.

3.2.1 Mean Squared Error (MSE)

As described above, the loss of all the models is estimated using MSE. It has been observed that train and test data show decrease in loss, with the iteration of each epoch. The test data loss almost follows the train data loss, showing the consistency of the models. Figure 7 shows the respective graphs of the loss of each model.



(a)

(b)

(c)

Fig. 7 PSNR distributions of deep learning models

3.2.2 Peak Signal-to-Noise Ratio

The PSNR represents the ratio between the maximum power value of a signal and the power of distorting noise that affects the quality of its representation. Because many signals tend to have a varied dynamic range, PSNR is expressed in logarithmic decibel scale. In the below formula, the numerator of the logarithmic fraction represents the highest possible pixel value of the reconstructed image.

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{\mathrm{MAX}_{y_{ij}}^{2}}{\mathrm{MSE}}\right) \tag{7}$$

The PSNR distributions are portrayed in Fig. 7 where (a), (b), and (c) represent the PSNR distribution patterns for the deep learning models auto-encoders without skip connections, auto-encoders with skip connections and DnCNN network respectively.

It can be concluded that all the graphs show normal distribution pattern. PSNR distribution for the first case has a single peak. The values are distributed in the range between 14 and 24 dB and at 20 dB, model showcases the peak value. This shows that the PSNR is distributed evenly across its mean value. The second graph shows a flattened peak between 14 and 16 dB, showcasing a range for high PSNR values, unlike the previous case. Here too, the values are evenly distributed across the averaged mean of the distribution. Since there is a range of values, the reconstruction may not be as close as that of the original image. However, the plot for DnCNN is left skewed, meaning the mean of the distribution is present on the right hand side (22–24 dB), indicating high PSNR value, and thus, it can be inferred that this has the best PSNR value and hence shows the best reconstruction of the image amongst all the models. Also, this shows that most of the images fall in the high PSNR range.

3.2.3 Structural Similarity Index

The structural similarity index [13] is an enduring metric that quantifies the degradation of an image quality due to extensive image processing or modelling. Here, SSIM is obtained between the reconstructed image y and the original image x (Fig. 8).

The SSIM distributions labeled (a), (b), and (c) are nomenclatured similar to Fig. 7. The graphs show similar distribution as that of PSNR and are normally distributed. It is evident that auto-encoders without skip connections and DnCNN portray a single peak, indicating that the features of the image are understood by the model and the black box models have interpreted the objective of deraining the image. However, the middle graph, pointing to the model with skip connections, demonstrates that the model has performed poorly when compared to the other two, as a wide range of SSIM values have been showcased.

The respective model values are summarized in Table 1. The reconstructed images of all the models along with the original image are shown in Fig. 9.

From the above figure, it can be observed that all the three models have tried to denoise the rainstreaks. Regular auto-encoders and DnCNN have worked on retaining the background image. However, the former model was unable to completely understand the nuances portrayed by the model and produced a blurry derained image and hence shows that some more tuning of the model is essential. The model with skip connections has also attempted to derain the image, but the output image shows tainted white patches with poor reconstruction of the original image. The rightmost picture clearly has performed the best by not just removing the synthetic rainstreaks, but has also sharpened the background image effectively, making it look closer to that of the original image.

4 Conclusion and Future Scope

In this work, three models have been analyzed thoroughly using deep neural networks. The auto-encoder models with and without skip connections aim in learning the important features of the image to produce a reconstructed clean image, without the rain. Feature maps are transferred from one layer to another layer thus improving the reconstructed image quality than using normal auto-encoders. However, the DnCNN focuses on learning the rain mapping and subsequently targets to subtract

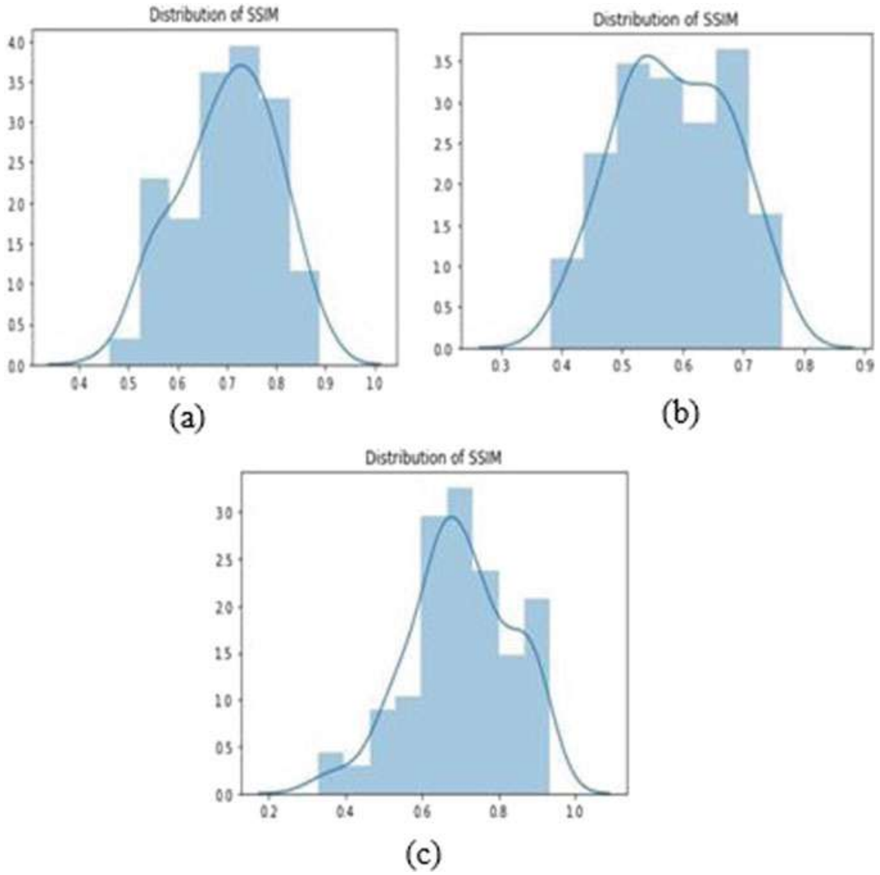


Fig. 8 SSIM distributions of deep learning models

this modeled function from the input image in order to obtain the clean image. It has been observed that DnCNN produces the best results when compared to that of the auto-encoders, as the reconstructed image does not lose the background information while trying to denoise the rainstreaks. The background of the reconstructed image sharpens, producing better quality of the clean image and thereby boosting the PSNR and SSIM values, which are 20.185 db and 0.700, respectively.

However, due to computational limitations, the models could not be optimized. Hence, this work can be further extended by aiming to improve the above-presented results using optimized models and work on tuning the parameters of each of the models. Also, the models can be experimented by accounting real-world rain data and checking for its performance. Finally, image processing techniques could be adapted to model the rainstreaks, closer to the real-world scenario.

Table 1 Multirow table

Deep learning model	PSNR (dB)	SSIM
Auto-encoders without skip connections	19.912	0.675
Auto-encoders with skip connections	15.582	0.585
Denoising convolutional neural network	20.185	0.700

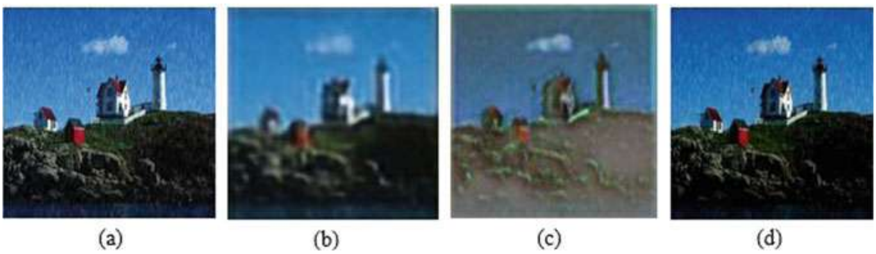


Fig. 9 From **a** to **d**: original image, reconstructed images of auto-encoders without skip connections, with skip connections and DnCNN, respectively

References

1. Sushmitha, S., Satheesh, N., Kanchana, V.: Multiple car detection, recognition and tracking in traffic. In: 2020 International Conference for Emerging Technology (INCET), pp. 1–5. IEEE (2020)
2. Haritha, H., Senthil Kumar, T.: Survey on various traffic monitoring and reasoning techniques. In: Silhavy, R., Senkerik, R., Kominkova Oplatkova, Z., Prokopova, Z., Silhavy, P. (eds.) Artificial Intelligence Trends in Intelligent Systems. CSOC 2017. Advances in Intelligent Systems and Computing, vol. 573. Springer, Cham (2017)
3. Shi, Z., Li, Y., Zhang, C., Zhao, M., Feng, Y., Jiang, B.: Weighted median guided filtering method for single image rain removal. EURASIP J. Image Video Process. **2018**(1), 1–8 (2018)
4. Himabindu, Y., Manjusha, R., Parameswaran, L.: Detection and removal of raindrop from images using deep learning. In: Smys, S., Tavares, J., Balas, V., Iliyasu, A. (eds.) Computational Vision and Bio-Inspired Computing. ICCVBIC 2019. Advances in Intelligent Systems and Computing, vol. 1108. Springer, Cham (2020)
5. Yasarla, R., Sindagi, V.A., Patel, V.M.: Syn2Real transfer learning for image deraining using Gaussian processes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2726–2736 (2020)
6. Du, Y., Xu, J., Qiu, Q., Zhen, X., Zhang, L.: Variational image deraining. In: The IEEE Winter Conference on Applications of Computer Vision, pp. 2406–2415 (2020)
7. Garg, K., Nayar, S.K.: Vision and rain. Int. J. Comput. Vis. **75**(1), 3–27 (2007)
8. Liu, T., Xu, M., Wang, Z.: Removing rain in videos: a large-scale database and a two-stream ConvLSTM approach. In: 2019 IEEE International Conference on Multimedia and Expo (ICME), pp. 664–669. IEEE (2019)
9. Zhang, H., Sindagi, V., Patel, V.M.: Image de-raining using a conditional generative adversarial network. IEEE Trans. Circ. Syst. Video Technol. (2019)

10. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising. IEEE Trans. Image Process. **26**(7), 3142–3155 (2017)
11. Cai, B., Xu, X., Jia, K., Qing, C., Tao, D.: Dehazenet: an end-to-end system for single image haze removal. IEEE Trans. Image Process. **25**(11), 5187–5198 (2016)
12. Li, S., Araujo, I.B., Ren, W., Wang, Z., Tokuda, E.K., Junior, R.H., Cesar-Junior, R., Zhang, J., Guo, X., Cao, X.: Single image deraining: a comprehensive benchmark analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3838–3847 (2019)
13. Similarity Index. https://en.wikipedia.org/wiki/Structural_similarity. Accessed 7 2020

Diverting Tantrum Behavior Using Percussion Instrument on Autistic Spectrum Disorders

Zefanya Lintang, Djohan, Fortunata Tyasrinestu, and Phakkharawat Sittiprapaporn

Abstract The aim of this study was to distract the tantrum of the patient with autistic spectrum disorders (ASD). Children with tantrums disturbed others and even the children themselves, as well as the activities they were doing. From this condition, we attempted to divert the tantrums of the children with ASD by using percussion instruments. The use of musical instruments is an effective method to redirect the behavior intended to be changed since musical instruments are interesting and pleasing objects for children suffering from ASD. Owing to the matter, music and its instruments have a chance to be utilized in diverting the behavior of ASD children. This experiment by single subject design ABA took a single subject of ASD which was 7 years old. The musical instruments applied are five percussions which are tambourine, glockenspiel, tifa, maracas, and bell. The result showed that the subject throwing tantrums are hardly managed in the first baseline. It was supported by the mean baseline, which is 7.3 min, whereas the average result in treatment showed 2.3 min. It revealed that the time span of the subject's response in treatment was less than in the baseline. However, in the treatment phase, the subject's behavior was distracted. It occurred because the subject seemed interested and enthusiastic as playing the percussions. The treatment also increased the cognitive aspect of the subject, focusing on ongoing

Z.L. originated the concept; designed the study; drafted the manuscript; analyzed the data. D. originated the concept; designed the study. F.T. performed a literature search. P.S. critically analyzed data; critically revised the final version.

Z. Lintang · Djohan · F. Tyasrinestu
Graduate School of Indonesia Institute of the Arts of Yogyakarta, Yogyakarta, Indonesia

F. Tyasrinestu
Department of Music Education, Indonesia Institute of the Arts of Yogyakarta, Yogyakarta, Indonesia

P. Sittiprapaporn (✉)
Brain Science and Engineering Innovation Research Group, Mae Fah Luang University, Bangkok, Thailand
e-mail: wichian.sit@mfu.ac.th

Neuropsychological Research Laboratory, Department of Anti-Aging and Regenerative Science, School of Anti-Aging and Regenerative Medicine, Mae Fah Luang University, Bangkok, Thailand

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_39

activities. The subject could play bell based on the therapist's instruction and holding on to it until 15 min. The percussion instrument was able to improve the subject's focus on playing bell and distract the tantrum.

Keywords Autistic syndrome disorder · Children · Music therapy · Tantrum behavior · Percussion

1 Introduction

Basically, human undergoes development from prenatal period, infancy, childhood, adolescence, adulthood, and old age. Nevertheless, not all human could grow optimally; human's development often faces disturbance or disorder. The disorders can be caused by external factor, such as environmental influence, and the internal one, biological factor. There are many cases of developmental disorders from the prenatal period until childhood. It often happens, for in that period, the body's anatomical system of infants and children has not been completed yet. One of the disorders that can occur in the prenatal period, infancy, and childhood is autistic spectrum disorders (ASD). ASD is a neurodevelopmental disorder affected by many factors: genetic and environmental factors that can already be detected since toddler and will be continued throughout the life span. People with ASD have communication, social interaction, language, mind, and behavior different from children in general. This disorder can affect how children learn, communicate, build the social relation, and adjust to the surroundings. The number of people with ASD has been developed since the last two decades [1]. United States is one of the countries with the highest number of ASD sufferers, 1:68, with male and female ratios 5:1 [2]. It also occurs in Asia, such as Indonesia. In 2011, ASD sufferers in Indonesia were 1:1000, much more than 10 years before [3]. Based on diagnostic and statistical manual of mental disorder (DSM V), mental development of ASD children is slower than children on their ages [4]. ASD is a part of pervasive developmental disorders (PDD). PDD is divided into two and an umbrella of autistic and non-autistic syndrome (PDDs). In PDDs, there are types of disorders, such as Asperger's syndrome, pervasive developmental disorder, Nos, fragile X syndrome, Ret's syndrome, and childhood disintegrative disorder. PDDs basically has similarity between autistics, but the symptoms are different [5]. The children with ASD have symptoms, like laughing, crying, being angry for no reasons, and being unable to control the emotion as temper tantrum (raging, rolling over, and screaming) or harming self by banging on his/her own head [6]. Tantrum behavior is the often-seen symptom of children with ASD. A behavior at the first glance looking the same as tantrum is meltdown. However, tantrum and meltdown are very distinctive.

Tantrum is an intentional behavior often happening if children feel tired, hungry, frustrated, and when children are not being in a good feeling. Tantrum always has purposes, for example, as children do not want to sleep. Then, they are raging in order to prevent the parents and nannies from asking them to sleep. Tantrum is designed by the children to get attention from the targeted people. Meltdown, on the other hand, is not a behavior caused by children's frustration, hunger, and exhaustion, yet external stimulation. Meltdown has no purposes and is not addressed to anyone. It is not also a behavior for children to seek attention. It can occur for the whole age of children with ASD while tantrum, as they are getting older, is able to disappear [7]. As having tantrums, children with ASD are very difficult to handle. Nonetheless, there are some ways to solve tantrums of autistics. One of the methods to soothe the children with tantrum is by diverting their focuses on other interesting and calming objects. With this, the children will forget the things or matters making them cry and throwing tantrums. There are many objects that can be applied to distract the focuses of ASD children having the tantrums; one of them is music [7]. In Philippine, music is proven effective to divert tantrums of children with ASD [8]. Hence, the aim of this study was to distract the tantrum of the patient with autistic spectrum disorders (ASD).

2 Materials and Methods

2.1 Subject

This study applied a single subject design who was suffering from ASD with the classification of level 2 "Requiring Substantial Support", without any comorbidity. The gender of the subject was male with 7 years old.

2.2 Research Design

This study utilized an experimental research method with a single subject design. It was commonly known as N of 1 design. The study used one subject out of three targeted subjects diagnosed with ASD. This study observed the response of the subject as the tantrum was distracted by percussion instruments. The method application was selected because it was used to evaluate the effect of the tantrum's diversion of the single subject taken from the Permata Ananda Therapy Center. The use of a single-case experimental design was used to measure an early condition of the subject as the pretest function. We used research design A-B-A because we intended to observe the upcoming change and impact after the treatment by measuring the duration of the subject's response in the tantrum diversion with percussion instruments. We assessed the target behavior in the baseline phase. In order to validate this study, the data, subject, and the analysis process were reliable as the prevention toward scientific data contamination with the effects that have nothing to do with the observation's goal, calculation, and analysis. Reliability in this study was done by two raters who encoded a video. The video was the recording during the baseline assessment up to second baseline. Raters monitored the video and gave a sign if the subject was having a tantrum. Raters were going to provide (+) symbol if the tantrum occurred while (−) if it did not. The sign was presented on a paper given by us. Data analysis was accessed into two phases. First, evaluation was created in accordance with design quality. Second, it was made for the assessment to treatment effectiveness (intervention).

2.3 Procedure

Before conducting the research, we did the informed consent to the subject's companion. It aimed to gain permission from the subject's representation to be involved in the study and to inform the research's overview, including the action done toward the subject. Then, we did the baseline observation conducted in four meetings. This study was begun on August 8, 2018, started with the subject's companion, therapist, and Permata Ananda's chief interview. After the interview, the baseline session was held on August 13, 2018. We did the assessment by using The Individualized Music Therapy Assessment Profile (Intake Form). This evaluation ended on September 10, 2018. The treatment phase was accomplished in 12 meetings started on September 10 until December 3, 2018. Before the treatment, we arranged a music handling plan. It was to determine the goals and activities in the study in order to be well-organized. The music handling plan included the subject's necessities, goals, indicators, and measured aspects. The items used for the test are five instruments including tambourine, glockenspiel, tifa, maracas, and bell, respectively. The last session held on December 10 was baseline; to evaluate treatment results, the subject had done and experienced.

3 Results

After doing the handling plan and assessment for 4 times and 12 times for treatment, the researcher reports the music evaluation results for therapy. The description is provided below (Table 1).

The reliability of the result depends on trustworthy data. In order to know whether or not the data were reliable, it needed to present the percent agreement. Thus, this study provided 12 intervals of a period of times done by two observers (see Tables 2, 3 and 4).

According to Sunanto [9], calculating the percent agreement can be performed by assessing the total of percent agreement with the formula below:

$$\frac{O+N}{T} \times 100$$

Table 1 Evaluation report of music for therapy

Date of therapeutic treatment Person in charge	August 13, 2018 ZL
Subject's personal data	
Name	Hg
Place and date of birth	Balikpapan April 5, 2010
Gender	Male
Diagnosis/special condition	Autistic spectrum disorder (ASD)
Client's general description	
Before music therapy session, Hg was speaking with unclear voices. He like dropping objects and patting his hands. Besides, his verbal skill was less. Hg lacked focus if he was doing an activity. Hg was less responsive if somebody called him as well	

$$\frac{7+3}{12} \times 100 = 95\%$$

Occurrence agreement was the interval as the subject responds to the percussion instruments to distract the tantrum, and it happened simultaneously (agreement) between observers 1 and 2. In the table, it took seven times on the interval 1, 2, 5, 8, 9, 10, and 11. The non-occurrence agreement was the interval when the subject did not respond to the instruments as the tantrum diversion based on both observers. It occurred three times. T was the number of intervals used. There were 10. It can be concluded that the data has 95% of the total percent agreement. The data analysis of this study applies visual graphic analysis. The data is presented in the table and graphic form along with the explanation as follows:

According to Fig. 1, it was shown that the tantrum occurred for 7 min in the first session. In the second session, the tantrum appeared for 8 min. In the third session, there was no tantrum taking place. In the fourth session, the tantrum lasted for 7 min. Therefore, the duration of the tantrum in the pre-treatment phase was long-lasting and stable. The average duration for pre-treatment was analyzed as follows:

$$\frac{7+8+7}{3} = 7.3 \text{ minutes}$$

Based on Fig. 2, it was revealed that the tantrum appeared in the first session lasts for 3 min. In the second session, it happened in 2 min and 0 duration in the third session. In the fourth session, the tantrum appeared, yet the subject did not respond so that it had no duration. Then, it occurred in the fifth session and lasts for 3 min. As the third session, the sixth and seventh sessions had 0 duration, since tantrum did not take place. The tantrum appeared in the eighth, ninth, tenth, and eleventh sessions with 1, 4, 2, 1-min(s) duration. The twelfth session showed 0 duration even though the tantrum happened, yet no response from the subject. As the result, the average of the treatment was

Table 2 Subject's goals and objectives

Subject's needs	Goals	Objectives
Fine Motor: Hg had not been consistent yet in differentiating right and left. Moreover, Hg's writing skill was weak	To increase the skill of differentiating right and left, and writing skill	Instructing him to play Tambourine with right and left hands. For the writing skill activity, Hg was asked to write the activity before it began
Sensory skills: Hg often dropped objects with his both hands. Then, he patted his hands	To reduce the habit of dropping objects and patting hands	Playing glockenspiel by instructing him to hit it hardly and slowly
Receptive communication: Hg found difficulty to sing along due to the weak of verbal skill. Sometimes, he only could say he wanted or not	To increase his verbal skill, so that he could sing along	Singing along by stimulating Hg to follow fragment of words given by the therapist
Expressive communication: Hg was still unable to build communication owing to the lack of verbal skill. Besides, Hg had the habit of producing voice while closing his mouth	To reduce the habit of producing unclear sounds and increase communication skill	Doing an activity of singing with the lyrics persuading him to communicate, like mentoring his own name, the therapist's name, and the name of days and hours
Cognitive: Hg found it hard to focus on doing activity. He could focus in 5 min maximum, and he was doing whatever he wanted for the rest. Hg was also weak in long-term memory	To increase the focus of doing the activity and develop the long- and short-term memory	Playing bell based on the instruction of the therapist. For instance, the therapist mentioned a color and Hg was asked to push the bell in accordance to what is uttered by the therapist. Hg could have been focusing for 15 min
Emotional: Hg could not regulate and tolerate himself sometimes as he was angry. However, when he was on the tantrum, he could be diverted by instruments, especially percussions	To increase the skill of self-regulation and tolerance	Giving reward for Hg if he could do the task as instructed by the therapist. It is expected to give motivation for him to regulate himself
Social: Hg often ignored the calls from others. He could respond to it after being called for several times. Hg could not build communication	To increase the skill of responding the calls and building communication	Attracting his attention by playing an instrument he liked, so that he wanted to respond the call from the therapist and singing the opening song by asking him the therapist's name, the name of the days and hours

(continued)

Table 2 (continued)

Subject's needs	Goals	Objectives
Musical: Hg could not sing the whole part of the song, only some fragments of the words in the song	To increase the verbal skill in order to sing all of the part of the song	Introducing and persuading Hg to sing the fragments of words from the song's lyrics sang by the therapist and giving him chance to sing by himself without any assistance from the therapist

Table 3 Evaluation of music for therapy

Evaluation	Description
Fine motor	In fine motor, Hg had not reached the goal yet. Hg still found difficulty to differentiate right and left consistently, and he was still unable to write
Sensory skill	In sensory skill, Hg still weakness, since he liked dropping objects and patting his hands
Receptive communication	In receptive communication, Hg had not achieved the goal yet due to the lack of his verbal skill
Expressive communication	In expressive communication, Hg had not reached the goal yet because he still liked to produce unclear sounds
Cognition	In cognitive aspect, Hg had achieved the goal. He could already increase his long- and short-term memory
Emotion	In emotional aspect, Hg had not reached the goal, for he still could not regulate and tolerate himself as angry. Nevertheless, his tantrum could be diverted by instruments, particularly percussions
Social	In social aspect, Hg had achieved the goal. He could answer if being called
Music	In musical aspect, Hg had not reached the goal, since he still lacked verbal skill

Table 4 Percentage agreement

Interval	1	2	3	4	5	6	7	8	9	10	11	12
Observer 1	+	+	−	−	+	−	−	+	+	+	+	−
Observer 2	+	+	−	+	+	−	−	+	+	+	+	+

$$\frac{3+2+3+1+4+2+1}{7} = 2.3 \text{ minutes}$$

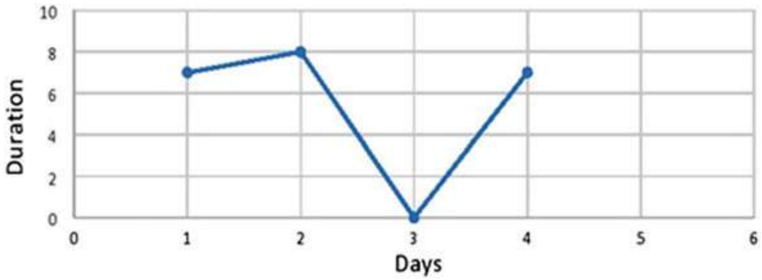


Fig. 1 Duration of tantrum behavior during the pre-treatment assessment

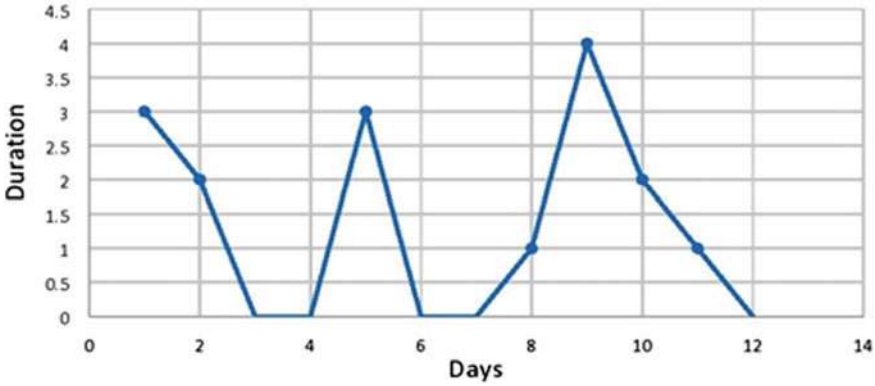


Fig. 2 Duration of tantrum behavior during the post-treatment assessment

Comparing pre-treatment and post-treatment, it was found that the tantrum behavior decreased in the tenth until the twelfth session implying that the durations were not stable. Comparison of duration for tantrum behavior between pre-treatment and post-treatment showed that duration of tantrum behavior in post-treatment decreased with statistical significance compared to the duration of tantrum behavior in pre-treatment (Pre-treatment: 7.3 ± 0.57 min., Post-treatment: 2.3 ± 0.74 min., $t(4) = 0.48$; $p < 0.0001$) (see Fig. 3).

4 Discussion

Our subject in the present study was the survivor of ASD with hyperactivity taking therapy in an Autistics Therapy of Permata Ananda. We only observed the response of the subject as he was having tantrums and diverted by percussion instruments. This study applied behavioral modification to distract the tantrum of the subject. The method utilized was a single subject with the A-B-A model. The data revealed that

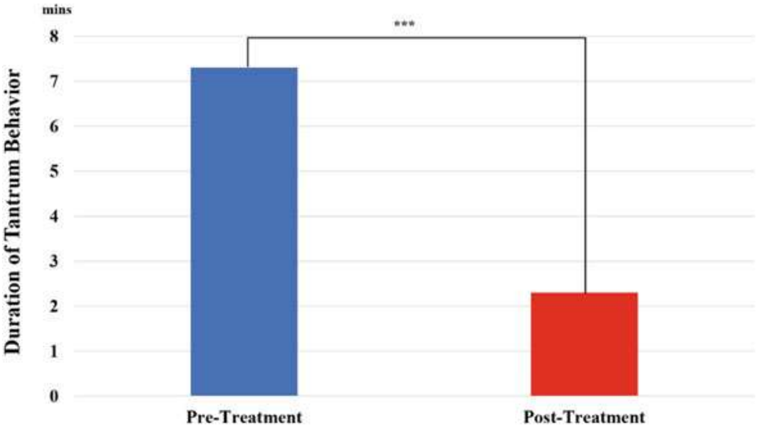


Fig. 3 Comparison of duration for tantrum behavior between pre-treatment and post-treatment (***$p < 0.001$)

in the first until the fourth baseline's session, the tantrum of the subject had a very long duration, 7.3 min. The subject needed a longer time to calm down and focus on the activities he was doing before. There was a significant difference in the treatment data stating that the tantrum's duration of the subject was less than on the baseline, which was 2.3 min. When the subject started throwing tantrums, we immediately diverted him to the percussions. With that diversion, the subject seemed distracted and took the instruments. We observed that the subject had an interest in percussions; it was shown when the activity of playing music instruments runs and his tantrum got distracted quickly. We found that percussion instruments were able to divert the tantrum behavior of the subject.

Many children respond to music positively since music brings out a positive experience, like happiness, joy, and safety. Music is also an interesting medium for children with ASD, and it enables them to express and divert their negative emotions, as anger, frustration, and sadness [10, 11]. Thaut [12] states that children suffering from ASD have a longer time with music than normal children do. ASD children can also perceive and process musical stimuli despite limitations in terms of perception. When children with ASD are focusing on one thing, it is often difficult to redirect their focuses, so it needs other objects to draw their attention. Music instruments are the objects able to move the autistics' concentrations due to their shapes and produced sounds. Children with ASD show that they have an appeal to certain musical instruments. Hyperactive children tend to be interested in percussions, the drum is chosen, since it can release negative energy. Besides, the rhythm produced can make them pleased. The drum can be a catharsis medium for children with ASD. Furthermore, cymbal can also draw the attention of the children because it stimulates them. The vibration they feel as they are hitting and touching it can make them mesmerized [13]. These previous studies are related to our observation done in Permata Ananda Therapy Center. From the observation, we found one of the children has a tantrum, for he did not get what he wanted. He kept crying and shouting. It happened along with the music therapy session, and another child was found playing bell. The child with a tantrum immediately went into the music therapy class and tried to play the bell. From the early observation, we did treatment children with ASD in the Permata Ananda Center. We hypothesized that percussion instruments could be used as an alternative to distract the tantrums of the children. After the focuses were transferred, the children could play the instruments and forgot their own tantrums. Percussions were applied owing to their varied shapes and sizes. Percussion instruments need stroke, shake, and friction to create sounds. The sounds came from the vibration of the instruments. Therefore, it gave impacts to children with ASD. Moreover, percussions had different timbre and pitch and could be accessed by children suffering from ASD as it was easy to play with [14].

Some previous studies focus on dynamic music therapy where patients, alone with the music therapist, or improvise music typically basic instruments, for example, cymbals, djembes, maracas, and others. Music therapy patients are invited to tune in, play, sing, or compose music, based on the strategy utilized [15]. In addition to pleasure of playing, which is of significance for patients who regularly have serious conditions that keep them from enjoying common distractions, such an activity has benefits including better motor coordination [15], improved social interaction, and capacity to express feeling [16]. Technology has been attacking most health-related fields for a long while, although several promising attempts were made to improve music therapy sessions using computers, none of the previous systems really managed to get out of the lab, most likely due to impracticality issues [17]. Music in itself was not remedial; what the kids express about their performance was what was important issue. One point of the system was to get the kids associated with the music production process as profoundly as could be expected under the circumstances. The investigation was an incredible achievement where the kids indicated critical inspiration to test and play with the system, and this for in at least three reasons: confidence, innovation, and capability [18]. Music can give a psychological benefit, and music therapy has been acknowledged for some individuals over the world. Iso-principle has been known as a significant concept in music therapy [19]. Music has been known to have psychological impacts and music therapy has been acknowledged for some individuals over the world [20]. Three components of music include rhythm, melody, and harmony, respectively [21].

In summary, the innovation that interfaces individuals and music has developed throughout the year, particularly in the modern era where new societies keep on rising. Accordingly, we have the chance to appreciate music anytime and anyplace. With respect to the issues and future viewpoint, it appears to be that every individual cannot perceive the existence and benefits of the music or has not completely used it. Music will make individuals more joyful if new innovations are broadly perceived and customized use is widespread [19].

5 Conclusion

The aim of this study was to distract the tantrum of the patient with autistic spectrum disorders. This study attempted to divert the tantrums of the children with autistic spectrum disorders by using percussion instruments. Throwing tantrums were hardly managed in the first baseline. After the treatment phase, the subject's behavior was distracted. The subject became interested and enthusiastic in playing the percussions. The treatment also increases the cognitive aspect of the subject, focusing on ongoing activities. Thus, the percussions instrument was able to improve the subject's focus on playing the bell and distract the tantrum.

Acknowledgements This study was technically supported by Mae Fah Luang University (Electroencephalogram Laboratory 2019), and Brain Science and Engineering Innovation Research Group, Mae Fah Luang University, Thailand. We thank subject who participated in this study.

References

1. World Health Organization: Autism Spectrum Disorders and Other Developmental Disorders from Raising Awareness to Building Capacity. WHO Press, Switzerland (2013)
2. Center for Disease Control and Prevention: Community Report on Autism. The Autism and Developmental Disabilities Monitoring Network, United States (2016)
3. Ministry of Health of Republic of Indonesia: Dedikasi untuk Autis. Mediakom Edisi, Jakarta (2015)
4. Markam, S., Suprapti: Peluang Perkembangan Psikoterapi Dinamik di Indonesia. Understanding the Vulnerable Ego. Konferensi Nasional Psikoterapi, Bali (2004)
5. Szatmari, P.: The Classification of autism, Asperger's syndrome, and pervasive developmental disorder. Can. J. Psychiat. **45**(8), 731–738 (2000)
6. Kusdiyanti, S.: Deteksi Dini Gangguan Perkembangan Autism Oleh Orang Tua. Mimbar **16**(3), 27–54 (2000)
7. Maureen, B.: Tantrums in Autism: new study says it's behaviour not frustration. https://autismawarenesscentre.com/tantrums-in-autism-frustration-at-poor-communication-or-behaviour-issue/. Last accessed 20/08/2019
8. Deala, R.E.: Multi-approach intervention in enhancing adaptive behavior of ASD: a within-subject experimental design. J. Child Adolescent Behav. **5**(2), 1–23 (2007)
9. Sunanto, J.: Pengantar Penelitian Dengan Subyek Tunggal. University of Tsukuba, Indonesia (2005)
10. Chiang, J.Y.K.: Music Therapy for Young Children Who Have Special Needs: The Music Therapy Experience from the Perspectives of Careers and Professionals. Massey University, New Zealand (2008)
11. Kim, J., Tony, W., Christian, G.: Emotional, motivational and interpersonal responsiveness of children with autism in improvisational music therapy. Sage Publications and the National Autistic Society **13**(4), 390–409 (2009)
12. Thaut, M.H.: A music therapy treatment model for autistic children. Music Therapy Perspect. **1**(4), 7–13 (1984)
13. Marin, M.V.: Exploring music therapy for Filipino autistic children. Philippine J. Psychol. **37**(2), 1–33 (2004)
14. Matney, B.: The use of percussion in therapy: a content analysis of the literature. Nordic J. Music Therapy **25**, 372–403 (2016)

15. Bruscia, K.: Case Studies in Music Therapy. Barcelona Publishers, Phoenixville, PA (1991)
16. Lecourt, Ě.: La Musicothérapie analytique de groupe. Improvisation, écoute et communication. Courlay, Fuzeau (2007)
17. Streeter, E.: Reactions and Responses from the Music Therapy Community to the Growth of Computers and Technology—Some Preliminary Thoughts. Voices: A World Forum for Music Therapy (2007)
18. Benveniste, S., Jouvelot, P., Michel, R.: Wii Game Technology for Music Therapy: A First Experiment with Children Suffering from Behavioral Disorders. Multi Conference on Computer Science and Information Systems, MCCIS'08 (Gaming), pp. 133–137. Amsterdam, Netherlands (2008)
19. Hirai, Y., Bando, H., Yoshioka, A., Nishikiori, Y.: Music and man in art: the future of media and technology. Glob. J. Arts Soc. Sci. **2**(1), 116 (2020)
20. Michel, E., Pinson, J.: Music Therapy in Principle and Practice. Springfield, IL, Charles C. Thomas (2005)
21. Bando, H.: Music therapy and internal medicine. Asian Med. J. **44**(1), 30–35 (2001)

Text Sentiment Analysis Using Artificial Intelligence Techniques

Sanskriti Srivastava, M. Vergin Raja Sarobin, Jani Anbarasi, and Namrata Sankaran

Abstract In today's world, data is being generated with such high velocity and variety that analyzing such large volumes of data to extract meaningful results is a taxing job and manually impossible. Developing methods to analyze such large volumes of data for the purpose of finding hidden patterns to achieve meaningful interpretations is necessary for many organizations for making informed decisions. Sentiment analysis is one such method that is used to interpret the emotions represented by text data. There exists a broad range of applications for sentiment analysis. Public opinion is an important "business insight". All businesses are interested in analyzing the consumer behavior, understanding their needs, understanding their likes and dislikes and their buying patterns, and as more and more people are becoming vocal about their preferences, the data required by these companies is becoming readily available on blogs and social media platforms, but the analysis of such large amount of raw data to derive useful conclusions is a hectic task if tried to perform manually. In such cases, sentiment analysis can be used to analyze this raw data. Sentiment analysis can be used for a variety of needs ranging from understanding the public opinion of a government policy to assigning movie ratings from analysis of viewer ratings. This method is especially an important factor in social media monitoring to gain a wider public opinion about a topic. NLP tools available today can be used to efficiently analyze this raw text and classify texts as positive, negative, or neutral. Different machine learning algorithms like random forest, logistic regression, and support vector machine can be used to train the model using the features extracted by the NLP techniques. The trained model can then be used to predict the polarity of the raw input data. ROC and PR curves have been plotted to check the accuracy of the algorithms.

Keywords Sentiment analysis · Random forest · Logistic regression · SVM

S. Srivastava (✉) · M. Vergin Raja Sarobin · J. Anbarasi · N. Sankaran
School of Computer Science and Engineering, VIT Chennai, Chennai 600127, India
e-mail: sanskriti.srivastava2016@vitstudent.ac.in

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_40

1 Introduction

In the digital era, public opinion is an important factor, and people are becoming more and more vocal about their opinions on social media and various other platforms. For business organizations and even governments to function smoothly, public opinion is an important factor to be considered. Most reviews and opinions given by the public are in the form of raw and unstructured text. Organizations and governments are trying to make sense of this raw and unstructured text to reveal patterns and derive useful conclusions that can help boost their success. Take the example of hotel reviews, multiple views about the hotel can be found on multiple platforms which can help people for deciding on a restaurant. However, it is difficult and time consuming for customers to go through reviews over multiple platforms before reaching a conclusion. The purpose of sentiment analysis is that it is possible to efficiently analyze this large volume of raw textual data to predict the overall customer feedback of the restaurant. Similarly, Twitter is another important platform where people post their opinions about various topics. Monitoring social media platforms like Twitter can help the government keep track of the public's currents interests, likes, dislikes, and opinions. Sentiment analysis is majorly employed in social media monitoring. With the fast-moving world, it is hard to keep track of all the data that comes up on social media platforms. Text sentiment analysis or opinion mining helps a lot in understanding the likes or dislikes of the majority.

Large volumes of data are being generated and shared every single day. The issue faced with this data is that along with being large in volume, they are also unstructured and improper statements. It is not necessary that all the reviews given by the users are properly framed and structured. Most users prefer using slangs, short-forms, touches of sarcasm, and emoticons to express their views, and these result in difficulties in decoding the emotion of the text. Sentiment analysis gives sense to this humongous dataset and tells us the inclination of the data that whether it is positive or negative. This can be achieved using NLP tools, which analyzes the emotion the user is trying to convey through the text. Different types of machine learning algorithms have been used to create models that can be used for predicting the sentiment of the data. The machine learning algorithms used are random forest, logistic regression, and support vector machine. The efficiency and time taken by each algorithm have been compared to reach the expected output.

In the proposed work, the dataset used for analysis is hotel review and Twitter dataset. The hotel review dataset has 5 lakh records of user reviews, and the Twitter dataset has around 14 thousand records of tweets. The main reason for choosing two datasets with such a huge difference in size is to analyze the behavior of each model toward the variant sized datasets. This data is used by most of the organizations to understand the trend, sentiment, or opinion of the people. The proposed work includes the collection of dataset and preprocessing and classification of the opinion using machine learning algorithms.

2 Literature Review

Facebook posts from official diabetes support groups have been crawled for a total of 15,000 pre-processed posts, 800 of which have been manually annotated by human experts, first classified according to Plutchik's wheel of emotions, consisting of eight dominant emotions. Detection of emotions using a string-based vector is better than the numerical vectors alone [1]. Emotions are also detected using Twitter data. These authors automatically detected emotions on Twitter messages and identified the characteristics using the LibLinear support vector machine template. Emotion mining has gained interest in the area of computer science because of the broad range of systems that are being developed [2].

An approach for the sentiment dictionary evaluation and correction is essential in engine development of sentiment analysis where F-measure has been increased by at most 1.24 times. In this corrective, the correction information analyzed is around 200 events. Although there are few verified data, it was possible to improve the validity through two different correct answer sources [3] by extracting the key aspects from the comments and identified the related emotions. This method adopts laws and lexicons for language processing to overcome multiple challenges in the study of emotions to achieve condensed outcomes. According to the reported results, when the implicit aspects are considered, the aspect extraction accuracy improves significantly. The unified classification system often outperforms the mean precision of the lexicon-based standard and the other combined rules by 5% [4].

Sentiment analysis for the Indian Languages—Hindi, Bengali, and Tamil using NLP preprocessing, feature generation, matrix construction, etc., is performed by [5]. The common goal was to classify tweets as positive, negative, and neutral in three polarity groups. The generated features include binary features, and statistical features generated using SentiWordNet and the presence of unigrams. The proposed method has less accuracy in correctly classifying Twitter test tweets into positive, negative, and neutral polarities due to the lack of language-specific feature and feelings-related feature [5]. Analysis of online social network micro-blogging Twitter includes more than 0.25 million tweets containing a reference to either a political party or an election politician held in India in April 2014. Analyzing the sentiment of the tweets regarding political parties and predicting the outcome for the elections using opinion mining [6] was proposed by these researchers.

In the proposed work, it contrasts neural network-derived sentiment classification approaches (backpropagation neural network (BPN), probabilistic neural network (PNN), and homogeneous collection of PNN (HEN)) utilizing different levels of word granularity as parameters for the classification of category level sentiment. Probabilistic neural networks (PNNs) outperform in classifying the feeling of product reviews among the two neural network methods used. The integration of neural network-based sentiment classification methods as a feature reduction technique with main component analysis (PCA) also provides superior performance in terms of training time [7, 8].

Sentiment analysis in Facebook and its application to e-learning was analyzed through SentBuk's classification system: a hybrid approach that incorporates methods from lexical and machine learning. The results obtained through this method show high reliability (83.27%) in conducting sentiment analysis on Facebook. It is very helpful to have knowledge about the emotions of the clients in the sense of e-learning [9]. Specific methods for data distillation from vast quantities of unstructured material are also performed. This article [10] discusses the basic characteristics of using linguistic variations in the identification of sentiments. Investigating all opinion extraction strategies in this study to produce positive and negative aspects of information with the correct feature set will help to reduce misclassification errors [10]. In [11], the authors have given the application of artificial intelligence techniques for the classification of heart disease data.

3 Proposed Work

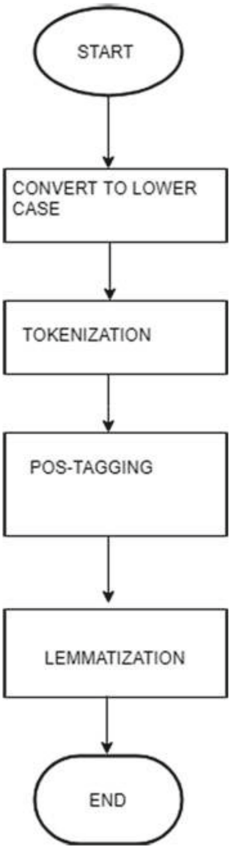
The proposed work includes data preprocessing where the collected raw data is cleaned and arranged into a format suitable for further processing. This processed data is analyzed, and elite features are extracted for classification into three polarity models. Random forest, logistic regression, and support vector machine classification are the algorithms used in this paper for classification and prediction.

3.1 Data Preprocessing

The data preprocessing includes representation of data knowledge, cleaning of the data so that features can be extracted and analyzed for better knowledge mining.

a. Data Knowledge Representation
The hotel review dataset consists of reviews and ratings given by visitors at a hotel, whereas the tweets datasets contain tweets extracted from Twitter about an airline. Due to the large size of the data, it has been sampled in order to speed up computation. All reviews are labeled with a rating 5 and above have been labeled as good and others as a bad review. The main objective is to predict the raw text review as good or bad. "No Positive "and "No Negative" reviews are removed as this may result in anomalies later.
b. Cleaning the Data
NLP tools such as NLTK and Spacy are used for the text cleaning process. The text cases all converted to lower case for all the available data. The text is tokenized (split the text into words) followed by the removal of the punctuation also the numbers and words like "the", "a", "this", etc. Stop words and empty tokens are also removed through the tokenization process. Grammatical tagging is performed which makes text to correspond the part of speech like noun, verb,

Fig. 1 Data processing



preposition, adjective, etc. POS tagging has been done with the help of NLTK tools. It assigns a tag to every word like a noun, verb, etc., using the WordNet lexical database. Lemmatization converts every word into its respective root form. For example, ate to eat, slept to sleep, etc. The Spacy tool is used to perform lemmatization as this is more efficient than NLTK for lemmatization. Figure 1 shows the steps performed for data preprocessing.

3.2 Feature Extraction

The sentiment analysis feature is extracted using Vader, which is part of the NLTK module. Vader uses a lexicon of words to find the positives and negative words. It also considers the context of the sentences to determine the sentiment score. For each text, Vader returns four values that are neutrality score, positive score, negative score, and an overall score that summarizes the previous scores. These values are integrated and fed as input to the machine learning algorithms as features to perform analysis followed by prediction. Term frequency-inverse document frequency checks the maximum number of times a word is occurring in any dataset and converts the words into vectors. Some words occur frequently in the dataset like good, bad but it is always not necessary that the words that occur frequently are the most important to the document. TF-IDF computes the relative importance of the words with respect to the context of the document rather than simply counting the occurrence of each word. Term frequency checks for the number of times a word occurring in a dataset, and inverse document frequency checks how important the word is in the dataset.

Vector representations of every review are performed which is a representation of document vocabulary. It can capture the context of a word in a document, semantic and syntactic similarity, relation with other words, etc. Word embedding is the texts converted into numbers, and there may be different numerical representations of the same text. Word embedding is the collective name for a set of language modeling and feature learning techniques in natural language processing where words or phrases from the vocabulary are mapped to vectors of real numbers. Word embedding converts the text into a vector form. If we have the vector forms of two different documents, its similarity can be predicted. For this project, Doc2Vec has been used with the help of the Gensim library in Python.

3.3 Classification and Prediction Using Machine Learning

The extracted features are classified using random forest, logistic regression and support vector machine to predict the data.

a. Random Forest
   Random forest was developed by Tim Kam Ho that uses the random subspace method, which creates the forest with several trees. In general, the more trees in the forest, the more robust the forest looks like. In the same way in the random forest classifier, the higher the number of trees in the forest gives the high accuracy results Approximately, 100 trees were created in the processed scheme. Figures 2 and 3 show the ROC curve, and PR curve obtained for the hotel review data.
b. Random Forest Pseudocode:
1. Randomly select "k" features from total "m" features where $k \ll m$.
2. Among the "k" features, calculate the node "d" that are the branches.
3. Split the nodes, consider it as a root node and repeat splitting on $n - 1$ features.
4. Repeat 1–3 steps until number of branches is reached to be 2.
5. Build forest by repeating steps 1–4 for creating "100" number of trees.
c. Logistic Regression
   Regression classification applies a logistic function to a linear combination of features to predict the category based on the dependent variable. The logistic regression algorithm helps to estimate the probability of a value falling into a category depending on the prediction variable. Binary categorical regression

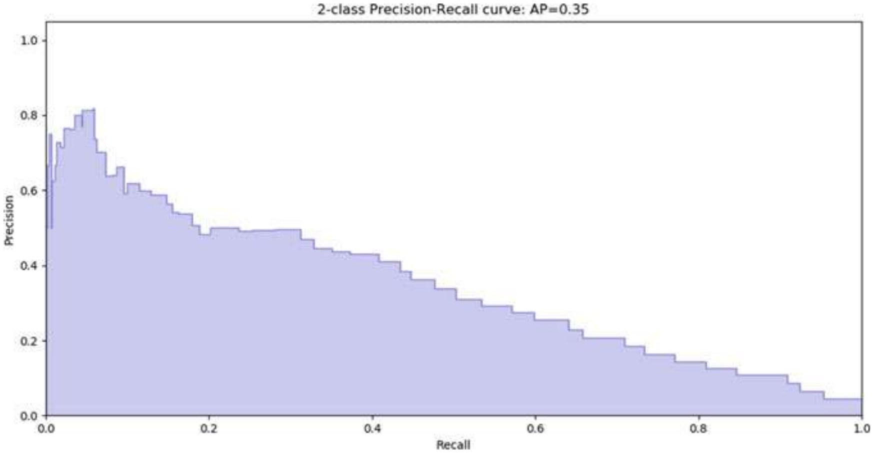


Fig. 2 Random forest PR curve (AP = 0.35)

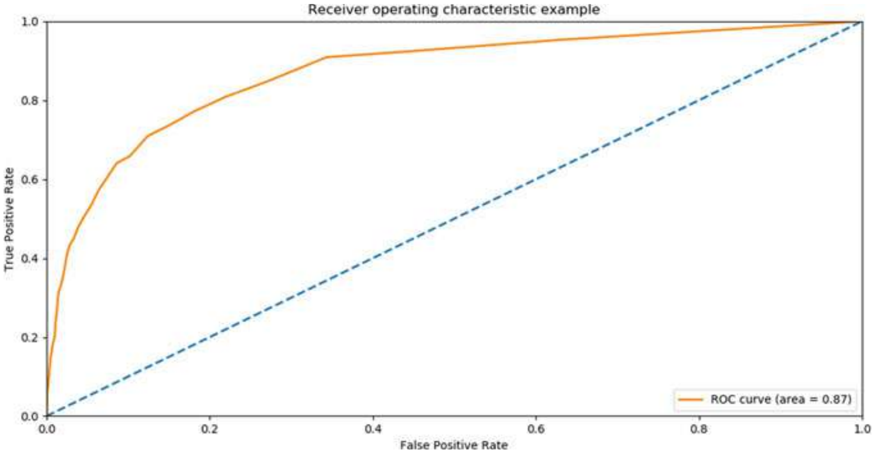


Fig. 3 Random forest ROC curve (area = 0.87)

is used in the proposed work to aim to classify the reviews as good or bad. Figures 4 and 5 show the ROC curve, and the PR curve obtained for the hotel review data.

d. Logistic Regression Pseudocode:

```
1. Assign variable $X1$ AND $X2$ and predictors $Y1$, $Y2$ … $YN$
2. Formulate the regression function using prediction variables
3. Identify the odds of the function to calculate the probability score
4. Classify into the appropriate category based on probability score
```

e. Support Vector Machine

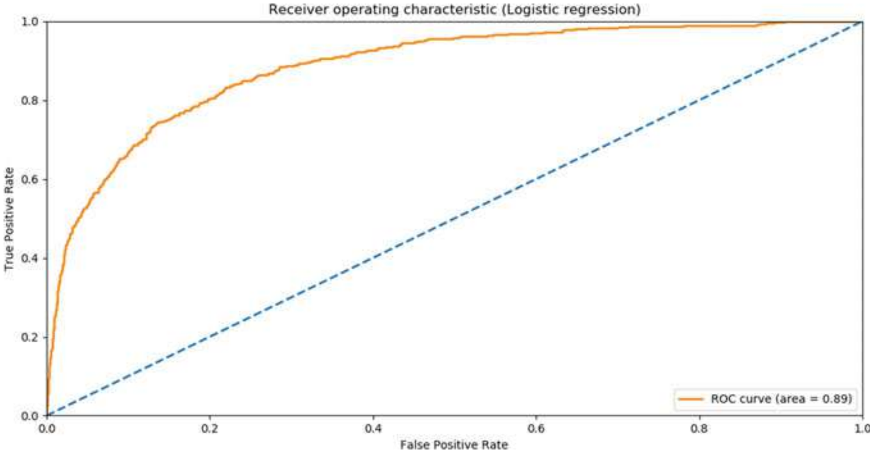


Fig. 4 Logistic regression ROC curve

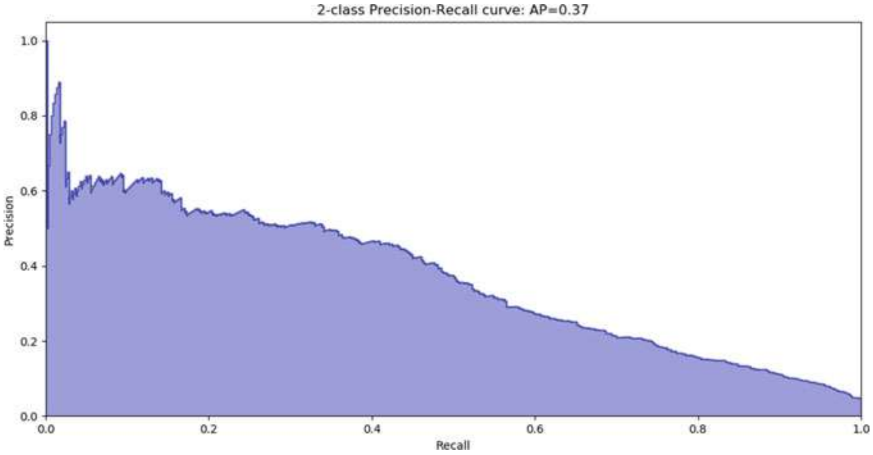


Fig. 5 Logistic regression PR curve (AP = 0.37)

Support vector machine is a supervised machine learning algorithm that can be used for the classification of datasets. It works by segregating the data into different classes by finding a line that separates the training dataset into classes. There can be numerous such lines that can split the data into two classes. SVM tries to find the line that has the greatest margin between the values of the two classes. SVMs are of generally two types, linear SVMs—in linear SVMs, the training data, i.e., classifiers are separated by a hyperplane and nonlinear SVMs—in nonlinear SVMs, it is not possible to separate the training data using a hyperplane. The proposed work uses linear SVM. Figures 6 and 7 show the ROC curve, and PR curve obtained through SVM.

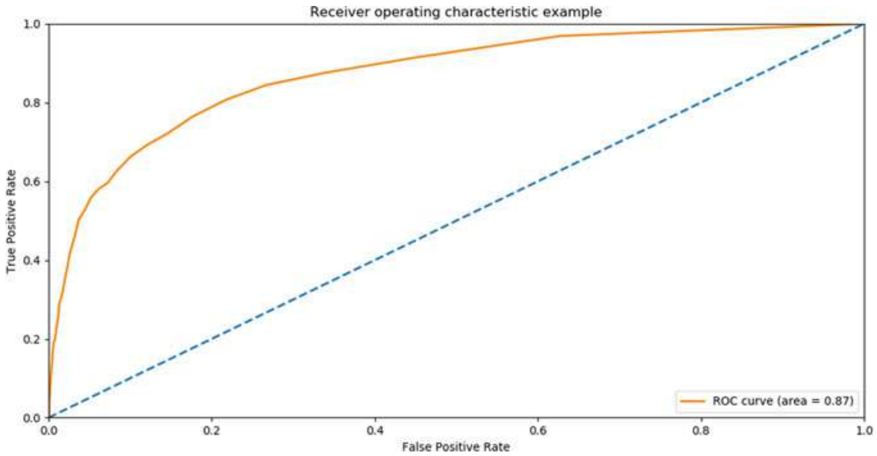


Fig. 6 Support vector machine ROC curve

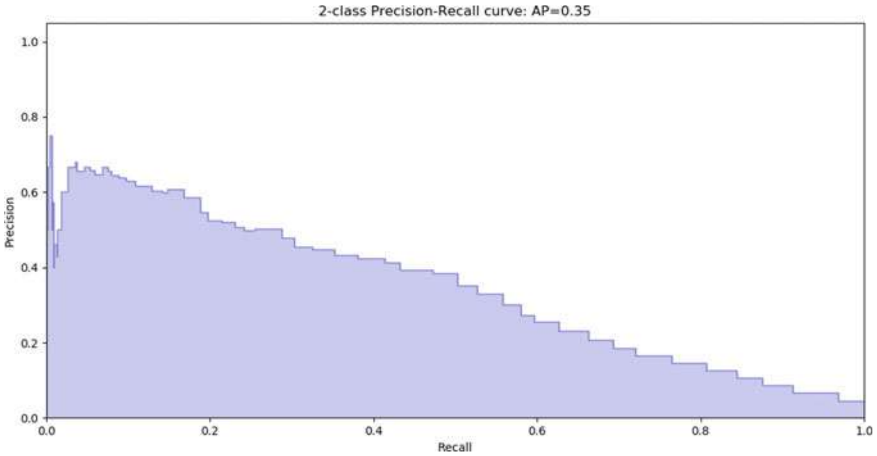


Fig. 7 Support vector machine PR curve (AP = 0.35)

Support Vector Machine pseudocode:

```
1. Let input training pair samples be $x1$, $x2$… $xn$, and the output result be defined as $y$.
2. Identify the right hyperplane by using the input features through the widest margin
3. Output value of the linear function is considered for classification.
```

4 Observations and Results

ROC curve and PR curve are analyzed to check the accuracy of the prediction model. The ROC curve is a pretty good method to analyze the quality of the model as well as gives a good visual view for accuracy. The higher the curve the more accurate our model is. The PR curve is used in situations of imbalanced dataset to measure the accuracy. The PR curve plots the precision and recall of the prediction model.

The hotel dataset analyzed has a user rating ranging from 0 to 10. All reviews with a rating lower than 5 have been classified as negative and all others as positive. The raw input text review is modeled to classify the review as positive or negative. Using NLP tools, features have been extracted that can be used to train the prediction model. The accuracy values for hotel review data are given in Table 1, and for Tweets, data is given in Table 2.

The random forest model achieved a good prediction, and the ROC, PR curves depict the same. The measure of the accuracy of the ROC curve is 0.87. The precision-recall curve, high precision-recall should be low, and the accuracy achieved for AP metric is around 0.35. The logistic regression model is used to train our prediction model, and the ROC curve evaluation of the model shows that the area under the curve is 0.89 (i.e., the measure of accuracy). The precision-recall curve shows for high precision the recall must be low and the AP metric achieved is around 0.37. SVM algorithm achieved the measure of accuracy using the ROC curve to be 0.87.

The tweets dataset used has a user rating ranging from 0 to 10. All reviews with a rating lower than 5 are classified as negative and all others as positive. Extracted features are used to train the prediction model. Compared to the hotel reviews dataset, the Twitter dataset is much smaller. Random forest algorithm, the ROC Curve measures that the area under the curve is 0.81 (i.e., the measure of accuracy). Since this dataset unlike the previous is imbalanced, no need to use the PR curve to analyze the accuracy. The logistic regression algorithm checks the accuracy of the predictions using the ROC curve and achieved 0.91. SVM algorithm achieved the measure of accuracy using the ROC curve to be 0.89. It is observed that the logistic regression model can give better precision in both cases, and also, it is seen that

Table 1 Precision for hotel review dataset

Algorithms	Accuracy
Random forest (100 trees)	0.87
Logistic regression	0.89
Support vector machine	0.87

Table 2 Precision for Twitter dataset

Algorithms	Accuracy
Random forest (100 trees)	0.81
Logistic regression	0.91
Support vector machine	0.89

almost all models give a prediction accuracy ranging from 0.8 to 0.91 which is pretty high which proves that the text analysis done is pretty good.

5 Conclusion

The project successfully predicts the sentiments of the reviews through different machine learning algorithms (random forest, support vector machine, and logistic regression), and the accuracy was calculated for every algorithm in the proposed work. The accuracy measure for the hotel review dataset through the random forest algorithm is 87%, logistic regression is 89%, and SVM is 87%. The Twitter data gives prediction results for the random forest as 81%, logistic regression as 91%, and SVM as 89%. Logistic regression gives maximum accuracy for hotel review data and Twitter data. The comparative values for sentiments of the reviews through different machine learning algorithm are given in Tables 1 and 2 which takes values from the ROC curve and PR curve of the respective algorithms. The algorithms efficiently analyze the large volume of raw textual data to predict the overall customer feedback of the restaurant and Twitter data for the opinion of the people.

Acknowledgements We wish to express our sincere thanks and deep sense of gratitude to our project guide and professor, Dr. Vergin Raja, School of Computer science (VIT, Chennai) for helping us understand more about sentiment analysis. The consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

References

1. Balakrishnan, V., Kaur, W.: String-based multinomial Naïve Bayes for emotion detection among Facebook diabetes community. Proc. Comput. Sci. **159**, 30–37 (2019)
2. Dhawan, S., Singh, K., Sehrawat, D.: Emotion mining techniques in social networking sites. Int. J. Inf. Comput. Technol. **4**(12), 1145–1153 (2014)
3. Nishiwaki, Y., Yoshida, Y., Teramra, T., Motoyama, A., Tsuda, K.: A consideration of evaluation method of sentiment analysis on social listening. Proc. Comput. Sci. **159**, 1314–1320 (2019)
4. Alqaryouti, O., Siyam, N., Monem, A.A. Shaalan, K.: Aspect-based sentiment analysis using smart government review data. Appl. Comput. Inform. (2019)
5. Kumar, S.S., Premjith, B., Kumar, M.A., Soman, K.P.: AMRITA_CEN-NLP@ SAIL2015: sentiment analysis in Indian Language using regularized least square approach with randomized feature learning. In: International Conference on Mining Intelligence and Knowledge Exploration, pp. 671–683, 2015 Dec. Springer, Cham
6. Mehndiratta, P., Sachdeva, S., Sachdeva, P., Sehgal, Y.: Elections again, twitter may help!!! a large scale study for predicting election results using twitter. In: International Conference on Big Data Analytics, pp. 133–144, 2014 Dec. Springer, Cham
7. Sharma, P., Moh, T.S.: Prediction of Indian election using sentiment analysis on hindi twitter. In: 2016 IEEE International Conference on Big Data (Big Data), pp. 1966–1971, 2016 Dec. IEEE.
8. Ortigosa, A., Martín, J.M., Carro, R.M.: Sentiment analysis in Facebook and its application to e-learning. Comput. Hum. Behav. **31**, 527–541 (2014)

9. Singh, J., Singh, G., Singh, R.: A review of sentiment analysis techniques for opinionated web text. CSI Trans. ICT **4**(2–4), 241–247 (2016)
10. Vinodhini, G., Chandrasekaran, R.M.: A comparative performance evaluation of neural network based approach for sentiment classification of online reviews. J. King Saud Univ.-Comput. Inf. Sci. **28**(1), 2–12 (2016)
11. Murali, M., Bhargava, M., Snehaa, G., Anand, A., Haque, M.A. Vergin M.: Data analytics on IoT-based health monitoring system. Int. J. Recent Technol. Eng. **8**(1), 220–223 (2019)

Internet Performance Profiling of Countries

Nikolay Todorov, Ivan Ganchev, and Máirtín O'Droma

Abstract The paper presents research into data capture and analysis techniques for creating spatial and temporal Internet quality of service (QoS) performance profiles of countries. Active Internet probing and traffic monitoring techniques are used, facilitated through the European RIPE Atlas project to capture raw QoS measurements. The research goal is to contribute toward developing large-scale network QoS performance profiling and testing methods for local to long-range and global-range Internet QoS performance analysis. The range of stakeholder interest in such profiles is wide, from network owners and Internet service providers (ISPs) to Internet service provisioning consultants, and corporate, business, and individual users. Also, applications are wide such as detection and location of temporal traffic bottleneck and faults, bottleneck incidence behavior as a function of geographic and temporal service demands, to Internet service level agreements (SLAs) and their policing. Twenty-six European countries are examined and profiled on the basis of a bidirectional north–south and east–west 'compass profiling' methodology over a one-month period. The worst-case scenarios detected are presented as an example. The results may serve as an initial benchmark for mapping evolving performance profiles over longer or continuous periods of time, employing more geographical spread testing probes, and a mix of profiling methodologies especially for verification purposes. A similar

This work has been accomplished with the financial support of the MES by the Grant No. D01-271/16.12.2019 for NCDSC part of the Bulgarian National Roadmap on RIs and the support of the Telecommunications Research Center (TRC), University of Limerick, Ireland.

N. Todorov
Ward Solutions, Dublin, Ireland
e-mail: niki.g.todorov@gmail.com

I. Ganchev (✉)
University of Plovdiv "Paisii Hilendarski", Plovdiv, Bulgaria
e-mail: ivan.ganchev@ul.ie

Institute of Mathematics and Informatics - Bulgarian Academy of Sciences, Sofia, Bulgaria

I. Ganchev · M. O'Droma
Telecommunications Research Centre (TRC), University of Limerick, Limerick, Ireland
e-mail: mairtin.odroma@ul.ie

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_41

approach may be taken to regional, international, and intercontinental Internet QoS profiling.

Keywords Internet tomography · Internet performance · Quality of service (QoS) · Profiles of countries · Compass profiling

1 Introduction

The Internet continues to evolve rapidly, in a largely open environment, being now a massive geographically distributed internetwork with continued exponential growth of users, service providers, and bandwidth-hungry services. National and international network infrastructure providers and Internet service providers (ISPs) need continuously to assess and monitor the quality of service (QoS) of their networks and backbone transport and interconnect environments. This includes monitoring packet loss rates, bandwidth utilization, delay and jitter distributions, throughput, reliability, and other Internet Protocol (IP) QoS metrics [1, 2]. Quantitative assessment of network performance is extremely challenging because of the native heterogeneous and largely unregulated structure of the Internet. Tasks such as dynamic routing, optimized service provision, service level agreement and verification (SLA & SLV), and detection of anomalous/malicious behavior are challenging as, for many good commercial reasons, one cannot rely on the cooperation of individual servers and routers to aide in the collection of 'big-picture' network traffic [1, 3].

The need for tools and techniques for monitoring network utilization and performance locally, regionally, internationally, and intercontinentally stimulates the *network tomography* discipline. The demand for this kind of support has increased exponentially in recent years among stakeholders such as network owners and ISPs through to Internet service provisioning consultants, and corporate, business, and individual users. The expansion of telephony and data services, video-conferencing, working-from-home, online gaming, virtual schools, universities and education generally, and multimedia multi-party streaming services all require and demand high-level QoS performance; and, among the network owners and administrators, the ISPs and corporate user entities, in particular, legally-binding QoS performance guarantees are needed [4, 5]. For all of the latter, it is very important to keep informed about, and be vigilant in regard to, live network QoS, e.g., on hourly, daily, weekday and weekend, office and out-of-office, and seasonal basis, and through all of these periods, what is happening in all parts of their networks so they can identify, or at least have reliable authentic information on, bottlenecks, faults, or network misconfiguration, whether these are transient or ingrained, local or regional, and so on.

The term *network tomography* was introduced in [6] to characterize two classes of inverse problems in computer and communication networks. The goal is to recover higher-dimensional network information from lower-dimensional data [7]. The first class deals with *passive* tomography where aggregate data is collected at the router level with the goal of recovering path-level information [4]. The most common application is the estimation of the origin–destination (O–D) traffic matrix of a network. The second class is the *active* tomography which deals with reconstructing link-level information from end-to-end (E2E) path-level measurements by sending packets from a source to several receiver nodes, all located on the periphery of a network. The goal is an estimation of QoS parameters such as loss rate and delay distributions [4]. In any of these cases, statistical models should be developed for the measurement process to yield desired information [3].

The aim of the conducted research, presented in this paper, was to test out an approach to capturing the Internet performance profiles of selected countries, both spatially and temporally, by using *active* tomography methods and techniques, and by implementing flexible yet tractable models which simultaneously account for long-term traffic dependence, latency, dynamic random routing, and spatial dependence [3]. This task is possible through collaborations with major coordinated global activities in this area. Of particular importance is the access to measurements collected in the very-large integrated Internet probing and QoS measuring done within the Atlas project coordinated by the Réseaux IP Européens (RIPE) [8]. This project encompasses thousands of active probes concentrated in the RIPE NCC service region (Europe, Middle East, and parts of Central Asia), capturing, for instance, dynamic profiles of the Internet connectivity and reachability in coarse or fine grained, spatial and temporal resolution over Internet paths or logical connections. RIPE's main goals are to produce Internet 'traffic maps', provide users with active measurement tools for baseline and on-demand measurements, and act as a trusted source, and an exchange point, of data regarding real-life, active measurements. Through this, an unprecedented understanding of the state of the Internet in real time is possible.

Minimally invasive RIPE measurement methodologies, based on software tools for generating probes and monitoring traffic flows, statistical modeling of the measurement process, and sampling strategies for online data collection [1], were used in this research. Model validation was focused on the application to real network data. By using available measurements databases and probing tools, monthly round-trip time (RTT) trends on selected routes were obtained by means of *ping*- and *traceroute*-based measurements as per the methodology described in [9]. After extracting the raw data from the measurements, performance profiles of selected European countries could be and were constructed. This was made possible with the generation of bar graphs and the calculation of congestion level index (CLI) [9].

The rest of the paper is organized as follows. Section 2 provides some background information. Section 3 presents the related work done in this area by the two major organizations—Center for Applied Internet Data Analysis (CAIDA) [10] and RIPE [8]. Section 4 describes the methodology used in the conducted research, whereas Sect. 5 analyzes some of the obtained results. Finally, Sect. 6 concludes the paper.

2 Background

The *passive* approach for assessing network performance, based on detailed queuing data and models at the individual router level, is not enough for capturing the overall network performance due to the complexity and size of modern networks. Estimating link-level QoS parameters requires access to the internal links and routers. As such access would require agreement and support of their owners, it is an approach which is resource-intensive and non-scalable and hence making the collection of detailed information at this level impracticable, including where networks themselves are capturing this information for their own network monitoring services. As a result, the external reconstruction of link-level data from path-level E2E measurements involves sending *probe packets* from a source to one or more destinations [7], i.e., by using *active* network tomography methods and techniques. The probe packets are sent using a (simulated) multicast communication protocol where a single packet sent by the origin is duplicated at each branching point on the way to the destinations. The inverse problem is to recover the delay distributions and loss rates associated with the interior network nodes from the E2E measurements [2, 3]. As indicated, characterizing these parameters enables the detection of bottlenecks, faults, congestions, and other anomalous behavior as well as serving applications such as ensuring compliance with SLAs for users (in the broad sense) [5] and for managing overlay networks [7, 11].

The term *active measurement* means the injection of (artificial) probe traffic into the network in a designed and calibrated way and extract from the behavior of this traffic, measurement of network characteristics at different points and along different paths and links. The main idea is that the initial structure of the probe traffic is known, and by measuring how it is affected by the section of the network it goes through, network conditions can be inferred. In the active E2E measurements, the probes are sent between controlled ‘origin–destination’ (O–D) pairs, and the measurements are made at those origins and destinations only [12]. The approach thus can be classed as invasive. To minimize probe traffic influencing what it seeks to measure, usually low-rate probe streams (periodic or pseudo-Poisson) are used and the sampling is performed at low-time resolution [12]. On the other hand, active measurements have great flexibility as the set of O–D pairs can be chosen and many different kinds of probe traffic can be selected, including packet streams which imitate different traffic types [13], the latter with a view to different goals to network QoS profiling. The results from the simulations made in [14] show that active probing considerably reduces the size of the probe set when compared with pre-planned, or passive, probing.

Another advantage of the active measurements is the flexibility to plan the probe streams in a way that match different measurement requirements. The measurement focus can be shifted from determining delay and loss on a route, to estimating bottlenecks [15], the available bandwidth [16], and the rate of the background or cross traffic [17]. Other advantages are the possibility of performing, repeating, and varying experiments as often as desired, great reduction of the amount of measurement data compared to the passive monitoring of high bandwidth links, and the avoidance of data privacy issues. Depending on the deterministic and/or statistical criteria, the design parameters can be chosen with different sequence of packet types, sizes, and inter-departure times [12].

The research presented in this paper is focused on measuring the delay, a key end-user QoS parameter for certain services such as real-time interactive teleconferences, distance education, and webinars, which have grown exponentially through the COVID-19 pandemic. *One-way* delay measurements are very useful for applications that are intrinsically one-way, as well as for detecting bottlenecks. However, this requires highly synchronized network clocks [12]. This contrasts with measuring the RTT, which is 'one of the key factors determining the performance of interactive applications and the efficiency of feedback based control mechanisms' [12]. Metrics for the measurement of one-way delay and RTT are standardized by the Internet Engineering Task Force (IETF) in RFC 7679 and RFC 2681, respectively.

The basic variables in the active probing approach are the communication protocol used to send the probes, the packet contents, the sending process, and the actual measurement tools used. There is big variety of possible probe streams generated as the sending process can be chosen from many stochastic and deterministic models.

Selecting the protocol is very important, as its messages could be treated differently by the network nodes. One popular choice for measuring packet loss and delay is the Internet Control Message Protocol (ICMP), which was designed for the exchange of control and management messages between network nodes (hosts and routers). Other protocols can be used as well, but the ICMP packets are much better suited for more specific purposes [12]. A disadvantage of ICMP-based measurements in the public Internet, however, is that most sites rate-limit or even block ICMP packets passing into their networks to prevent attacks or to hide their internal topology. The ICMP packets directed from routers are also frequently given lower priority than IP packets. For overcoming these problems, special measures must be taken such as setting up and using special probe infrastructures across the Internet, e.g., as part of big projects such as RIPE Atlas or CAIDA Archipelago, with an ability to utilize the capacity of ICMP to a full extent by using relevant measurement tools and techniques.

Various ICMP-based tools for measuring the packet delay and loss exist. The most common and simple among them is the *ping* tool available on a variety of operating systems. It is used to test the reachability and availability of nodes. In addition, it returns the current values of RTT and packet loss. Another popular ICMP-based tool is *traceroute*, which reconstructs path information using a sequence of IP packets with an incremented *time-to-live* (TTL) value. Both small- and large-scale measurements could be made using *ping* and *traceroute*.

3 Related Work (CAIDA vs RIPE)

The CAIDA [10] and RIPE [8] are the two largest organizations involved in this area of research of Internet performance and QoS data collection.

The primary focus of CAIDA is the development of measurement, analysis, and visualization tools in order to benefit the community and advance the robustness and scalability of the Internet. The tools are intended to be made publicly available as well as the data acquired from public sources to be made available to researchers and other interested parties through the CAIDA website.

CAIDA helps organizations to jointly address areas of common interest on topics related to the infrastructure level measurements and analyses, scaling up the performance of the Internet through the evolution of the supporting technologies, promoting the understanding and characterization of traffic behavior using active and passive measurements along with analysis of routing data, affecting individual and global Internet infrastructures, help ISPs manage their networks with improved traffic analysis and visualization tools, etc.

Archipelago (Ark) is an active measurement infrastructure developed by CAIDA since 2007, involving multiple Ark monitors distributed in many countries. It is formed specifically for active network measurements with the main goals being the reduction of the effort needed to develop and deploy sophisticated large-scale measurements and providing a step toward a community-oriented measurement infrastructure. For measurements requiring highly precise time synchronization over the Internet, the Ark monitors use RADclock. The Ark monitor itself is a small, inexpensive measurement node, based on the Raspberry Pi and provides a flexible Linux-based programmable platform for conducting networking research. Using a process called team probing, monitors are put into groups and the measurement work is divided up between the group members. This is done so as to create a coordinated large-scale topology for performing measurements by using a powerful and flexible active measurement tool, called *scamper*, which supports *traceroute* and *ping*. The results are then distributed as an IPv4 Routed /24 Topology Dataset. An IPv4 Routed /24 autonomous system (AS) Links Dataset is also provided, which includes the AS links obtained from the IP paths of the topology dataset, along with few other publicly available datasets (for both IPv4 and IPv6).

While powerful, the Archipelago project does not yet support customized measurements. Individuals hosting an Ark monitor are not able to configure or perform specific measurements on their own. If this aspect is incorporated into the future development of the project, it would be a big step forward as it will catch the attention of wider audience, allowing people to conduct their own measurements with almost no effect on the network performance and to various destinations that might not be within the scope CAIDA is covering at the moment.

The RIPE Network Coordination Center (NCC) is an independent, not-for-profit membership organization that acts as a regional Internet registry (RIR). RIPE NCC supports the infrastructure of the Internet and provides global Internet resources and services in its service region such as allocation of IPv4/IPv6 addresses and

AS numbers. The organization also manages one of the 13 root name servers (K-root), coordinates the support of ENUM delegations, deploys a routing database, and manages a neutral measurement network that offers freely accessible and reliable statistics on the operation of the Internet.

Introduced by the RIPE NCC, the RIPE Atlas is the next-generation Internet active measurement network. Built from thousands of probes distributed around the globe, it measures connectivity and reachability, collects behavior data, and provides valuable live and historical information about the network reliability. All measurements are pooled and provide an unprecedented understanding of the state of the Internet, from local to global, in real time. The development of the project started in late 2010 and is built on the previous RIPE NCC's measurement network—the Test Traffic Measurement Service. The measurement network is constantly growing, and the data collected is providing useful maps and graphs that help building a big-picture view of the Internet.

The Atlas probe is a small USB-powered hardware device, running software developed by the RIPE NCC and attached to an Ethernet port of a router via a UTP cable. Once powered up and connected to the Internet, the probe acquires an IP address and DNS resolver information using DHCP. A static network configuration can also be assigned if the host does not want to use the dynamic one. In order to allow the probe to connect to the network, the minimum array of services that need to be enabled are DHCP, DNS, and outgoing TCP port 443 (HTTPS) so as to allow the probe to do measurements. When all this is done, the probe starts performing its built-in measurements which include *ping* and *traceroute*, SSL/TLS queries to some predetermined destinations, DNS queries to root DNS servers, etc. The bandwidth consumed by the probe depends on the number of measurements the probe is running at any given time but it also has the option for the host to specify the maximum bandwidth the probe can use.

Everyone hosting a probe can conduct their own user-defined parameters. The host has access to all the probes on the network and can run measurements from them to destinations of its choice. Once the probe is connected, the host starts earning credits for the time that the probe is connected. It should be noted that a probe is considered 'connected' only when it has a working Internet connection and is connected to the RIPE Atlas infrastructure. A host earns credits for each of their active probes, if more than one. The collected credits can be used for user-defined measurement either from the hosted probe itself, or from any other probe in the RIPE Atlas network. When a user logs into their account on the RIPE Atlas website, they can check their credit balance or start a user-defined measurement. Of course, there are limitations as to the number of measurements, probes, and credits that can be used. At the time of conducting this research, the maximum permissible amount of simultaneous measurements was 100, with no more than 500 probes used per measurement and no more than 270,000 credits used per day. There are, however, enhanced RIPE Atlas probes with greater measurement capacity than the regular probes and they are called *anchors*. These not only perform more measurements, but also act as stable, cooperating regional targets for measurements within the RIPE Atlas network. They allow users to observe the traffic at both the source and destination, providing valuable information about the local and regional connectivity and reachability of the Internet. The data collected is made publicly available. Organizations hosting RIPE Atlas anchors receive additional benefits as a return of their contribution.

The RIPE Atlas provides users with an active measurement tool for baseline and on-demand customized active measurements from thousands of vantage points across the Internet and records the responses. The technical community can use the data and the Internet 'traffic maps' produced for different purposes. This neutral, transparent data collection infrastructure provides data sets that can help analyze the operation and growth of the Internet, using active data collection methods it offers and timely data such as country-wide Internet outages or measurement data on the state of the Internet. The data collected is of a great value to the network operators, researchers, the technical community, and anyone with a professional or other interest in the healthy functioning of the Internet. It can also help people understand and learn more about the underlying networking infrastructure and the data flows which keep the Internet running globally. For this reason, the development of the RIPE Atlas project has been guided by the users. RIPE NCC continues to expand the command, control, and collection infrastructure to accommodate the growing number of probes and create new and enhanced Internet traffic maps, as well as integrating the RIPE Atlas data into other tools, such as RIPEstat (https://stat.ripe.net/).

4 Research Methodology

For the purpose of this research, a choice needed to be made as to which measurement infrastructure and dataset to employ. After comparing the CAIDA Archipelago and the RIPE Atlas projects, it was decided that the more suitable dataset for this research would be RIPE Atlas's, especially as at present independent customized measurement campaigns are not facilitated on CAIDA Archipelago.

Our idea was to pilot test a system for building Internet performance profiles of selected countries within a European area by extracting raw QoS data. To capture traffic measurements, a probe network was constructed of selected RIPE Atlas probes located in 26 European countries. In each country, four probes were sought and selected based on their availability and their geographical positioning being at the furthest east, west, north, and south point of the country. Where probes did not exist in a specific compass location, or were not available, that test dimension was omitted from the profiling.

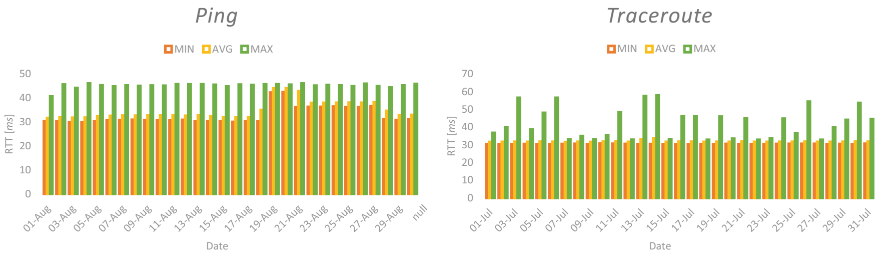


Fig. 1 *Ping* (left) versus *Traceroute* (right) day-to-day minimum, average, and maximum RTTs for Denmark (‘north→south’ direction)

With the probe network in place, *ping*- and *traceroute*-based measurements were performed for the duration of one month[1] in each of the selected EU countries,[2] and then analyzed in order to reveal Internet QoS performance variations.

Traceroute-based measurements were taken first. The results were presented in bar graphs, considering the RTT from the destination. If the destination was unreachable, then the results from the last responsive hop were taken. The only downside of the *traceroute*-based measurements is that it cannot be checked if the destination is reachable or not, while the measurement is running. The only thing that can be done is to check regularly if the destination probe conducting the measurement is active or not. In case it is not, a new probe, if available, was selected as soon as possible in order to continue the measurement.

The second group of measurements then taken were *ping*-based measurements, using the same source and destination address as in the *traceroute*-based measurements. A useful feature of the *ping*-based measurements is that it is immediately noticeable if a destination is reachable or not, as shown on the measurement personal page. It was checked regularly to see if all the probes are active and destinations reachable. In case any of the probes were not active or the IP address was unreachable, then actions were taken quickly, and new ones were sought and selected to continue with the measurements.

Following this measurements campaign, the raw QoS data was extracted from the RIPE archive in the form of JSON and TXT files. Two Java programs were written specifically for the purpose of extracting the needed data from the raw *ping* and *traceroute* measurements. For both types of measurements, behavior may be visualized and patterns and dependencies observed, through the creation of bar graphs (Figs. 1 to 4), generated from the harnessed data. Using a confidence interval, the highest 5% of the RTTs were not taken into account. Each bar graph shows the minimum, average, and maximum RTT on a day-to-day basis for each day the measurement was run

[1] The period of one month for the measurements campaigns was also dictated by our available credit under the credit system employed for the RIPE Atlas project participants (https://atlas.ripe.net/landing/about/).

[2] The set of measurements was confined to EU countries due to certain limitations of how many measurements can be performed simultaneously within the RIPE Atlas project.

for a particular origin–destination (O–D) pair along a compass direction within each country. The minimum and maximum values are respectively the smallest and biggest RTT values for the specific day, and the average value was calculated by averaging all measured RTTs of the day.

Congestion level index (CLI) is a useful parameter for highlighting differential performance whether along specific (compass) paths or between paths in or across countries. The CLIs between the farthest measurement points within each country was calculated. First, the 'distance delay' was calculated as follows [9]:

$$\text{distance\_delay} = \text{geographical\_distance}/0.6c,$$

where c is the speed of light in a vacuum and 0.6 is a propagation factor used for the speed of light in single-mode optical fiber over that in a vacuum. Taking the one-way delay based on the RTT measurements, the CLI was calculated as [9]:

$$\text{CLI} = \text{one\_way\_delay}/\text{distance\_delay}.$$

Separate CLI values were calculated for each compass direction, based on the *ping*- and *traceroute*-based RTT measurements, respectively. These were averaged to obtain the final CLI value for each direction. Table 1 shows an example of CLIs calculation, based on the data for Denmark.

By comparing the calculated CLIs and considering how many measurements were conducted for the specific country, the countries were separated into six different CLI groups. The comparison of the CLIs was based on their symmetry. If one of the CLIs, for a group of measurements on a specific compass path (e.g., 'west→east' / 'east→west'), was found greater than twice the CLI in the opposite direction, then they were considered *asymmetric*. Otherwise, they were considered *symmetric*.

As a result, the following CLI groups were formed:

1. *Group 1*—with *asymmetric* measurements in all *four* compass directions ('north→south,' 'south→north,' 'east→west,' 'west→east'). The only country in this group was Denmark;
2. *Group 2*—with *symmetric* measurements in all *four* directions. Six countries identified in this group were Austria, Belgium, Bulgaria, Lithuania, Spain, and the UK;
3. *Group 3*—with measurements in all four directions, whereby one group of them (e.g., 'east→west' and 'west→east') is *symmetric* and the other one is *asymmetric*. Ten countries, Czech Republic, Finland, France, Germany, Hungary, Italy, Netherlands, Poland, Romania, and Sweden, formed this group;
4. *Group 4*—with *asymmetric* measurements for one group and no measurements or not enough data for the other one (e.g., asymmetric results for 'north→south' and 'south→north,' and no data for the other two measurements). Cyprus, Ireland, Latvia, and Luxembourg were identified as members of this group;

Table 1 Calculated CLI values for Denmark (for the purposes of presentation, all entries in the table are rounded to one decimal point)

Direction	Distance (m)	Distance delay (ms)	Ping			Traceroute			Final CLI
			Average RTT (ms)	Average one-way delay (ms)	CLI	Average RTT (ms)	Average one-way delay (ms)	CLI	
‘north→south’	186,078	1.0	35.0	17.5	16.9	31.8	15.9	15.4	16.2
‘south→north’			8.2	4.1	4.0	7.9	4.0	3.8	3.9
‘east→west’	123,509	0.7	45.0	22.5	32.8	39.1	19.5	28.5	30.6
‘west→east’			100.8	50.4	73.4	104.6	52.3	76.3	74.8

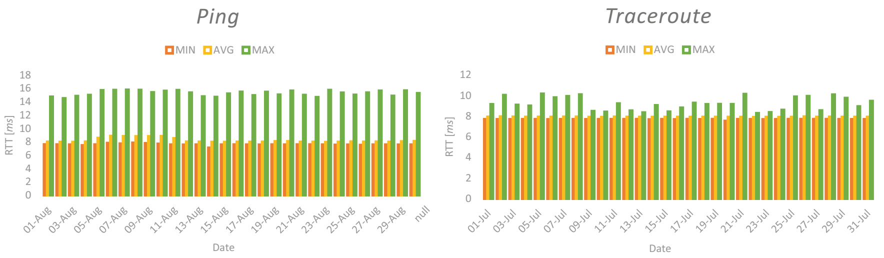


Fig. 2 *Ping* (left) versus *Traceroute* (right) day-to-day minimum, average, and maximum RTTs for Denmark (‘south→north’ direction)

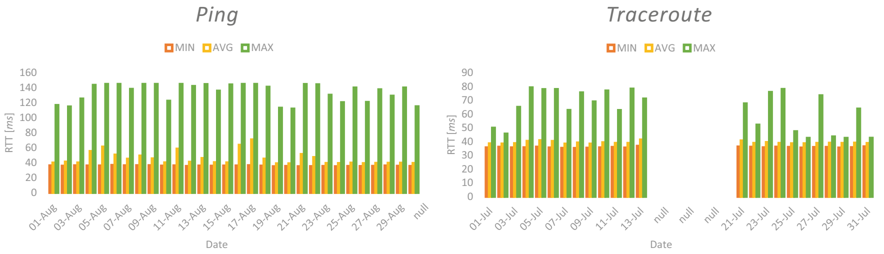


Fig. 3 *Ping* (left) versus *Traceroute* (right) day-to-day minimum, average, and maximum RTTs for Denmark (‘east→west’ direction)

5. *Group 5*—with *symmetric* measurements for one group and no measurements or not enough data for the other one. Greece and Portugal were the only members in this group;
6. *Group 6*—with measurements only in one direction and no data for the opposite one, so they can’t be compared to anything. The countries in this group were Estonia, Slovakia, and Slovenia.

Bar graphs were generated for each country, from both the *ping*- and *traceroute*-based measurements, giving more detailed information on the network performance throughout the whole month on a day-to-day basis. Of these, one example is presented here. Figures 1, 2, 3 and 4 depict the generated bar graphs for each of the four directions within Denmark. If a measurement for a day or a couple of days is missing, it is represented as ‘null’ in the graphs and the dates are skipped.

5 Results and Analysis

In this research, the focus was on the *round-trip time* (*RTT*) component of the QoS. The exact QoS requirements are hard to pinpoint, especially having in mind the fact that there was a wide variation based on the countries selected for the research.

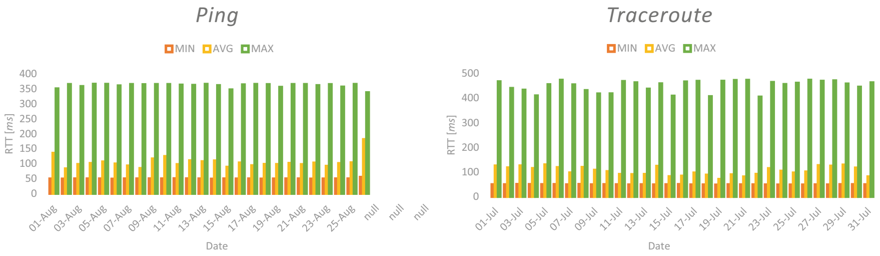


Fig. 4 *Ping* (left) versus *Traceroute* (right) day-to-day minimum, average, and maximum RTTs for Denmark (‘west→east’ direction)

Depending on the underlying network infrastructure, the service providers, the number of users and the traffic, and other service factors, comparing the QoS for two separate countries is a fraught exercise. The obtained results, however, may serve as a benchmark when conducting further measurements in future measurement campaigns. Comparisons could be made to discern improvements or new issues across the network. Therefore, the information acquired even in this pilot measurement program can be very useful to network administrators, ISPs, and any other interested party. With the help of the CLIs, one can perform investigations and compare the different results for a specific country, based on the *asymmetry* of the CLI results, and discern if there are bottlenecks or faults, or indications of these, in any of the regions. Simply, an *asymmetric* result illustrates a problem along the path the measurement was conducted. By exploring the CLIs, the bottleneck/fault problem can be located, examined, and alleviated where possible. In this case, it is easy to see which CLI of the country under consideration is bigger and that its path would be the one experiencing a bottleneck or fault problem. Exploring the bottleneck/fault could further facilitate the search for the issue causing the problem. Starting from cabling, switches, and routers, going through the network, up to the final connecting components located closer to the end-user, any possibility should be explored as any of these might be in poor condition and causing problems and unsatisfied customers.

Looking closely into the CLI groups, identified as part of the conducted research, the worst cases were explored, i.e., for countries where the CLI *asymmetry* for a set of opposite directions was the biggest. As an example of this analysis, Group 1 (containing only one country—Denmark) is presented in this paper, as in this group all CLIs are *asymmetric*. The CLI analysis indeed shows, there is ’Something is rotten in the state of Denmark’ [18] in regard to the ‘north→south’ and ‘west→east’ Internet communications performances. The CLI in the ‘south→north’ direction is considerably smaller than the one for ‘north→south’ (with the ratio of 1:4.2) and, similarly, the ‘east→west’ CLI is noticeably smaller than the ‘west→east’ one (with the ratio of 1:2.4).

Further exploration of the network problems within a particular country could be done by analyzing the day-to-day bar graphs. By exploring patterns and dependencies, one can conclude what days of the week or the month, the network was the busiest, congested or unavailable.

In countries with identified network problems, such as Denmark in this paper, the problem along the path can be located through further examination of the *traceroute* results. By analyzing the RTTs of the different hops, it is readily possible to locate the source of a problem and discern the requisite actions in order to improve the network performance.

Refinement of these performance behaviors can be achieved by other traffic volume pattern analyses. Such information would not be available of course through the RIPE Atlas QoS measurements, nor would it normally be made available by the network providers without specific agreements. More remote socio-economic drivers could be used to apply weights to raw QoS results, e.g., beginning with the obvious ones such as office hours, non-office hours, labor holiday periods, school holiday periods, etc. and on to more localized drivers such as large festival and sporting events. The research presented here does not include such remote-factor influencer analyses.

6 Conclusions

The research presented in this paper was focused on showing how *active* network probing strategy may be used for the analysis of performance of national networks. The innovative intra-country profiling approach focused on capturing the bidirectional compass-based QoS performance data between the four furthest north–south, and east–west points in each country. Being part of the RIPE Atlas project presented the opportunity to carry out such a measurement campaign. Successful performance mapping in each direction required appropriately placed RIPE probe points. The bidirectional compass-based methodology was just our arbitrarily chosen scheme for measuring a county's Internet QoS profile, decided on in the absence of any defined method or system for extracting a profile. *Ping*- and *traceroute*-based measurements on all chosen routes over full month measurement campaigns provided the raw QoS data.

Naturally profiling systems could be much more refined than that used here. Given the rather large variations from country to country and within countries, as discovered using our simple QoS profiling methodology, it would be advisable that other approaches be examined, e.g., a *rotating compass* methodology using many more probes, to test for consistency, and at minimum to verify the effectiveness of our scheme, or to help establish some confidence limits on the error bounds of these QoS profile measurements. A *wheel-spoke* system, centered on the capital city of a country, could be another profiling methodology.

Caution is urged here not to consider the results of our pilot measurement campaign as indicative of definitive country Internet QoS profiles. This is due not just to the use of a single compass-based methodology but also to the measurement campaign constraints, particularly the RIPE constraints mentioned above such as the one-month constraint, over which the measurement campaigns were carried out. However, the approach presented could be used as a QoS profiling methodology, or at least as an initial benchmark for such a methodology.

It should also be borne in mind, that all profiles will be dynamic for lots of reasons, such as the continual upgrading of networks on the one hand and the continued explosion of the variety of bandwidth-hungry services on the other, especially, in regard to this latter, the uptick in the very large real-time services such as teleconference services for meetings, education and so on which has been a phenomenon seen in the COVID-19 pandemic period.

Naturally also there are many stakeholders involved who would have an interest in accessing and monitoring such Internet QoS profiles such as ISPs, network infrastructure providers, service providers and individual and especially large corporation end-users. Their interest would vary. For instance, this information could be very useful not only to the ISPs in those countries but also to ISPs' customers and subscribers, and to IT and Internet consultants advising businesses and corporations on Internet services. For an ISP or network provider, for instance, as in the country example presented in this paper, by exploring the CLI values and the bar graphs generated from the raw QoS data extracted, one first can discern problems such as bottlenecks across the tested networks, and from there move to locate the bottlenecks and faults, assess and implement solutions and then assess the success or otherwise of solutions implemented. In time, experts will gain ever better understanding of the dynamic QoS performance for a particular network path(s), sub-net(s), and on to the whole network within a country, and likewise onwards to QoS performance statuses for that country's international connections.

The above form initial arguments for further Internet QoS profiling studies to be conducted for longer periods of time or over different paths of interest. With the help of a dedicated website built for the purpose, all measurements can be uploaded regularly, presenting the most recent results. They could be made easy to understand even for customers. Even performance comparisons of different ISPs may be discerned.

Depending on how it is organized, funded, and undertaken, the QoS information could be made widely publicly available. It could also respond to specific profiling requests, provide detailed profile analyses, and so on. Such a development would provide a public service and help build a much better understanding and overview for a much wider population of the Internet QoS performance over specific Internet paths nationally and internationally.

References

1. Coates, A., Hero III, A.O., Nowak, R., Yu, B.: Internet tomography. IEEE Signal Process. Mag. **19**(3), 47–65 (2002)

2. Lawrence, E., Michailidis, G., Nair, V.: Fast, moment-based estimation methods for delay tomography. Trans. Signal Process. (2008)
3. Castro, R., Coates, M., Liang, G., Nowak, R., Yu, B.: Network tomography: recent developments. Stat. Sci. **19**(3), 499–517 (2004)
4. Lawrence, E., Michailidis, G., Nair, V., Xi, B.: Network tomography: a review and recent developments. Front. Stat. (2006)
5. Tsang, Y., Coates, M.J., Nowak, R.: Network delay tomography. IEEE Trans. Signal Process. **51**(8), 2125–2136 (2003)
6. Vardi, Y.: Estimating source-destination traffic intensities from link data. J. Am. Stat. Assoc. **91**, 365–377 (1996)
7. Xi, B., Michailidis, G., Nair, V.N.: Estimating network loss rates using active tomography. J. Am. Stat. Assoc. **101**(476), 1430–1448 (2006)
8. Réseaux IP Européens (RIPE). https://atlas.ripe.net/landing/about/. Accessed 21 Sept 2020
9. Moloisane, A., Ganchev, I., O'Droma, M.: Internet Tomography: An Introduction to Concepts, Techniques. Tools and Applications. Cambridge Scholars Publishing, UK (2013)
10. Center for Applied Internet Data Analysis (CAIDA). https://www.caida.org/home/about/. Accessed 21 Sept 2020
11. Chen, Y., Bindel, D., Katz, R.H.: Tomography-based overlay network monitoring. In: Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement (2003)
12. Pásztor, A.: Accurate active measurement in the Internet and its applications. PhD thesis, The University of Melbourne (2003)
13. CAIDA Metrics Working Group: Network Measurement FAQ (2001)
14. Brodie, M., et al.: Active probing strategies for problem diagnosis in distributed systems (2003)
15. Bolot, J.-C.: Characterizing end-to-end packet delay and loss in the Internet. J. High-Speed Netw. **2**(3), 305–323 (1993)
16. Carter, R.L., Crovella, M.E.: Measuring bottleneck link speed in packet-switched networks. Perform. Eval. **27–28**, 297–318 (1996)
17. Ribeiro, V., Riedi, R., Coates, M., Baraniuk, R.G.: Multifractal cross-traffic estimation. In: ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management. Monterey, USA (2000)
18. Shakespeare, W.: In: Hibbard, G.R. (ed.) Hamlet (Act 1, Scene 4, Line 90). Oxford University Press, UK (2008)

Modelling a Folded N-Hypercube Topology for Migration in Fog Computing

Pedro Juan Roig, Salvador Alcaraz, Katja Gilly, and Carlos Juiz

Abstract IoT moving devices need to have their associated VMs as close as possible so as to minimize latency, jitter, bandwidth use and even power consumption. It implies that Data Centers in Fog Computing environments must be ready to move VMs around its different hosts on a discretionary manner in order to cope with such movements. In this paper, a Folded N-Hypercube switching infrastructure is being modelled according to different views, such as arithmetic, logical and algebraic, paying special attention as to how to manage VM migrations around the Fog domain.

Keywords Algebraic modelling · Fog computing · Folded N-hypercube · IoT · Live VM migration

1 Introduction

Virtual Machine (VM) migration is a key factor when dealing with Fog Computing environments [1], as VMs are located in a host within the Fog area and they need to be able to follow their associated users in case those move around the Fog domain so as to improve their key performance indicators [2].

The process of managing live VM migrations between a source host and a destination host within a Fog environment is a matter of debate in literature, although the pre-copy technique seems to prevail in current deployments due to its optimal trade-

P. J. Roig · S. Alcaraz · K. Gilly
Miguel Hernández University, Elche, Spain
e-mail: salcaraz@umh.es

K. Gilly
e-mail: katya@umh.es

P. J. Roig (✉) · C. Juiz
University of the Balearic Islands, Palma de Mallorca, Spain
e-mail: proig@umh.es

C. Juiz
e-mail: cjuiz@uib.es

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_42

off between downtime and migration time and the way its deals with dirty pages [3]. Basically, it forwards all RAM contents and CPU state from source to target VM.

In order to optimize the migration process, the selection of a Data Center topology is important as a balance is needed between some factors, such as minimizing the number of hops and design complexity. Furthermore, the number of users in a Fog environment is supposed to be lower than its counterpart in the Cloud. All those considerations make Folded N-Hypercube a topology that may well fit in these specifications. Therefore, the target in this paper is to propose models for the VM migration process in a Folded N-Hypercube topology from different points of view, such as arithmetic, logical and algebraic ways.

The organization of this paper is as follows: to start with, Sect. 2 presents the folded N-Hypercube topology, after that, Sect. 3 introduces the basics of the model, next, Sect. 4 brings up the models proposed for hosts, afterwards, Sect. 5 introduces the models proposed for nodes, then, Sect. 6 undertakes the model verification, and finally, Sect. 7 wraps up with some conclusions.

2 Folded N-Hypercube

The Folded N-Hypercube topology is an extension of the Plain N-Hypercube, and as such, it is also a graph-like structure where the nodes are distributed on the vertices of an N-Hypercube. N is the dimension of the structure and it drives the number of nodes and links among them, such that there has to be 2^N nodes and $(N+1) \cdot 2^{N-1}$ links among those nodes, following a specific layout. Additionally, each node has $N+1$ links to other nodes.

The key difference between plain N-Hypercube and Folded N-Hypercube is the introduction of extra edges linking all opposite nodes [4]. This extension helps decrease a great deal the distances between any two given nodes, as the new links bring all distances between any two opposite nodes to just 1 hop away, thus reducing the way to get to the further nodes. This fact reduces the maximum distance between nodes to $\lceil \frac{N}{2} \rceil$, which accounts for the successor integer out of $N/2$, in case it is not integer. Moreover, the introduction of such links help increase the number of redundant paths within the topology [5].

In order to find out which is the opposite node of a given node i, it may be done by either using binary or decimal arithmetics. The former needs to get i in binary notation, and then, calculate its $1's$ complement, by swapping all 0 into 1, and subsequently, all 1 to 0. The latter needs to have i in decimal notation, and then, calculate the following expression: $2^N - 1 - i$. Figure 1 shows how to transform a plain N-Hypercube into a Folded N-Hypercube, just by linking all opposite nodes.

Regarding the distances between nodes, it is obvious that a given node i only has one node at 0 hops away, which is itself. The rest of the distances will be in the range between 0 and $\lceil \frac{N}{2} \rceil$, where the amount of nodes that are at a particular distance within that range may be obtained through different procedures.

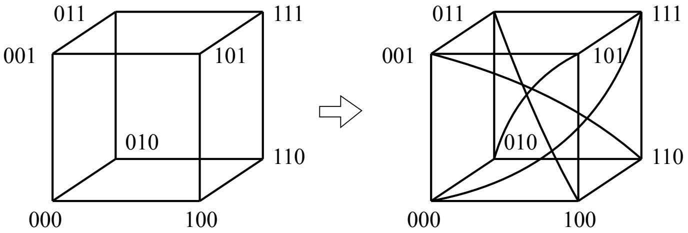


Fig. 1 Converting a plain N-Hypercube into a folded N-Hypercube

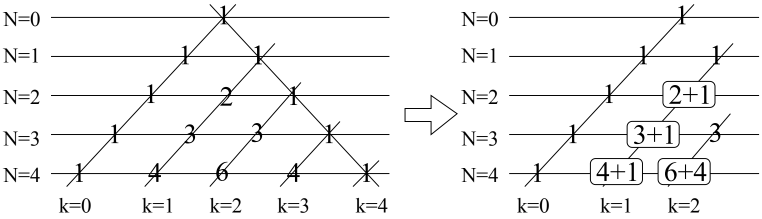


Fig. 2 Converting a Pascal's triangle into a folded Pascal's triangle

The first method is by looking at the N-th row and the k-th column of the Folded Pascal's triangle [6]. This type of triangle is a variation from the well known Pascal's triangle, such that the leftmost side of the triangle is left as it is, which is column $k = 0$ and is an all 1's column, whilst the rightmost side of the triangle is bent over the left, in a way that the last column ($k = N$) of each row gets over the second column ($k = 1$).

This way, it happens that each column from $k = 1$ onwards to $\lfloor \frac{N}{2} \rfloor$ happens to accumulate two values in it, and both those values get added up. It is to be reminded that $\lfloor \frac{N}{2} \rfloor$ accounts for the predecessor integer out of $N/2$, in case it is not integer. Therefore, values at column $k = 1$ and $k = N$ are added up, so are values at column $k = 2$ and $k = N - 1$, and so on. However, it has to be taken into account that when N has an odd value, the column $\lceil \frac{N}{2} \rceil$ is left alone after having done the folding, so such a value is left as it is, whereas if N is even, $\lfloor \frac{N}{2} \rfloor = \lceil \frac{N}{2} \rceil = \frac{N}{2}$ and this issue does not apply. Figure 2 shows the folding process of the Pascal's triangle.

In particular, and focusing on the row N, each k-th column gives the number of nodes being at a distance k from the given node i, where those columns start from 0 and goes onwards in a sequential manner from left to right, as shown in Fig. 3.

It is easy to see that the Folded Pascal's triangle does not meet the main feature of Pascal's triangle, this being that a value located in a certain position (that being row N and column k) is the sum of both its upper values (those being the one located at row $N - 1$ and column $k - 1$ and the one located at row $N - 1$ and column k). In the same sense, other known characteristics of the original triangle are not maintained in its folded version, such as column $k = 1$ containing the list of natural numbers $\{t_n = N\}$, column $k = 2$ bringing the list of triangular numbers $\{t_n = N \cdot (N - 1)/2\}$, column

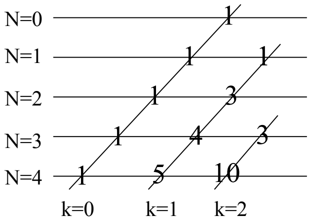


Fig. 3 Folded Pascal's triangle for the first rows

$k = 3$ bearing the list of tetrahedral numbers $\{t_n = N \cdot (N+1) \cdot (N+2)/6\}$ or the diagonal sums holding the list of Fibonacci numbers $\{t_n = t_{n-1} + t_{n-2}\}$. Nonetheless, it perfectly fits the task of counting how many nodes are at a distance k from a given node i in a Folded N-Hypercube topology, and that is why it has been quoted herein.

The second method to achieve that task is to use a expression involving the work with binomial coefficients, where N and k keep the same meaning as above. Some consideration needs to be made for this job. On the one hand, the mathematical definition of factorial, which is $n! = n \cdot (n-1)!$, yields to $(n-1)! = n!/n$. Departing from that expression, it is easily deduced that $0! = 1!/1 = 1$, and additionally, $(-1)! = 0!/0 =$, whose result is an indeterminate form in $\mathbb{R}$, which may not be considered in case the Gamma function is not applied, therefore, the combinations where this fact appears may well be skipped.

On the other hand, regarding the particular issue in rows where N is odd, namely, the column $\lceil \frac{N}{2} \rceil$ is left alone after the action of triangle bending, the mathematical expression to fill in the Folded Pascal's triangle needs to take that into account. The easiest way to do so is to apply a corrective coefficient which only is active in such cases. It may be implemented in a couple of ways, such as dividing each term by $\lfloor \frac{i}{N-i+1} + 1 \rfloor$, or otherwise, multiplying it by $2^{-\lfloor \frac{i}{N-i+1} \rfloor}$. Either way works, but the second option is going to be selected herein in expression (1).

$$W_{k=0}^{\lceil \frac{N}{2} \rceil}\left(\binom{N}{i} + \binom{N}{N-i+1}\right) \cdot 2^{-\lfloor \frac{i}{N-i+1} \rfloor} \qquad (1)$$

Here they are the first values attained with the expression presented, which gives the same values as with the Folded Pascal's triangle.

- $N = 2 \rightarrow w_k = \left(\binom{2}{0} + \binom{2}{3}\right) \cdot 1, \left(\binom{2}{1} + \binom{2}{2}\right) \cdot 1 \rightarrow w_k = 1, 3$
- $N = 3 \rightarrow w_k = \left(\binom{3}{0} + \binom{3}{4}\right) \cdot 1, \left(\binom{3}{1} + \binom{3}{3}\right) \cdot 1, \left(\binom{3}{2} + \binom{3}{2}\right) \cdot 1/2 \rightarrow w_k = 1, 4, 3$
- $N = 4 \rightarrow w_k = \left(\binom{4}{0} + \binom{4}{5}\right) \cdot 1, \left(\binom{4}{1} + \binom{4}{4}\right) \cdot 1, \left(\binom{4}{2} + \binom{4}{3}\right) \cdot 1 \rightarrow w_k = 1, 5, 10$
- $N = 5 \rightarrow w_k = \left(\binom{5}{0} + \binom{5}{6}\right) \cdot 1, \left(\binom{5}{1} + \binom{5}{5}\right) \cdot 1, \left(\binom{5}{2} + \binom{5}{4}\right) \cdot 1, \left(\binom{5}{3} + \binom{5}{3}\right) \cdot 1/2 \rightarrow w_k = 1, 6, 15, 10$

- $N = 6 \rightarrow w_k = \left(\binom{6}{0} + \binom{6}{7}\right) \cdot 1, \left(\binom{6}{1} + \binom{6}{6}\right) \cdot 1, \left(\binom{6}{2} + \binom{6}{5}\right) \cdot 1, \left(\binom{6}{3} + \binom{6}{4}\right) \cdot 1 \rightarrow w_k = 1, 7, 21, 35.$

The third method to obtain the number of nodes at a certain distance k from a given node i is the use of expression (2), which depends on the parity of N [7]:

$$W_k = \begin{cases} \text{for } 0 \leq k < \lceil \frac{N}{2} \rceil, & \binom{N+1}{k} \\ \text{for } k = \lceil \frac{N}{2} \rceil, & \begin{cases} \text{for even } N, & \binom{N+1}{N/2+1} \\ \text{for odd } N, & \binom{N}{\lceil \frac{N}{2} \rceil} \end{cases} \end{cases} \tag{2}$$

Here they are the first values achieved with the expression shown, which attains the same values as in the previous methods.

- $N = 2 \rightarrow w_k = \binom{3}{0}, \binom{3}{2} \rightarrow w_k = 1, 3$
- $N = 3 \rightarrow w_k = \binom{4}{0}, \binom{4}{1}, \binom{3}{2} \rightarrow w_k = 1, 4, 3$
- $N = 4 \rightarrow w_k = \binom{5}{0}, \binom{5}{1}, \binom{5}{3} \rightarrow w_k = 1, 5, 10$
- $N = 5 \rightarrow w_k = \binom{6}{0}, \binom{6}{1}, \binom{6}{2}, \binom{5}{3} \rightarrow w_k = 1, 6, 15, 10$
- $*$ N=6 $\rightarrow w_k = \binom{7}{0}, \binom{7}{1}, \binom{7}{2}, \binom{7}{4} \rightarrow w_k = 1, 7, 21, 35.$

In summary, all three methods presented have rendered the same results. Anyway, the series of results achieved may be interpreted as follows: if $N = 2$, a given node i has 1 node at 0 hops away (that's itself) and 3 nodes at 1 hops away (all its neighbors), whereas if $N = 3$, i has 1 node at 0 hops away (itself), 4 nodes at 1 hop away (its neighbors), and 3 nodes at 2 hops away, and the same reasoning applies to the any series related to any certain value of N.

Additionally, another compelling point is to attain the average number of hops away that a given node i has to take to get to the rest of the nodes within the topology. Furthermore, the average number of hops away between any two hosts may be achieved as well, taking into account that every node is connected to the same quantity of hosts. In such a case, the previous result should be incremented by 2 so as to consider the links connecting source and destination hosts to their respective nodes [8].

For both average distances, a couple of expressions are going to be given, where one is a full expression and the other one is a consolidated one, the former being longer than the latter, even though both reach the same results [9], as seen in Table 1. Specifically, as per the average distance among nodes, a summatory is performed with the quantity of nodes being at a certain amount of hops getting multiplied by the number of such hopes, and then, that quantity gets divided by the overall number of nodes. On the other hand, as per the average distance among hosts, the procedure is the same, with the particularity that the number of hops get increased by 2, so as to consider the links connecting source and destination hosts to the infrastructure.

Table 1 Average distance among switches and among hosts

	Full expression	Consolidated expression
Average hops away between any two **nodes**	$\frac{\sum_{i=0}^{\lceil \frac{N}{2} \rceil} \left(\binom{N}{i} + \binom{N}{N-i+1} \right) \cdot 2^{-int(\frac{i}{N-i+1})} \cdot i}{2^N}$	$\frac{\frac{N+1}{2} \cdot \left(2^N - \binom{N}{\lfloor \frac{N}{2} \rfloor} \right)}{2^N}$
Average hops away between any two **hosts**	$\frac{\sum_{i=0}^{\lceil \frac{N}{2} \rceil} \left(\binom{N}{i} + \binom{N}{N-i+1} \right) \cdot 2^{-int(\frac{i}{N-i+1})} \cdot (i+2)}{2^N}$	$\frac{\frac{N+1}{2} \cdot \left(2^N - \binom{N}{\lfloor \frac{N}{2} \rfloor} \right) + 2^{N+1}}{2^N}$

3 Model

Nodes may be identified with N binary digits, where each bit represents one of the dimensions, which facilitates the understanding of what node is being referenced. An initial node is selected to have all bits set to 0, and from there on out, a 0 bit value means that there is no movement in such a dimension related to the initial node, whereas a 1 bit value means quite the opposite. This may be seen in Fig. 4.

Nodes may be identified with decimal digits as well, only by converting each binary digit into its decimal counterpart, making the identifiers easier to work with, as exhibited in Fig. 5.

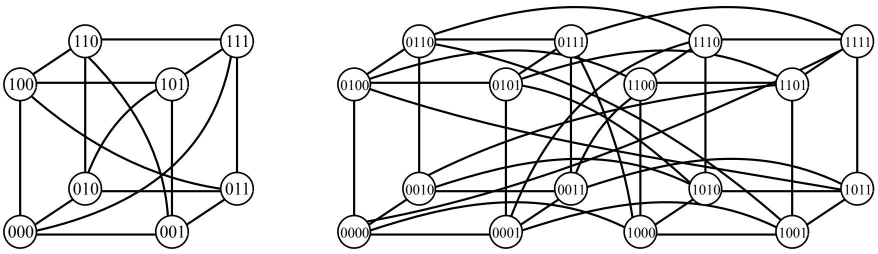


Fig. 4 Node nomenclature using N binary values: $N = 3$ (left); $N = 4$ (right)

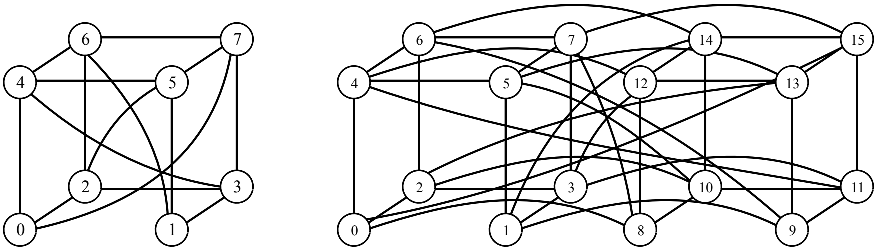


Fig. 5 Node nomenclature using decimal values: $N = 3$ (left); $N = 4$ (right)

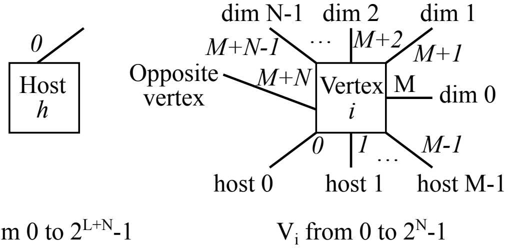


Fig. 6 Links on each host H_h and each vertex V_i of the N-Hypercube architecture

Additionally, the model proposed contains hosts h and switches i, where the latter are located at each node of the N-Hypercube shape, and the hosts are hanging on each of those switches. Hence, there are 2^N switches, standing at each of the vertices within the N-Hypercube shape, and moreover, there are M hosts hanging on each switch, going from $h_{i \cdot M}$ to $h_{(i+1) \cdot M-1}$, all of them with a single port 0. Furthermore, in each switch, the first M ports (ranging from 0 to $M-1$) are dedicated to connect to its corresponding hosts, whilst the next N ports are dedicated to connect to its neighboring nodes (ranging from M to $M+N-1$), and on top of that, there is an extra port connecting to its opposite node (identified by $M+N$), as depicted in Fig. 6. Furthermore, it is to be noted that M is a power of 2, meaning that $M = 2^L$, where L is the natural number resulting of the logarithmic expression $log_2 M$. Anyway, 2^L will get referenced in expressions, whilst M will do it in port numbers.

4 Host

4.1 Preamble

There are 2^N nodes, each one with $M = 2^L$ hosts, accounting for 2^{L+N} hosts overall. The point is that if a VM associated to a user VM(u) is not held within a host h, then that host gets ready to receive it at any time, whereas if it is held, then it gets ready to send it at any time, whereas such a VM keeps working therein.

Three types of modellings are going to be presented for both hosts and nodes, called arithmetic, logical and algebraic, according to the mathematical tools employed. The first and the second are going to be described by means of coding, specifically with a C-style coding, as it gets better readability than other options.

4.2 Coding Model

Focusing on host models, both arithmetic and logical ones match, with a and b being source host and destination host, and h is the current host, as may be seen in

Algorithm 1. Furthermore, *send* and *receive* functions take 3 arguments, such as the initial host, the final host and the associated VM. Moreover, they also take 2 parameters such as the device involved and its corresponding port.

```
for (h = 0; h < $2^{L+N}$; h++) {
 while (1) {
  for (u = 0; u < TotalUsers; u++) {
   if NOT (VM(u) in h)
    receive$_{HOST\{h\},PORT\{0\}}$(a,h,VM(u));
   else
    send$_{HOST\{h\},PORT\{0\}}$(h,b,VM(u));
}}}
```

Algorithm 1 HOST().

4.3 ACP Model

Algebraic models are presented by means of Algebra of Communicating Processes (ACP), whose recursive expressions are formed by process terms composed by atomic actions such as send and read a generic message d, where d stands for the associated VM. Hence, those actions are represented by $s(d)$ and $r(d)$, whose parameters are the device involved, being H_h, and its corresponding port, being *0*, as seen in (3). Besides, some basic operators are used, such as sequential and alternate, represented by the multiplication and the addition sign, respectively [10].

$$H_h = \sum_{h=0}^{2^{L+N}-1} \Big(s_{H_h,0}\big(d\big) + r_{H_h,0}\big(d\big) \Big) \cdot H_h \tag{3}$$

5 Switch

5.1 Preamble

There are 2^N nodes, and if any of them receives a VM in any of their ports, the first thing is to check if that node i holds the destination host b, hence, i is the destination node, thus, $i = \left\lfloor \frac{b}{2^L} \right\rfloor$. If that is the case, then that VM gets forwarded to that host. However, if that is not the case, a lookup for mismatching dimensions are carried out between the given node and the destination node, along with another lookup for the matching dimensions, adding up the extra opposite node. Whichever path is shorter, the VM will get forwarded over all its corresponding ports. In case both paths are equally long, then the VM is sent over all available paths. Hence, 2 rounds are needed: the first one to get the shortest path and the second one to send the VM.

The key point here is to check for every dimension *j* if the values of the present node *i* and the destination node $\lfloor \frac{b}{2^L} \rfloor$ match or otherwise. This may be done by using the decimal or the binary identifier of the node. The former is used by the arithmetic and algebraic models, by means of comparing $\lfloor \frac{i}{2^j} \rfloor_{|2}$ and $\left\lfloor \frac{\lfloor b/2^L \rfloor}{2^j} \right\rfloor_{|2}$, whereas the latter is used by the logical model, by means of the logical XOR operator in all bits.

Furthermore, in both the arithmetic and logical models, two designs are presented whose main difference is the use of arrays to keep the results obtained in the first round so as to apply them in the second round. Hence, the first design shows the full algorithm and the rest of them just shows the *else* block, as the rest is the same.

5.2 Arithmetic Model

Option A: Algorithm 2 is shown, where mismatched dimensions may be found out by means of *arithmetic* operations only, with *no array* involved.

```
   for (i = 0; i < $2^N$; i++) {
   while (1) {
     for (p = 0; p < M + N + 1; p++) {
      if (receive$_{NODE\{i\},PORT\{p\}}$(a,h,VM(u))) {
     if (i == int (b / $2^L$))
      send$_{NODE\{i\},PORT\{b \bmod M\}}$(a,h,VM(u));
     else {
      distance = 0;
        for (j = 0; j < N; j++)
         if (((int (i / $2^j$)) mod 2) != ((int (int (b / $2^L$) / $2^j$))
  mod 2))
          distance = distance + 1;
      if (distance <= (N + 1 - distance)) {
         for (j = 0; j < N; j++)
          if (((int (i / $2^j$)) mod 2) != ((int (int (b / $2^L$) / $2^j$))
  mod 2))
           send$_{NODE\{i\},PORT\{M+j\}}$(a,h,VM(u));
      }
      if (distance >= (N + 1 - distance)) {
         for (j = 0; j < N; j++)
          if (((int (i / $2^j$)) mod 2) == ((int (int (b / $2^L$) / $2^j$))
  mod 2))
           send$_{NODE\{i\},PORT\{M+j\}}$(a,h,VM(u));
         send$_{NODE\{i\},PORT\{M+N\}}$(a,h,VM(u));
   }}}}}}
```

Algorithm 2 VERTEX_a().

Option B: Algorithm 3 is shown, where mismatched dimensions may be found out by means of *arithmetic* operations, with the help of *one array*.

```
       else {
        distance = 0;
        for (j = 0; j < N; j++) {
         if (((int (i / $2^{j}$)) mod 2) != ((int (int (b / $2^{L}$) / $2^{j}$))
  mod 2)){
          index[distance] = j;
          distance = distance + 1;
        }}
        if (distance <= (N + 1 - distance)) {
         x = 0;
         for (y = 0; y < N; y++) {
          if (index[x] == y) {
           $\text{send}_{\text{NODE\{i\},PORT\{M+y\}}}$(a,h,VM(u));
           x = x + 1;
        }}}
        if (distance >= (N + 1 - distance)) {
         x = 0;
         for (y = 0; y < N; y++) {
          if (index[x] != y)
           $\text{send}_{\text{NODE\{i\},PORT\{M+y\}}}$(a,h,VM(u));
          else
           x = x + 1;
         }$\text{send}_{\text{NODE\{i\},PORT\{M+N\}}}$(a,h,VM(u));
       }}
```

Algorithm 3 VERTEX_b(), just the *else* block, where the difference lays.

5.3 Logical Model

Option C: Algorithm 4 is shown, where mismatched dimensions may be found out by means of *logical* operations, with the help of *two arrays*.

```
    else {
     distance1 = 0;
     iNodeID = DecimalToBinary(i);
     dNodeID = DecimalToBinary(int(b / 2^L));
     logicalAND1 = iNodeID XOR dNodeID;
     distance2 = 0;
     oNodeID = DecimalToBinary(2^N - 1 - i);
     logicalAND2 = iNodeID XOR oNodeID;
     for (j = 0; j < N; j++) {
      if (logicalAND1[j] == 0) {
       index1[distance1] = j;
       distance1 = distance1 + 1;
      }
      if (logicalAND2[j] == 0) {
       index2[distance2] = j;
       distance2 = distance2 + 1;
     }}
     index2[distance2] = N;
     distance2 = distance2 + 1;
     if (distance1 <= distance2)
      for (x = 0; x < distance1; x++)
       send_{NODE{i},PORT{M+index1[x]}}(a,h,VM(u));
     if (distance1 >= distance2)
      for (x = 0; x < distance2; x++)
       send_{NODE{i},PORT{M+index2[x]}}(a,h,VM(u));
}}}}}
```

Algorithm 4 VERTEX_c(), just the *else* block, where the difference lays.

Option D: Algorithm 5 is shown, where mismatched dimensions may be found out by means of *logical* operations, with the help of only *one array*.

```
 else {
  distance = 0;
  iNodeID = DecimalToBinary(i);
  dNodeID = DecimalToBinary(int(b / $2^{L}$));
  logicalAND = iNodeID XOR dNodeID;
  for (j = 0; j < N; j++) {
   if (logicalAND[j] == 0) {
    index[distance] = j;
    distance = distance + 1;
  }}
  if (distance <= (N + 1 - distance)) {
   x = 0;
   for (y = 0; y < N; y++)
    if (index[x] == y) {
     send$_{NODE\{i\},PORT\{M+y\}}$(a,h,VM(u));
     x = x + 1;
  }}
  if (distance >= (N + 1 - distance)) {
   x = 0;
   for (y = 0; y < N; y++) {
    if (index[x] != y)
     send$_{NODE\{i\},PORT\{M+y\}}$(a,h,VM(u));
    else
     x = x + 1;
   }send$_{NODE\{i\},PORT\{M+N\}}$(a,h,VM(u));
}}
```

Algorithm 5 VERTEX_d(), just the *else* block, where the difference lays.

5.4 ACP Model

The ACP model proposed for a switch is an expression based on searching mismatched dimensions by means of *arithmetic* operations, with two additional functions to count the distance in number of nodes within the shortest path to go from a source node to a destination node. One of them is called neighborsPath$_i$, shown in (4), as it considers the shortest path through only neighboring nodes, whereas the latter is named oppositePath$_i$, shown in (5), as it does the same by using an opposite node link at some point. It is to be reminded that i identifies a particular node and variable *dist* keeps the count of nodes. Regarding variable *dist*, it is first initialized, and then, it is incremented each time the corresponding path passes through a new node. This fact is represented by a mismatched dimension in the neighboring paths, whereas for the matched dimensions, the count is not incremented, which is represented by an empty set. On the other hand, the opposite paths work the other way around, this is, counting up the matched dimensions and discarding the mismatched ones, although in that case it has to be added up another unit to take note of the direct link to the opposite node.

Regarding the syntax of the expression V_i, shown in (6), it involves all the nodes, identified by i, and all the ports within each node, identified by p, no matter if those ports are connecting to other nodes, or otherwise, connecting to hosts. Hence, if a message is received in any port of a node, if that node is the destination one, then the message will be sent over the port connecting to the destination host, identified by $b_{|M}$, meaning the operation b *modulo* M. In that is not the case, then 3 things may happen. The first one is that all mismatched dimensions are sought after in case the neighboring path$_i$ is shorter than opposite path$_i$, and the message is sent through the ports corresponding to such dimensions. The second one is that all matched dimensions are sought after in case the opposite path$_i$ is shorter than the neighboring path$_i$, and the message is forwarded on across the ports corresponding to those dimensions. The third one is that both paths have the same length, so the message is forwarded through all ports connecting to other nodes.

$$\text{neighborsPath} = \sim\text{dist} \cdot \sum_{j=0}^{N-1} \left((\text{dist} + +) \triangleleft \left\lfloor \frac{i}{2^j} \right\rfloor_{|2} \neq \left\lfloor \frac{\lfloor b/2^L \rfloor}{2^j} \right\rfloor_{|2} \triangleright \emptyset \right) \tag{4}$$

$$\text{oppositePath} = \sim\text{dist} \cdot \sum_{j=0}^{N-1} \left((\text{dist} + +) \triangleleft \left\lfloor \frac{i}{2^j} \right\rfloor_{|2} = \left\lfloor \frac{\lfloor b/2^L \rfloor}{2^j} \right\rfloor_{|2} \triangleright \emptyset \right) \cdot (\text{dist} + +) \tag{5}$$

$$
\begin{aligned}
V_i = \sum_{i=0}^{2^N-1} \Bigg(\sum_{p=0}^{M+N} \Bigg(r_{V_i,p}(d) \cdot \Bigg(s_{V_i,b_{|M}}(d) \lhd i = \left\lfloor \frac{b}{2^L} \right\rfloor \rhd \\
\Bigg(\Bigg(\Bigg(\sum_{j=0}^{N-1} s_{V_i,M+j}(d) \lhd \left\lfloor \frac{i}{2^j} \right\rfloor_{|2} \neq \left\lfloor \frac{\lfloor b/2^L \rfloor}{2^j} \right\rfloor_{|2} \rhd \emptyset \Bigg) \\
\lhd \text{ neighborsPath} <= \text{oppositePath} \rhd \emptyset \Bigg) \\
\cdot \Bigg(\Bigg(\Bigg(\sum_{j=0}^{N-1} s_{V_i,M+j}(d) \lhd \left\lfloor \frac{i}{2^j} \right\rfloor_{|2} = \left\lfloor \frac{\lfloor b/2^L \rfloor}{2^j} \right\rfloor_{|2} \rhd \emptyset \Bigg) + s_{V_i,b_{|M+N}}(d) \Bigg) \\
\lhd \text{ neighborsPath} >= \text{oppositePath} \rhd \emptyset \Bigg) \Bigg) \Bigg) \Bigg) \Bigg) \cdot V_i
\end{aligned}
\tag{6}
$$

5.5 Redundant Paths

Algorithm 6 attains the list of movements available to move from a source node, where the source host is linked, to a destination node, where the destination host is linked. If both nodes match, this expression applies: $\left\lfloor \frac{a}{2^L} \right\rfloor = \left\lfloor \frac{b}{2^L} \right\rfloor$, whereas if they do not, the shortest path may be calculated between the mismatching and matching dimensions j, by comparing $\left\lfloor \frac{\lfloor a/2^L \rfloor}{2^j} \right\rfloor_{|2}$ and $\left\lfloor \frac{\lfloor b/2^L \rfloor}{2^j} \right\rfloor_{|2}$. Eventually, the movements induced by the shortest path to get from source to destination are found.

```
Paths(a,b) {
 items = [ $H_a$ $V_{int(a/2^L)}$ ];
 t = 0;
 if (int (a / $2^L$) == int (b / $2^L$)) {
  items += [ Hb ];
  topology = "INTRASWITCH";
 }
 else {
  distance = 0;
  for (j = 0; j < N; j++)
  if (((int (int (a / $2^L$) / $2^j$)) mod 2) != ((int (int (b / $2^L$) / $2^j$)) mod 2))
    distance = distance + 1;
  items += [ " ( PERMUTATIONS OF THESE MOVEMENTS: " ];
  if (distance <= (N + 1 - distance)) {
   for (j = 0; j < N; j++) {
    if (((int (int (a / $2^L$) / $2^j$)) mod 2) != ((int (int (b / $2^L$) / $2^j$)) mod 2)) {
     dimension[t] = j;
     exponent = ((int (int (a / $2^L$) / $2^j$)) mod 2);
     direction[t] = $(-1)^{exponent}$;
     movement[t] = direction[t].$2^{dimension[t]}$;
     items += [ movement[t] + "," ];
     t = t + 1;
  }}}
  if (distance >= (N + 1 - distance)) {
   for (j = 0; j < N; j++) {
    if (((int (int (a / $2^L$) / $2^j$)) mod 2) == ((int (int (b / $2^L$) / $2^j$)) mod 2)) {
     dimension[t] = j;
     exponent = ((int (int (a / $2^L$) / $2^j$)) mod 2);
     direction[t] = $(-1)^{exponent}$;
     movement[t] = direction[t].$2^{dimension[t]}$;
     items += [ movement[t] + "," ];
     t = t + 1;
   }}
   exponent = int((int (a / $2^L$)) / (N/2));
   moveOpposite = $(-1)^{exponent}$.($2^N$ - 1 - a);
   movement[t] = moveOpposite;
   items += [ movement[t] ];
   t = t + 1;
  }
  items += [ " ) " + $V_{int(b/2^L)}$ $H_b$ ];
  topology = "INTERSWITCH";
}}
```

Algorithm 6 Paths(a,b).

6 Verification

The Folded N-Hypercube represents a bounded N-dimensional geometric shape, and as such, all links are kept within such a shape. Additionally, each node has N links going to its neighboring nodes, as well as an extra link going to its opposite node. Therefore, all links are bounded into the structure and there are redundant paths to get from any two nodes whatsoever. Hence, by applying the fairness condition, a message departing from a given node will always reach another given node, no matter how much time it takes or how many links it goes through, thus no message will ever get lost forever.

Additionally, ACP is an abstract algebra, hence proof by contradiction is one of the verification methods available. Thus, taking into account that the Folded N-Hypercube is a closed graph and applying the fairness condition, it is clear that a given source host a will reach a given destination host b through a Folded N-Hypercube topology if enough time is granted to make as many moves as necessary to get there. Therefore, it is no possible that a message departing from a never reaches b, hence, the model gets verified.

Furthermore, if the external behaviour of the ACP model and that of the real system share the same string of actions and have the same branching structure, then it may be said that both are rooted branching bisimilar, that being a sufficient condition to get a model verified.

- External behavior of the model, shown in (7):

$$\tau_I\left(\partial_H\left(||_{i=0}^{2^N-1} V_i\right)\right) = r_{H_h,0}(d) \cdot s_{H_h,0}(d) \cdot \tau_I\left(\partial_H\left(||_{i=0}^{2^N-1} V_i\right)\right) \tag{7}$$

- External behavior of the real system, shown in (8): A message is sent from the source host to a node in the Folded N-Hypercube infrastructure, and in turn, is received in the destination host from a node hanging out of that topology:

$$X = r_{H_h,0}(d) \cdot s_{H_h,0}(d) \cdot X \tag{8}$$

Therefore, it is obvious that this pair of expressions are both recursive and both are being multiplied by the same factors. Thus, both expressions are rooted branching bisimilar, as shown in (9), because they share the same string of actions and they also have the same branching structure.

$$X \longleftrightarrow \tau_I\left(\partial_H\left(||_{i=0}^{2^N-1} V_i\right)\right) \tag{9}$$

This a sufficient condition for a model to get verified, hence, the model proposed is verified.

7 Conclusions

In this paper, a study has been carried out regarding the modelling of VM migration between hosts being connected to a switching topology with the shape of a Folded N-Hypercube. This work has started with the introduction of the key features of such a geometric shape, followed by the presentation of the basic features of the model proposed for both hosts and switches.

Three different types of models have been shown, such as arithmetic, logical and algebraic ones. Furthermore, an algorithm focused on searching redundant paths has also been exposed. Eventually, a model verification has been undertaken to verify the model proposed.

References

1. Habibi, P., et al.: Fog Computing: a comprehensive architectural survey. IEEE Access **8**, 69105–69133 (2020). https://doi.org/10.1109/ACCESS.2020.2983253
2. Waqas, M., et al.: Mobility-aware Fog Computing in dynamic environments: understandings and implementation. IEEE Access **7**, 38867–38879 (2018). https://doi.org/10.1109/ACCESS.2018.2883662
3. Osanaiye, O., Chen, S., Yan, Z., Lu, R., Choo, K.R., Dlodlo, M.: From cloud to Fog Computing: a review and a conceptual live VM migration framework. In: Pecht, M. (ed.) IEEE Access, vol. 5, pp. 8284–8300 (2017). https://doi.org/10.1109/ACCESS.2017.2692960
4. El-Amawy, A., Latifi, S.: Properties and performance of folded hypercubes. IEEE Trans. Parallel Distrib. Algorithms **2**(1), 31–42 (1991). https://doi.org/10.1109/71.80187
5. Zhang, M.M., Zhou, J.X.: On g-extra connectivity of folded hypercubes. Theor. Comput. Sci. **593**, 146–153 (2015). https://doi.org/10.1016/j.tcs.2015.06.008
6. Roig, P.J., Alcaraz, S., Gilly, K.: Arithmetic study on energy saving for some common data centre topologies. In: Proceedings of the 8th European Conference ECRES 2020, Istanbul (Turkey), pp. 1–7 (accepted by not yet published)
7. Khan, Z.A., Siddiqui, J., Samad, A.: Topological evaluation of variants hypercube network. Asia J. Comput. Sci. Inf. Technol. **3**(9), 125–128 (2013). http://innovativejournal.in/index.php/ajcsit/article/view/15
8. Qiao, Y., Yang, W.: Edge disjoint paths in hypercubes and folded hypercubes with conditional faults. Appl. Math. Comput. **294**(C), 96–101 (2017). https://doi.org/10.1016/j.amc.2016.09.002
9. Disanto, F., Frosini, A., Rinaldi, S.: Square involutions. J. Integer Seq. **14**, Article 11.3.5, 1–15 (2011). https://cs.uwaterloo.ca/journals/JIS/VOL14/Rinaldi/square.html
10. Fokkink, W.: Introduction to Process Algebra. Springer, Heidelberg (2000). https://doi.org/10.1007/978-3-662-04293-9

Brain Electric Microstate of Karawitan Musicians' Brain in Traditional Music Perception

Indra K. Wardani, Djohan, Fortunata Tyasrinestu, and Phakkharawat Sittiprapaporn

Abstract The rapid development of music research leads to many interdisciplinary topics. In the neuroscience field of study, music is being studied related to either its effect on the cognitive process or the cognitive process behind it. Generally, neuroscience is a field of study to focus on the music's integration. The difference in brain structure and activities between non-musician and musicians has been shown by several previous studies. Instead of differentiating brain activity between musician and non-musician, the present study demonstrated the different brain activity while musicians listened to music regarding their musical experience. Applying the electroencephalography recording in the experimental approach toward Karawitan musicians, the results showed higher brain activity in listening to familiar music, *Gendhing Lancaran*, traditional music of Java. In addition, the dominant brain activity happened in the temporal lobe while Karawitan musicians listened to *Gendhing Lancaran*, the traditional music of Java.

Keywords Electroencephalography · Microstate · Brain · Karawitan · *Gendhing Lancaran*

1 Introduction

Neuroscience is known as a discipline focusing on cognition and had examined music and its cognition process for years. Neuroscience explored the human brain structure and its function as the effect of particular music (or music activity). It helps people

I. K. Wardani · Djohan · F. Tyasrinestu
Indonesia Institute of the Arts of Yogyakarta, Yogyakarta, Indonesia

P. Sittiprapaporn (✉)
Brain Science and Engineering Innovation Research Group, Mae Fah Luang University, Bangkok, Thailand
e-mail: wichian.sit@mfu.ac.th

Neuropsychological Research Laboratory, Department of Anti-Aging and Regenerative Science, School of Anti-Aging and Regenerative Medicine, Mae Fah Luang University, Bangkok, Thailand

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_43

to understand the human brain mechanism behind music perception. Many pieces of research had been done to understand the effect generated by particular music activity toward cognitive function. Listening to music passively as a background of certain activity has been proved to improve cognitive performance [1]. Meanwhile, certain treatments of music in terms of tempo and mode were able to stimulate arousal and mood to increase cognitive abilities in spatial tasks [2]. A previous study revealed how musical training is beneficial to alter brain function and enhance cognitive performance outside music [3]. Using the visuospatial task applied to orchestral musicians and non-musicians subject, this previous study tried to provide evidence on how complexity in musical training done by orchestral musicians can affect the activation of Broca's area during the test that enhanced their performance [3]. One previous study revealed how the structure of professional musicians, amateur musicians, and non-musicians was different from each other's. This previous study found that both amateur and professional musicians' brain structure looked similar because of a relative contribution of long-term musical practice [4].

Another previous study compared musicians and non-musicians where the musical experience became an important consideration as prior conditioning to differentiate subjects [1, 5]. The long-lasting musical training and certain skill obtaining experienced by both musicians and non-musicians empower them to have an alternate cognitive process. However, musicians would practice more complex rehearsal processes, i.e., physical and mental operation, in order to construe musical notation as a piece of visual information. Alter it as a motoric response, remember the musical phrase, and do musical improvisation, as well as identify a note without any referential note were included in the rehearsal processes [5]. One of several stimuli in which musicians used to train their function and enhance their performance as well as build up their recognition was known as music practice particularly its complexities. Additionally, those previous studies which employed musicians and non-musicians as the primary subject to distinguish musical experience in terms of "had received" and "had not received" musical teaching. It is quite clear to create such a distinction between the subject and to hypotheses the possible difference occurred.

Referring to one previous study of musical preference and cognitive style showed that a tendency of individuals who have certain cognitive style as a base of their personality traits and how musical genre became a distinctive variable. Instead of examined brain activity differences between musicians and non-musicians, the present study aimed to describe the brain activity of Karawitan musicians listening to traditional Javanese music, *Gendhing Lancaran.* The present study used a microstate segmentation technique and electroencephalography to measure the degree of cortical activation contributing to scalp recorded by auditory stimuli. Thus, the objective of the present study was to identify the current distribution of musical experiences by means of listening to *Gendhing Lancaran*, traditional music of Java.

2 Materials and Methods

2.1 Participants

Twenty healthy right-handed Karawitan musicians participated in this study. All participants were adults with normal hearing and no referred to neurological issues as self-reported. They were aged between 23 and 29 (mean 28.25 ± 1.41) and were employed to be engaged with this experiment. All Karawitan musicians employed in the present study actively learned practical music lessons of *Gendhing Lancaran* for consecutive three years. All participants had been explained and had approved the applied procedure. The approval of the Graduate School of Indonesia Institute of the Arts of Yogyakarta, Yogyakarta, Indonesia, on experiment and written consent from each subject were obtained.

2.2 Stimuli

Stimuli consisted of *Gendhing Lancaran*. The sounds were presented binaurally via earphones at 85 dB SPL. Participants were told to focus on the stimuli introduced through earphones.

2.3 Electroencephalographic Recording

For EEG recording, the standard 14 locations of the 10–20 system, EEG, Epoc Plus (EMOTIV, San Francisco, USA) (see Fig. 1: Left)., was recorded through wireless neuroheadset (EMOTIV Epoc) from 14 active electrodes (AF3, F3, F7, FC5, T7, P7, O1, O2, P8, T8, FC6, F8, F4, AF4) positioned by the 10–20 international system of electrode placement. The additional ground was applied. Reference electrodes were physically applied to the left and right mastoids (see Fig. 1: Right).

EEGs were acquired as continuous signals and amplified with a gain of 30,000. EEGs signal also filtered by a bandpass of 0.1–100 Hz. As epochs with voltage variation exceeding ± 100 μV at any EEG channel were dismissed from the further examination, and the EEG recordings were additionally filtered and carefully assessed for eye movement and muscle artifacts. In any case, all reactions were recalculated offline against average reference for additional investigation.

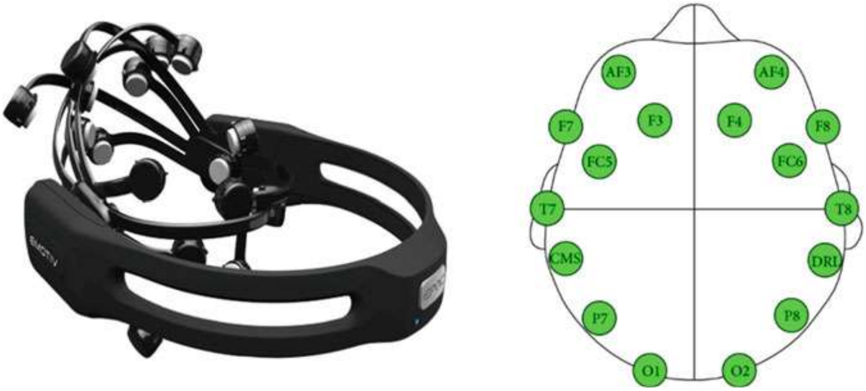


Fig. 1 Left: EEG device (EMOTIV Epoc) Right: 14-channel electrode montages: electroencephalographic recording from 14 channels were selectively analyzed in this experiment

2.4 Data Pre-processing and Feature Extraction

The testbench program along with EMOTIV devices was used to collect electroencephalographic data and generated it as edf. file before converted into.csv file for further analysis. These files were compatible with Microsoft Excel where we reduced unnecessary tables and make sure that collected data would be compatible with spatial analysis. The peak of global field power (GFP) was detected and used for analysis. The electric strength or hilliness of each brain electric field map was measured independently from its spatial configuration by GFP. The GFP peak measure was equivalent to the spatial standard deviation of all voltage values of one spontaneous EEG map. If a map was very hilly, the GFP peak measure would be higher than if the map was relatively flat. The GFP as defined above was independent of the reference [6, 7]. For each participant, mean GFP peak amplitudes over participants were computed, and all mean spontaneous EEG map series were carefully inspected for artifacts. For all mean spontaneous EEG map series, the locations of the centroid of each spontaneous EEG map series were computed according to Wackermann, et al. [8]. The points where the gravity of positive and negative areas of an average reference-referred map were named as centroids. For each of these centroid location points, the location coordinates were determined on the left-right axis and on the anterior-posterior axis. Thus, a single map was described by four coordinate values (see Fig. 2). All subsequent analysis steps (segmentation of the data into microstates and statistical analysis) were based on these extracted spatial descriptors of the maps.

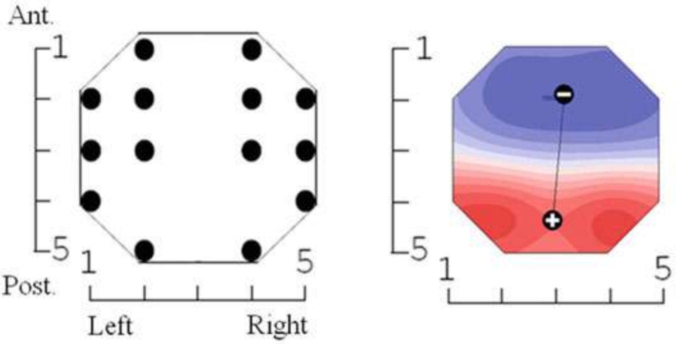


Fig. 2 Left: Array of the 14 recording electrodes (Crosses) within the 10/20 system. Electrode positions were defined by row number (anterior-posterior, vertical) and column number (left-right, horizontal) of the two-dimensional electrode matrix. Note that the most posterior electrode pair O1/O2 in reality had about the same lateral separation as the more anterior pairs. Right: Momentary potential distribution map displayed as equipotential line map (Equipotential lines in steps of 2.0 μV); positive areas red, negative areas blue. The locations of the centroids of the positive area (plus) and negative area (minus) were connected by a line. Head seen from above, left ear left

2.5 Assessment of Map Landscape

The spatial configuration of its potential distribution was further computed and assessed numerically according to the locations of the centroids of both positive and negative map areas [6]. This was called a landscape map. In this way, four parameters were obtained for each map; the location of the centroid of the positive map on the left-right axis and on the anterior-posterior axis; and the location of the negative map area centroid on the left-right axis and on the anterior-posterior axis. The use of centroids of the positive and negative areas of the potential map was used to describe spatial map configurations in the present study. Positive and negative centroids were the points of gravity of the average reference referred to as positive and negative map areas. In addition, the centroid locations described each landscape map by showing it as anterior-posterior and left-right coordinates of positive and negative centroid location. Other than in the case of the extremes, centroids were located anywhere within the mapped electrode array and were not restricted to the electrode positions [7, 8]. According to Fig. 2, the centroid locations appeared nearer to the center of the field. Moreover, they less occurred near the borders of the electrode array.

2.6 Statistical Analysis

While Karawitan musician listening to *Gendhing Lancaran,* the electric activities of the Karawitan musicians' brains were continuously recorded. Statistical analysis was performed on GFP peak amplitudes of 14 electrodes sites within the time range of EEG waveform (0–50,000 ms). Five sites were the prefrontal line, frontal line, central line, parietal line, and occipital line, respectively. GFP peak amplitudes were analyzed with ANOVAs measurement. The statistical significance of the electrical activities of each spontaneous EEG map was tested with a two-way sample *t*-test.

3 Results

Microstate segmentation of electrical activities in the Karawitan musicians' brain while listening to *Gendhing Lancaran* showed four map landscapes identifying as class A, B, C, and D, respectively. The two-way sample *t*-test comparing mean GFP peak amplitudes of spontaneous EEG map series across participants was statistically significant (0.85 ± 0.36 μV; $t(19) = 10.67$, $p < 0.0001$) (see Fig. 3).

Class D had highest electrical activity (1.13 ± 0.23 μV, $t(1) = 7.06$; $p = 0.089$) compared to other classes (e.g., class A: 0.99 ± 0.27 μV, $t(2) = 6.48$; $p = 0.02$, class B: 0.84 ± 0.33 μV, $t(4) = 5.66$; $p = 0.004$, and class C: 0.75 ± 0.40 μV, $t(9) = 5.92$; $p = 0.0002$), respectively. A repeated measures ANOVA conducted on the mean GFP peak amplitudes of each spontaneous EEG map series did not yield main effects of group ($F_{3,16} = 0.81$, $p = 0.50$). Table 1 shows mean GFP peak amplitudes of each spontaneous EEG map series elicited in the study. (see Table 1 and Fig. 4).

4 Discussion

The present finding shows that as Karawitan musician listened to familiar music, the dominant brainwave occurs at the temporal lobe. The temporal lobe was the region of the brain responsible for a memory processing task. It could indicate that memory processing occurred when participants were listening to familiar music. In the present study, Karawitan musician listened to *Gendhing Lancaran.* In terms of brain activity, Karawitan musician achieved higher brain activity when listening to *Gendhing Lancaran.* This might be indicated that a higher brain activity occurred when the participants were more familiar with the culturally particular music, they listened to it. The previous study [9] showed the importance of familiarity instead of liking in increasing brain activity of individuals. Using fMRI toward the participant, this previous study showed certain emotion-related areas activation during familiar music listening such as the amygdala, putamen, anterior cingulate cortex, and thalamus. Meanwhile, liking only to contribute a marginal effect on brain activation [9]. In the present study, familiar music tends to activate Karawitan musicians' brains in memory processing area: temporal lobe, specifically in the middle temporal gyrus. Meanwhile, a meta-analysis study [10] showed different results where the location activated by familiar music left superior frontal gyrus as the most activated area, followed by the ventral lateral as the second-highest area activated by familiar music. The assumption is that the frontal gyrus being activated by the semantic memory of

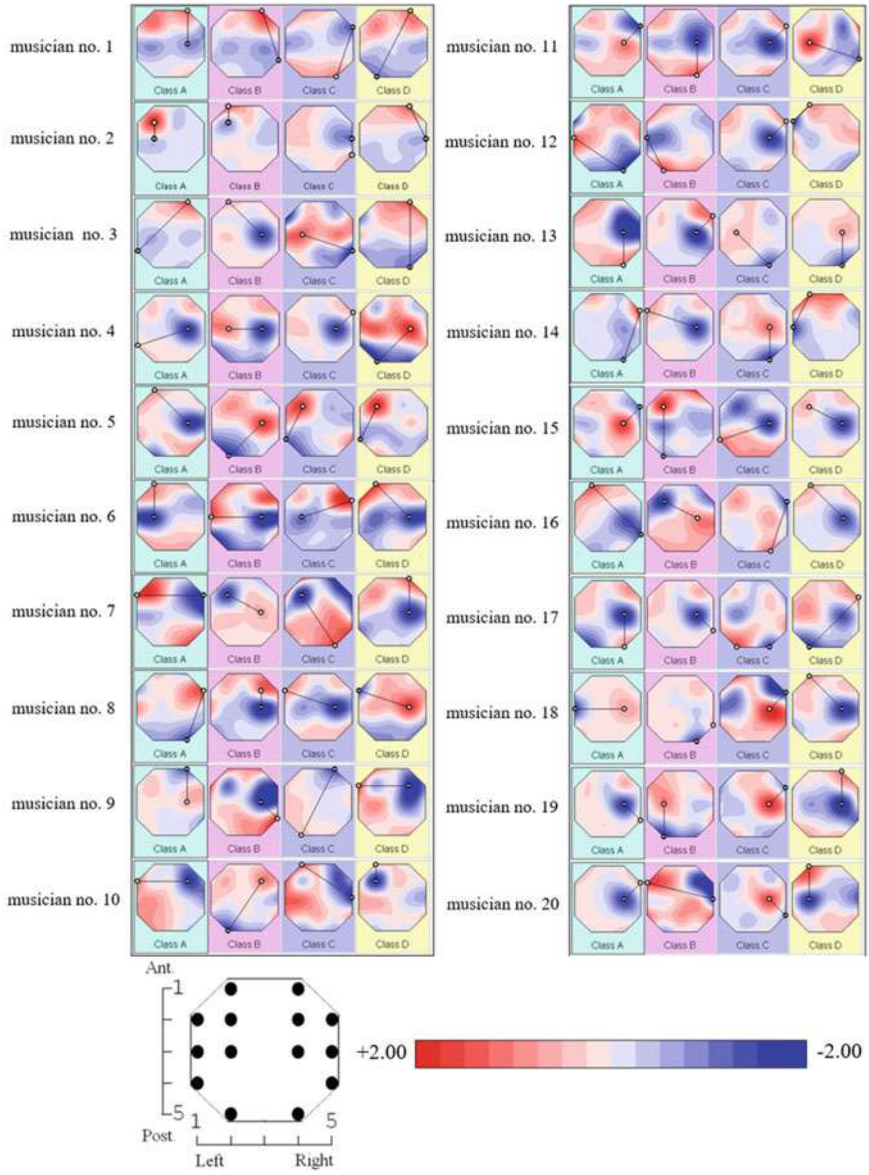
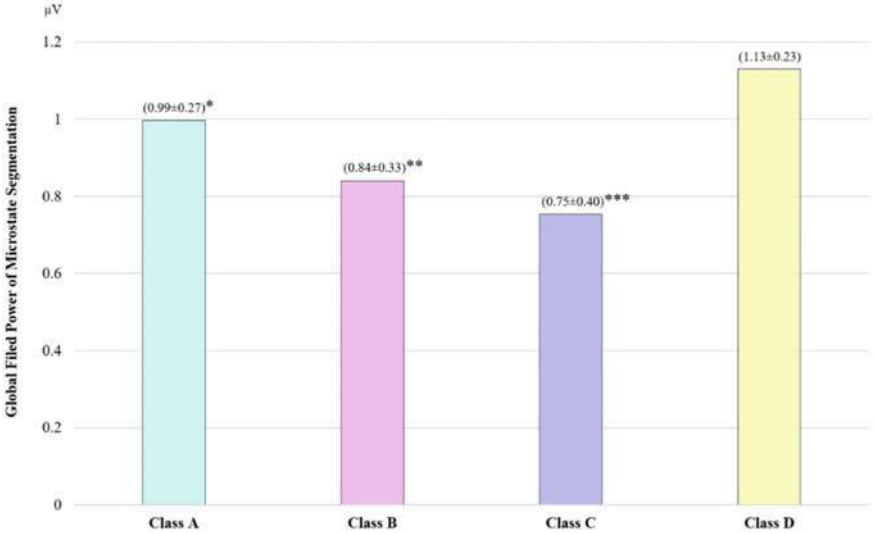


Fig. 3 Momentary potential distribution of centroids positive and negative spontaneous EEG map series of each map across participants. Schematic of the electrode array on the scalp; the head is seen from above, left ear left. Tick marks at the coordinate axes demonstrate electrode numbers, in anterior-posterior direction (vertical axis), and in the left-right direction (horizontal axis). Centroids of the positive and negative map area computed for spatial feature extraction. The configuration of the map was described by the anterior-posterior and left-right coordinates of the centroid extreme positive and negative area

Table 1 Microstate segmentation and electrical activities of each spontaneous EEG map series in the Karawitan musicians' brain while listening to *Gendhing Lancaran*

Microstate segmentation	Electrical activities (μV)
Class A	0.99 ± 0.27
Class B	0.84 ± 0.33
Class C	0.75 ± 0.40
Class D	1.13 ± 0.23



*$p<0.05$; **$p<0.01$; ***$p<0.001$

Fig. 4 Global filed power (GFP) peak amplitudes of electrical activities of Karawitan musicians' brains while listening to *Gendhing Lancaran*

certain familiar music, and meanwhile the ventral lateral, which is related to the motor cortex, might be stimulated by the motoric response given by the subject when they are anticipating rhythms of familiar music they listened to. The different results when Freitas et al. [10] study showed the ventral lateral (VL) as an area activated by familiar music might not happen in our research because we have forbidden our subject to make any movement during the listening session to avoid biased data caused by body movement. In terms of musical experience and perception, a previous study [11] demonstrated through a cultural difference approach. Furthermore, a previous study has been shown the auditory sequence memory in the musicians' brains by explaining how the early musical experience could support this mechanism [12]. In this previous study, they employed musicians, gymnasts, video game players, and psychology students. The memory task on audio, visual, and audio-visual stimuli was applied to all participated subjects. The results showed an insignificant difference in the visual or audio-visual task but the significant difference in the audio task where musicians scored higher. Finally, a longitudinal study in early childhood also showed a change in brain structure in children after only fifteen months of musical training [13]. A previous study also showed that simultaneous activation of widely distributed neural networks created the human brain's processing of information. This brain processing composed of convergence and sequential routing as well as large parallel processing of information in the brain [14].

The momentary state of all neural networks is equivalent to the global functional state of the brain [15]. This corresponds to what is momentarily represented in the mind [16]. When the brain receives information from the external environment, from the body, or from itself, it will produce an overt or covert response including also the possibility of not reacting at all [17, 18]. In terms of brain states, this means that the state of the brain has to move from an initial representation of the input information toward an eventual representation of the internal or external response [16–18]. The determinants would be varied as a function of the progress of this brain's information processing. Generally, automatic parallel processing would largely control the internal representations of stimuli activation. Nevertheless, the result regularly relies upon the internal state of the human brain during information income. Moreover, it was limited capacity processes as the use of different strategies that were guided by conscious control [19, 20]. Brain states are suggested to represent both bottom-up and top-down processes [21]. The putative, short functional states, will be called microstates following [22]. They interact both with the brain's microstate [23], as covering arousal, motivation, and emotional state and the anatomical structure of the brain which includes long-term memory and genetically inherited information [17, 18, 20]. The sequence of adaptive functional microstates of the brain eventually can induce limited capacity processes like conscious experience. The distribution of the electric field on the scalp reflects the spatial patterns of neural activity in the brain. Any changes in the geometry of neural activity in the brain could influence the changes of spatial distribution, and other neural elements become active. When this occurs, it is reasonable to assume that the brain has changed its functional state [17, 18, 20, 23]. The examination of sequences of momentary field maps shows that these changes of the brain electrical field topography occur in jumps taking place in the millisecond time range; the transitions between one field topography to the next one are not smooth but tend to occur quickly and stepwise, separated by longer periods of relatively stable field topography [7, 8, 24–26]. In order to obtain insights into important features of the brain's processing of information, the distribution and time course of the momentary brain electric filed topography should be further investigated.

5 Conclusion

Applying the electroencephalographic recording, the results showed higher brain activity of Karawitan musician's brains while listening to familiar music, *Gendhing Lancaran,* traditional music of Java. The dominant brain activity happened in the temporal lobe while Karawitan musicians listened to *Gendhing Lancaran.* Instead of differentiating brain activity between musician and non-musician, the present study demonstrated the different brain activity while musicians listened to music regarding their musical experience. This result will evoke music's integration to further consideration, such as the neuroscience of music.

Acknowledgements This study was technically supported by Mae Fah Luang University (electroencephalogram laboratory 2019), and Brain Science and Engineering Innovation Research Group, Mae Fah Luang University, Thailand. We thank all subjects who participated in this study.

Contributors

IKW originated the concept and recorded the data; IKW, D, FT, and PS performed literature search; PS critically analyzed data, reviewed, drafted the manuscript, and revised the final version. All authors approved the final version.

References

1. Patston, L.L.M., Tippett, L.J.: The effect of background music on cognitive performance in musicians and nonmusicians. Music. Percept. **29**(2), 173–183 (2011)
2. Husain, G., Thompson, W., Schellenberg, E.: Effects of musical tempo and mode on arousal, mood, and spatial abilities. Music. Percept.: Interdiscip. J. **20**(2), 151–171 (2002)
3. Sluming, V., Brooks, J., Howard, M., Downes, J.J., Roberts, N.: Broca's area supports enhanced visuospatial cognition in ochestral musicians. J. Neurosci. **27**(14), 3799–3806 (2007)
4. Gaser, C., Schlaug, G.: Brain structures differ between musicians and non-musicians. J. Neurosci. **23**(27), 9240–9245 (2003)
5. Schlaug, G.: The brain of musicians. In: Peretz, I., Zatorre, R.J. (eds.) The Cognitive Neuroscience of Music, pp. 366–381. Oxford University Press, New York (2003)
6. Lehmann, D., Skrandies, W.: Reference-free identification of components of checkerboard-evoked multichannel potential fields. Electroencephalogr. Clin. Neurophysiol. **48**(6), 609–621 (1980)
7. Lehmann, D.: Principles of spatial analysis. In: Gevins, A.S., Remond, A. (eds.) Handbook of Electroencephalography and Clinical Neurophysiology, Volume 1: Methods of Analysis of Brain Electrical and Magnetic Signals, pp. 309–354. Elsevier, Amsterdam (1987)
8. Wackermann, J., Lehmann, D., Michel, C.M., Strik, W.K.: Adaptive segmentation of spontaneous EEG map series into spatially defined microstates. Int. J. Psychophysiol. **14**(3), 269–283 (1993)
9. Pereira, C.S., Teixeira, J., Figueiredo, P., Xavier, J., Castro, S.L., Brattico, E.: Music and emotions in the brain: Familiarity matters. PLoS ONE **6**(11), e27241 (2011)
10. Freitas, C., Manzato, E., Burini, A., Taylor, M.J., Lerch, J.P., Anagnostou, E.: Neural correlates of familiarity in music listening: A systematic review and a neuroimaging meta-analysis. Front. Neurosci. **12**, 686 (2018)
11. Cross, I.: Music, cognition, culture, and evolution. In: Peretz, I., Zatorre, R.J. (eds.) The Cognitive Neuroscience of Music, pp. 42–56. Oxford University Press, New York (2003)
12. Tierney, A.T., Bergeson-Dana, T.R., Pisoni, D.B.: Effects of early musical experience on auditory sequence memory. Empir. Music. Rev. **3**(4), 178–186 (2008)
13. Hyde, K.L., Lerch, J., Norton, A., Forgeard, M., Winner, E., Evans, A.C., Schlaug, G.: The effects of musical training on structural brain development: A longitudinal study. Ann. N. Y. Acad. Sci. **1169**(1), 182–186 (2009)
14. Mesulam, M.-M.: Large-scale neurocognitive networks and distributed processing for attention, language and memory. Ann. Neurol. **28**, 597–613 (1990)
15. Ashby, R.W.: Design for a Brain: The Origin of Adaptive Behaviour, 2nd edn. Wiley, New York (1960)

16. Bunge, M.: Emergence and the mind. Neurosci. **2**(4), 501–509 (1977)
17. Koukkou, M., Lehmann, D.: An information-processing perspective of psychophysiological measurements. J. Psychophysiol. **1**(2), 109–112 (1987)
18. Koukkou, M., Lehmann, D.: A reply to R.C. Howard's commentary on our paper: An information-processing perspective of psychophysiological measurements. J. Psychophysiol. **1**, 219–220 (1987)
19. Posner, M.I., Snyder, C.R.R.: Attention and cognitive control. In: Solso, R.L. (ed.) Information Processing and Cognition, pp. 55–85. Erlbaum, Hillsdale NJ (1975)
20. Koukkou, M., Lehmann, D.: Dreaming: The functional state-shift hypothesis. A neuropsychophysiological model. Br. J. Psychiatry **142**, 221–231 (1983)
21. Dixon, N.F.: The conscious / unconscious interface: Contributions to an understanding. Psychol. Res. Bull. **5**, 1–15 (1981)
22. Lehmann, D.: Brain electric microstates and cognition: The atoms of thought. In: John, E.R. (ed.) Machinery of the Mind, pp. 209–244. Birkhauser, Boston (1990)
23. Lehmann, D.: Brain electric fields and brain functional states. In: Friedrich, R., Wunderlin, A. (eds.) Evolution of Dynamical Structures in Complex Systems, Springer Proceedings in Physics, vol. 69, pp. 235–248. Springer, Berlin (1992)
24. Lehmann, D., Skrandies, W.: Segmentation of evoked potentials based on spatial field configuration in multichannel recordings. In: McCallum, W.C., Zappoli, R., Denoth, F. (eds), Cerebral Psychophysiology: Studies in Event-Related Potentials. Electroencephalography and Clinical Neurophysiology. Suppl. **38**, 506–512 (1986)
25. Lehmann, D.: Multichannel topography of human alpha EEG fields. Electroencephalogr. Clin. Neurophysiol. **31**(5), 439–449 (1971)
26. Brandeis, D., Lehmann, D.: Segments of event-related potential map series reveal landscape changes with visual attention and subjective contour. Electroencephalogr. Clin. Neurophysiol. **73**(6), 507–519 (1989)

MIMO-Based 5G Data Communication Systems

Shubham Mathesul, Ayush Rambhad, Parth Shrivastav, and Sudhanshu Gonge

Abstract Many of the main targets or expectations that need to be met in the immediate term, i.e., after 5G, are expanded efficiency, higher data rate, reduced average latency, and enhanced coverage quality. Energy use for the networks is a big concern. To satisfy these requirements, there need to be dramatic changes in the design of the telecommunications network. This paper discusses the findings of a comprehensive study on the telecommunications network of the fifth generation (5G), including some of the main new innovations that further develop the infrastructure. The main subject of this comprehensive study is the 5G wireless network, massive MIMO infrastructure, and device-to-device compatibility (D2D). A comprehensive survey is provided in Sect. 4 addressing detailed study about deployments and predictions.

Keywords 5G · Massive MIMO · D2D · Cloud · Mm-wave · Relay · Small cell

1 Introduction

To address the presumptions and demands in the immediate future, today's wireless-based networks would have to evolve in different respects, even in the modern future. Cisco expects 5.5 billion individuals would hold smart phones by 2021. Tens of millions of these smartphone consumers in the UK alone would typically use 20 GB of data every month and use more than 25 separate digital devices in their day-to-day life. New network members such as high-speed packet access (HSPA) and long-term

S. Mathesul (✉) · A. Rambhad · P. Shrivastav · S. Gonge
Department of Computer Engineering, Vishwakarma Institute of Technology, Pune, India
e-mail: shubham.mathesul19@vit.edu

A. Rambhad
e-mail: ayush.rambhad19@vit.edu

P. Shrivastav
e-mail: parth.shrivastav19@vit.edu

S. Gonge
e-mail: sudhanshu.gonge@vit.edu

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_44

evolution (LTE) were initiated as part of the ongoing development in cellular technology. New elements, though, may also be future emerging wireless applications complementing existing technologies. Specimen of these latest generation devices is used to reach the bandwidth in various forms and to achieve significantly larger frequency levels. Among the examples are large antenna arrays, direct D2D connectivity, and ultra-dense deployments [1]. Mobile cellular networking brought analog voice calls to modern digital technologies following its introduction in the late 1970s. It is now capable of providing high-quality mobile telecommunications networks and several megabits per second end user data speeds across large regions. The substantial developments in the capacity of mobile networking networks, coupled launching of mobile devices like smartphones and tablets have culminated in the explosion of modern technologies that can be found in mobile access cases and an unprecedented increase of network traffic resulting. Within this article, we identify the new innovations that potential wireless communication will face and test them, thus supporting the networked community. In fact, there are growing network routes for overcoming these problems [1]. Our dream of the future is a networked society with unbounded exposure to intelligence and information sharing, available at any moment, everywhere and for all. To understand this vision, the future of current wireless systems requires to be explored with new product components. Present wireless innovations, such as third-generation partnership project (3GPP) LTE technology, HSPA and Wi-Fi, would integrate emerging infrastructure components and will help address potential needs.

1.1 5G

With an increasing rise in consumer demand, 4G can soon be quickly supplemented by 5G of beam division multiple access (BDMA) enhanced network technologies and multi-access non- and quasi-orthogonal or filter bank multi-carrier (FBMC). The concept behind the BDMA approach is explained by taking into account the link situation with the cell stations at the base line. In this correspondence, each mobile station is given an orthogonal beam, and the BDMA technique divides antenna beams by mobile station location to provide simultaneous connections to cell stations, thereby raising the efficiency of the system. The assumption of transitioning to 5G is focused on existing drifts, and it is widely believed that 5G wireless networks have to tackle six problems that are not handled successfully by 4G, i.e., greater efficiency, higher data rate, reduced end-to-end latency, large mobile availability, lower cost, and better consistency of service supply [1, 2]. Such problems are described succinct in Fig. 1. Together with other possible facilitators, the summary of the 5G obstacles, facilitators, and associated architecture principles is shown in Fig. 1. IEEE 802.11ac, 802.11ad, and 802.11af specifications newly adopted are quite useful and serve as building blocks on the journey toward 5G [2]. The functional comparison between 802.11 standards is shown in Table 1.

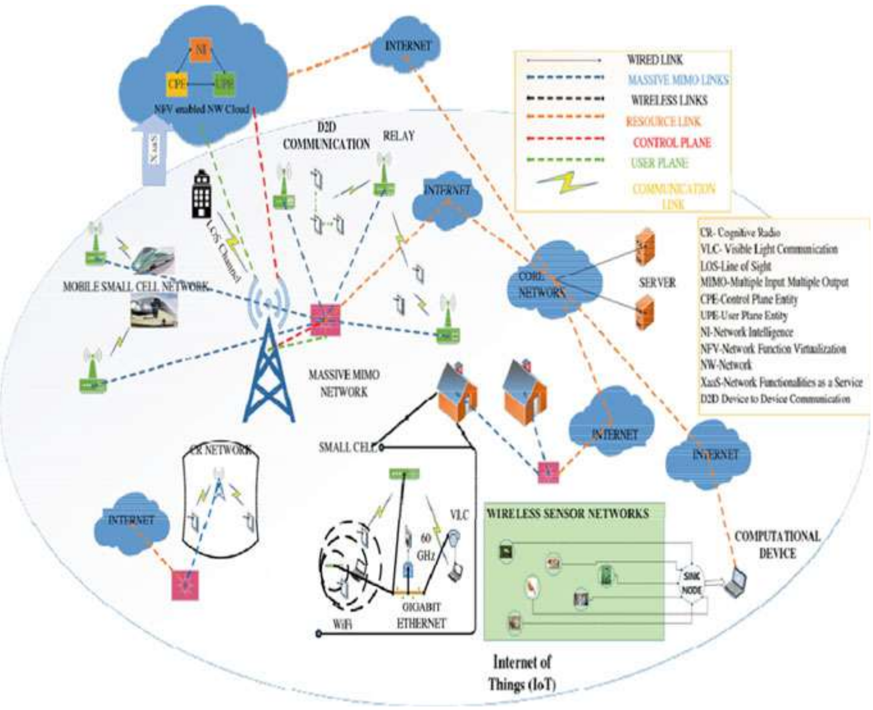


Fig. 1 Design of the general 5G wireless network

2 5G Cellular Network Architecture

To foresee the 5G network on the ground now, it is apparent that the variety of networks and networking topologies is used for data communication, which are at a standstill and require drastic upgrades. Present innovations such as OFDMA are expected to operate over at least the next 50 years. In fact, there is no reason for a transition in the network configuration that had arisen from 1G to 4G. Alternatively, only an application should be applied to the standard network to meet consumer requirements. This would allow service suppliers to migrate onto a 5G network as long as 4G is commercially developed [3]. A survey by general researchers noticed in [3] that the many wireless devices reside indoors for 80% of the day and outside for around 20% of the day. A casual research shows that the use of connectivity since the introduction of smartphones is growing tremendously. That prompted service providers to move to a 5G network as soon as 4G was introduced publicly. A general study finding noticed in Fig. 1 that the bulk of wireless users stay indoors for about 80% of the time and outside for around 20% of the time. Throughout the latest wireless cellular system, an additional terminal station in the center of a cell allows a smartphone device to connect from inside or outside. Therefore, for indoor users to link to the external terminal, the signals will have to travel through the indoor walls

Table 1 The functional comparison between 802.11 standards

802.11 standards	802.11ad	802.11ac	802.11af	802.11ah	802.11ax
Release date	December 2012	December 2013	February 2014	December 2016	September 2019
Frequency band	60 GHz	5 GHz	54–790 MHz	700–900 MHz	2.4, 5, 6 GHz
Modulation scheme	OFDM	MIMO-OFDM	MIMO-OFDM	MIMO-OFDM	MIMO-OFDM
Bandwidth	2160 MHz	20, 40, 80, 80 + 80, 160 MHz	6, 7, 8 MHz	1, 2, 4, 8, 16 MHz	20, 40, 80, 80 + 80, 160 MHz
Maximum data rate	6.76 Gbps	6.93 Gbps	26.7 Mbps	40 Mbps	9.60 Gbps
Range	10 m	35 m	1 km	1 km	120 m
Penetrating walls	Y	Y	Y	Y	Y
Advanced technologies used	MIMO, MU-MIMO	Beam forming	MIMO, MU-MIMO	MIMO, MU-MIMO	MIMO, MU-MIMO
NLOS	Y	Y	Y	Y	Y

resulting in very high losses of propagation, and this may decrease spectral capacity, data rate, and wireless transmission energy consumption costs. In order to address this obstacle, a new concept or modeling strategy that has arisen for the scheming of 5G cellular infrastructure is to differentiate setups from outside and within. Using this construction strategy, the lack of entry across the building's walls should be minimized marginally. This principle can be activated with the assistance of massive MIMO technology that deploys tens or hundreds of antenna modules in a globally distributed antenna network. MIMO systems usually use two or four antennas, but massive MIMO systems have been developed to enable the advantages of large-scale antenna components to be optimized in terms of enormous efficiency gains. To build or mount the massive MIMO network, the external terminal stations must be fitted with wide antenna clusters, and all of these are scattered around the hexagonal cell and fastened with optical fiber cables to the terminal station, assisted by massive MIMO technologies. External cell consumers are normally equipped with a variety of antenna modules; however, a broad virtual antenna array may be installed with cooperation, and together with base station antenna arrays create virtual massive MIMO links. Second, every building will have wide outdoor antenna clusters that will link to external base stations utilizing sight line components.

The wireless access points are linked to the wide antenna clusters within the house through cables for indoor computer connectivity. This would greatly boost the cellular system's energy consumption, cell level capacity, data rate, and spectral quality but at the risk of higher network costs. Through the adoption of such a system, indoor users would only have to link or interact through wireless access points within while wider antenna arrays outside the buildings stay mounted. Certain apps, such as Wi-Fi, small cell, ultra-wideband, millimeter-wave communications [3], and visible light communications, gain low-range connectivity and fast indoor networking data rates.

Since 5G cellular architecture is heterogeneous, macrocells, microcells, tiny cells, and relays must be used. Small cell system architecture is an essential component of the 5G cellular broadband network that involves satellite relays and portion mobile cell technology. There is need to develop maximum versatile application for cars and high-speed trains for better communication. Mobile small cells are mounted within the moving vehicles to communicate with users within the car, while the massive MIMO network of large antenna clusters is built outside the car to link with the base station outside the vehicle. According to the dispatcher, a cheap android cell is converted into a regular terminal station and its accompanying representatives are all treated at the base terminal as a single unit, demonstrating the above definition of indoor and outdoor separation. When seen in Table 2 [4], hand-held small cell customers provide a large utilization volume for low rate networks with greatly decreased overhead signaling, while the 5G cellular broadband network system comprises just two conceptual layers: a radio network and a network cloud. The radio network is composed of various forms of modules conducting specific functions. The network function virtualization (NFV) cloud comprises a user plane entity (UPE) and a control plane entity (CPE) executing higher-layer user and control plane functionalities, respectively. Unique network functionality as a service (XaaS) can offer as desired, and the pooling of resources is an example of the same.

Generally speaking, this current 5G cellular network design will provide a strong forum for potential 5G network standardization. This suggested framework also describes the importance of cloud network function virtualization (NFV) in the design of 5G cellular networks. This emerging 5G wireless network design has already implemented the idea of device-to-device (D2D) connectivity, small cell access points, and Internet of Things (IoT).

3 Technology Emerging for 5G Broadband Networks

The amount of Internet and wireless traffic is projected to rise a thousand fold in the next decade, powered by the estimated 50 billion paired gadgets connecting to the web by 2021, and all the data needs to be processed and exchanged anytime, anywhere. With a significant growth in the number of connected devices, there tend to be several problems that must be solved by growing power and enhancing energy quality, cost and application of the bandwidth and better scalability to handle the increased number of linked devices. For the vision of the future all communicating along with today’s network, the ultimate technological goal is to have an image of a structure that supports [5]:

Table 2 Wireless technology growth

Generations and technology	Data rate	Service	Frequency band	Launched in year	Forward error correction	Switching	Handoff
1G: Analog Cellular Technology	2.4 Kbps	Mobile calling	800 MHz	1970	NA	Circuit	Horizontal
2G: Digital Cellular Technology	64 Kbps	Digital voice and messaging with certain limit	850/900/1800/1900 MHz	1980	NA	Circuit	Horizontal
3G: Broad Bandwidth Code Division Multiple Access (CDMA) IP technology	2 Mbps	High-quality end-to-end audio, video and data	800/850/900/1800/1900/2100 MHz	1990	Turbo codes	Packet except for air interface	Horizontal
4G: Unified IP	200 Mbps	Dynamic information, variable devices	3.5 GHz initially	2000	Turbo codes	Packet only	Horizontal and vertical
5G: 4G + Beam Division Multiple Access (BDMA)	More than 1 Gbps	4G + AI capabilities	600 MHz-to-6 GHz, 24–86 GHz	2010	Low density parity check codes (LDPC)	Packet only	Horizontal and vertical

- 1000 times the amount of data per field,
- 10–100 times the number of linked devices,
- 10–100 times the average mobile data rate, and
- 10 times the battery life with high-quality massive machine communication (MMC) gadgets. The solution to some of the objectives is mentioned earlier [5].

3.1 Device-to-Device (D2D)

The connectivity refers to direct communication between devices enabling local sharing of consumer plane traffic without a network infrastructure.

3.2 Massive Machine Communications (MMC)

It would shape the foundation for the Internet of Things, spanning a broad variety of fields from the car industry, public protection, emergency care, and the medical field.

3.3 Moving Networks (MN)

It can strengthen and expand linkages between potentially broad numbers of communications devices that are traveling together.

3.4 Ultra-Dense Networks (UDN)

It would be the key factor aiming to improve bandwidth, improving the energy efficiency of wireless broadcasts, and allowing the greater use of underused spectrum.

3.5 Ultra-Reliable Networks (URNs)

It should make for high rates of connectivity. Many innovations, listed in expected importance, would be essential for potential wireless requirements, which is shown in Sect. 4.

4 Massive MIMO

Massive MIMO is an emerging system, updated from the latest version of MIMO [5]. The massive MIMO network utilizes antenna clusters that include a few hundred antennas that are concurrently in one position, with frequency slot servicing minimum 25–50 user terminals. Massive MIMO technology's aim is to provide all the benefits of MIMO, but on a wider scale. Massive MIMO is focused on multiplexing spatially, which also relies on providing channel status information on the base point, both from the uplink and downlink. It is not simple in downlink situation, but it is quick in uplink situation, as the terminals submit pilots. For every terminal, the channel response is determined dependent on the pilots [5, 6].

In typical MIMO schemes, the base terminal shall give the pilot waveforms to and on the basis of the terminals, and the terminal performs the route estimation, quantizes it, and feeds it back to the base terminal. This method is not feasible for massive MIMO structures, particularly under high conditions of mobility, for two reasons. Compared with traditional MIMO systems, massive MIMO systems will still need a vast number of identical spaces. First, when the number of antennas at base terminals often raises the amount of channel forecasts for each terminal also rises, which often demanded hundreds of times more uplink slots to report on the link responds to the base station. Massive MIMO technology is based on consistent phase signals from all antennas in base terminal, yet detection of these signals in computing is fast. The following are some important aspects of a massive MIMO network [6].

4.1 Massive MIMO Potential Energy and Power

The successful growth of capability is due to the spatial multiplexing technique employed in massive MIMO systems. As regards improving the radiated energy efficiency, the radiation can now be transmitted in small space regions owing to the rise in the amount of antennas. The principle of coherent front wave superposition is based on this. The main station little function to play after transmitting the formed signals obtained from antennas by verifying that or all the front waves released from the antennas that add constructively to the positions of the intended terminal and destructively elsewhere. Minimal pressing is used to prevent any remaining disturbance between the terminals, albeit at the cost of improved transmitting efficiency [6, 7].

The channel answers they collect are amplified due to their technical ease, i.e., the received signals, and since this is done in a distributed mode, with each antenna function autonomous, the worthiness of maximum ratio combining (MRC) is more like linked to zero forcing (ZF) [7], although ZF works almost as good as an orthodox MIMO device which MRC does not usually perform task. The key explanation for the effective usage of the MRC for a massive MIMO containing a vast number of main station antennas appears to be almost orthogonal [7].

We may work in a noise limited network using MRC receiver. MRC in massive MIMO network should scale down the power to the maximum practicable without really disrupting the total spectral performance and multi-user intrusion; however, the consequences of equipment shortcomings are likely to be offset by thermal noise. In this situation, ten times more terminals are operated concurrently in frequency resource.

4.2 Flexibility of Low Power and Cost Materials for Massive MIMO Systems

Massive MIMO has brought about a revolution in terms of design, schemes, and implementation. Massive MIMO devices require hundreds less than costly amplifiers than powerful 50-watt ultra-linear amplifiers, since earlier the peak capacity was in the mill watt scale, which is much higher than the latter found in traditional systems in general. It is distinct from traditional array systems, since it can only use a tiny antenna that is operated by high-power amplifiers that has a large effect. The main change is the removal of certain costly and large pieces, such as long coaxial cables [6–9].

The vibration, flickering, and equipment deficiencies can be compounded for the use of multiple antennas in massive MIMO networks, as signals from a large range of antennas are mixed outdoors. It condenses the constraints on the efficiency and linearity of each amplifier, and radio frequency chain and their reciprocal behavior are all relevant. It would improve massive MIMO's robustness toward fading and loss of one of the components of the antenna.

4.3 Degree of Freedom for a Massive MIMO Device

For illustration, ten terminals display the presence with 100 antennas while the remaining 90 "degrees of independence" are still usable. Such accessible degrees of freedom can be used for forming signals that would be hardware compatible. In particular, through the usage of very low-cost and power-efficient RF amplifiers, every antenna can relay small peak-to-average signals and a steady envelope [8, 9] at a moderate size, overall radiated output increased. Using the continuous pre-coding of multi-user envelopes, the signals generated from different antennas are neither beam-shaped nor symbol-weighted. Rather, in addition to the position of the terminals, a wave field is generated and analyzed so they will see the signals exactly what we wanted to let them see. Massive MIMO has a critical property that allows for this. The vast MIMO channel has wide null spaces where almost anything is feasible without disrupting the terminals.

Precisely in this null space, modules can be installed that allow the transmitted waveforms to meet the required envelope restrictions. Nonetheless, the transmission channels at the main station and each terminal can be managed without PSK modulation [8]. So any signal configuration may be used as a guide [8, 9].

The significant increase in energy quality allows massive MIMO systems to operate on the overall generated RF capacity by two measures of lower magnitude than with current technologies. It is important as the cellular base terminals use a great deal of energy and that is a concern. Moreover, since base stations can use fewer power, it may be powered by sustainable energy such as wind or solar, and then it is useful to install base terminal at areas where there is no electricity. Additionally, there would be far less heightened questions over radiation sensitivity.

4.4 Massive MIMO Calls Reduction in Air Connection Latency

In the next-generation networks, the principal issue is latency. The key source of latency fades in wireless contact. This phenomenon happens in the main station and terminal, i.e., as the signal is forwarded from the base terminal, it passes across various numerous pathways owing to the interference, absorption, and diffraction of the phenomena before entering the terminal. Once the pulse enters the terminal via these multiple paths, whether constructive or harmful, it interferes, in the case of waves coming destructively from these various routes, the pulse amplitude obtained decreases to a considerably low level. If the terminal is stuck in a decline, the route of transmission can have to wait until the data is obtained. Owing to the large number of antennas and the beam forming principle, massive MIMO should prohibit fading and thus latency cannot be decreased much further [9, 10].

4.5 Massive MIMO Multi-layer Exposure

The channel improves with the introduction of massive MIMO, and now the management of a frequency domain is not sufficient. OFDM provides any subcarrier in a massive MIMO network with exactly the same path resulting in the supply of each and every terminal can be supplied with maximum bandwidth, eliminating much of the terminated physical layer control signals [9, 10].

4.6 Massive MIMO Power Against Unintentional Intrusion and Intentional Jamming

Jamming civilian communications networks is a key concern and presents a significant challenge to information protection. Owing to the restricted capacity, it is just not feasible to transmit knowledge over time. Massive MIMO provides methods to boost wireless connectivity robustness with the assistance of numerous antennas. This provides plenty of flexibility and can be useful for canceling the signals from intended jammers. When massive MIMO networks use integrated channel prediction and decoding to predict channels rather than uplink pilots, the issue with the predicted jammers can be minimized drastically [9, 10].

The implications of massive MIMO networks should be analyzed from a theoretical knowledge viewpoint. Small MIMO systems will accomplish the exciting multiplexing advantage from minimal point-to-point MIMO systems and dispose of issues regardless of adverse transmission conditions [10–12]. The biggest field of focus in today's cellular wireless connection is around energy consumption and resource optimization. So, several researchers are focusing on rising energy conservation and resource management. Research on power conservation was carried out in [12] and seen in Fig. 2.

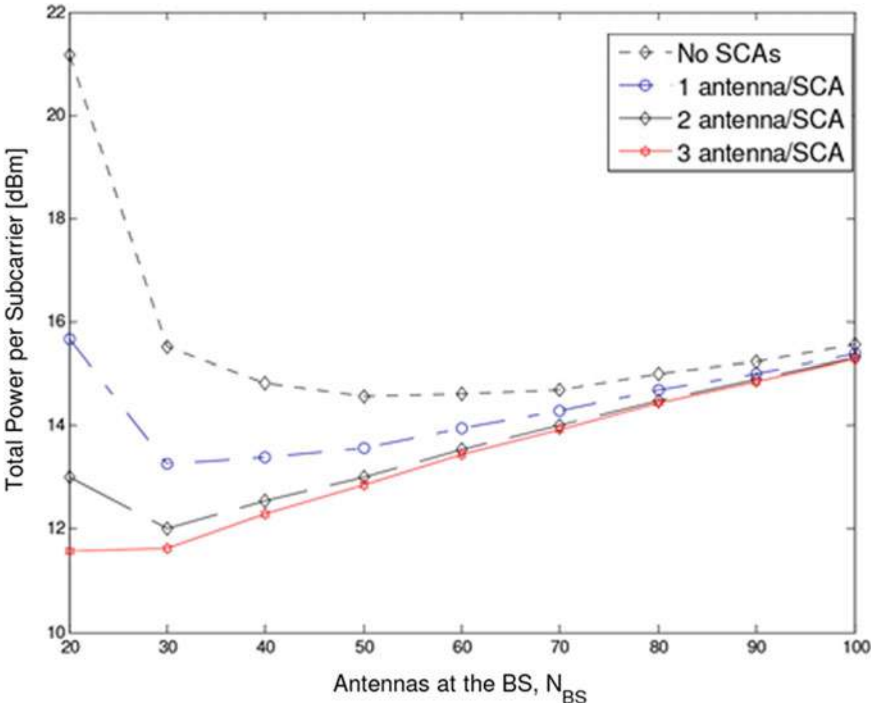


Fig. 2 Total overall power usage of small cell connection points in scenario

Figure 2 clearly demonstrates explicitly that if we maximize the amount of antennas both at the main station along with the small cell access point, unlike the case with no antenna at the small cell connection stage, the combined output per subcarrier dropped to 10 times. There are peak stages, though, where extra equipment can no doubt outweigh the overall power. Table 3 offers a short overview of the research performed on the massive MIMO system, enhancing energy efficiency and harnessing wireless cellular network power.

Table 3 Effect of the wireless cellular system MIMO technology over energy efficiency

References	Objectives	Observations
[12]	The main objective is for improving the quality of cellular resources and demystifying the network topology for better spatial reuse and utilizing two densification strategies, including large multiple input multiple output (MIMO) main stations and limited access points	It is seen that, during incorporating more equipment, power usage can be reduced considerably, as the dynamic component reduces, resulting in less lack of dissemination and increased energy performance It is observed that the improvement in energy quality can be accomplished by the introduction of a network topology incorporating massive MIMO and the deployment of a few single small cell antenna access points in areas with no additional hardware
[13]	Effect on energy efficiency (EE) of a massive MIMO system in terms of following points: The amount of antennas usable at the base station (BS), i.e., M The number of active user equipments (UEs), i.e., K It transmits power It gives maximum area throughput For different linear processing, schemes have objectives like: It has zero forcing (ZF) property It provides maximum ratio transmission/combining (MRT/MRC) It has minimum mean square error (MMSE) processing, which is very important parameter	The energy efficiency (EE) values for different linear processing schemes and with different values of M & K. The objectives base on it are as follows: It has ZF processing with values $M = 165$ and $K = 104$, EE = 30.7 Mbit/J The MMSE processing has different value of $M = 145$ and $K = 95$, EE = 30.3 Mbit/J In MRT/MRC processing having the parameter value $M = 81$ and $K = 77$, EE = 9.86 Mbit/J It is also observed that for optimum energy output as a function of the number of base station antennas, MMSE and ZF are approximately equivalent at high SNR The strongest energy-efficient approach of all the manufacturing schemes considered is to maximize radio frequency capacity with M at the same moment, and there is a eightfold improvement in field, ZF and MMSE processing throughput as compared to MRT/MRC is also observed

5 A Millimeter Wireless Network Wave Approach

The wireless industry has evolved every day, and given the industrial developers 'attempts to develop the professional wireless technology, the wireless industry continues to face the daunting demands of the new technologies. Recent computer and networking advances and the advent of mobile phones and the ability to access the Web pose fresh anxieties toward the telecom industry. Such demands and anxieties will rise over the coming years for 4G LTE and suggest that a question of congestion over cellular networks may emerge sometime around 2021. To satisfy the growing demands of customers, the implementation of new technology and architectures will be essential for the testing sector. The new report envisages a vision for broadband, where download rates rise to multi-gigabit per second. Such large data speeds can be reached with steerable antennas and the wave width of a millimeter, thus helping broadband and backhaul networks [14–16].

Latest work has indicated that 2.6 GHz mm-wave radio spectrum frequencies may complement the already crowded 700 MHz wireless band. The viability of wireless communication using a millimeter-wave is demonstrated by the truth of using high-profit, steerable antennas for mobile and base terminals and cost-effective CMOS systems can now work well into wave frequency bands [16, 17]. In comparison, the usage of millimeter-wave carrier frequencies would lead in higher data access speeds and wider spectrum networks and wireless carriers now use 20 MHz channels and 4G subscribers must also dramatically increase the frequency reach. Capacity will also improve with the rise in capacity, although latency will decrease, resulting in smoother Internet-based connectivity and apps such as streaming in real time. As the range of the millimeter-wave signals is very low, it will include polarization and various spatial processing techniques, such as massive MIMO and adaptive beam shaping [16, 17]. Despite the huge improvement in speed, the Internet connections to heavily populated regions can now accommodate more power than current 4G networks. Similarly, base terminals regularly raising the cell coverage areas for spatial reuse, mutual MIMO, relays, and disturbance reduction between base terminals. Since the base terminals are more clustered and located in regional centers, this will minimize costs per base station. Spectrum ranges of more than 1 GHz of bandwidth are commonly found in the bands 28 and 38 GHz.

When it comes to a product concept, the antenna is essentially very similar to the combined 28 GHz radio frequency circuit and the front-end assembly, since the attenuation signal will be high at 28 GHz. Realizing the antenna arrangement directly on the 5G wireless devices, printed circuit board minimizes the absence of antenna convergence with the optimized radio frequency circuit, which means that the usage of the radio spectrum 5G network bases until the intermediate spectrum stage relies on the cell phone range of 28 GHz antenna array.

Taking this definition into account, for millimeter-wave 5G cellular applications, a minimum set of two antenna arrays of 28 GHz is suggested in [16, 17], both antenna arrays are used both at the top and the bottom of the cellular package. Table 4

Table 4 Antenna array setup for 5G network wireless mobile and its contrast with the 4G standard for 28 GHz

Cellular standards	4G	5G [16, 21, 22]
Antenna type	Sub wavelength antennas	Phased array antennas
Radiation patterns	Omnidirectional	Directional fan-beam
Diversity and MIMO	Yes	Yes
Polarization	Single and constant	Multiple and reconfigurable

displays the 28 GHz antenna system design for 5G network cellular terminals and their contrast with the 4G model.

The millimeter spectrum of waves is underused and lies inactive before the present years. The main cause for this abuse is its inadequacy in cellular contact due to unfriendly channel conditions such as track loss effect, absorption by weather and by mist, less diffraction, and pervasion of barriers and objects.

There is another explanation that is unsuitable due to high phase noise and unnecessary expense of the equipment. Yet the key explanation for this is that the broad unlicensed band about 60 GHz was mainly ideal for very limited transmission [17, 18]. Thus, all fixed wireless implementations in the GHz 28, 38, 71–76, and 81–86 GHz and Wi-Fi in the mainstream 802.11ad in the 60 GHz band have been stressed.

Half-conductors continue to rise as their cost and energy consumption values decline steadily due to the creation of the aforementioned short-range standards.

6 Service Management Trends and Quality to 5G

In 2020, 5G services began to emerge on the market. Given the growing amount of traffic in mobile networks and the proliferation of wireless apps with a variety of applications, standard of service is expected to significantly improve customers. Many general developments related to 5G can be clarified in terms of network activity and amount of user-to-machine links in mobile apps.

Projections centered is shown in Fig. 3. The survey conducted by statistic is shown in Fig. 3. In 2021, the amount of machine-to-machine (M2M) connections in the mobile operator networks will reach 3.3 billion [18], which is double the current levels, and telecom companies will provide more than 26 billion phone to phone links in 2022. The machine-to-machine share of the total number of connections in the telecom provider's networks would rise during the same span from 18% in 2018 to 20% in 2021 and 22% in 2022.

A main phenomenon is connectivity, as wireless Internet penetration is projected to grow in the coming years with more than 2.4 billion subscribers worldwide (as

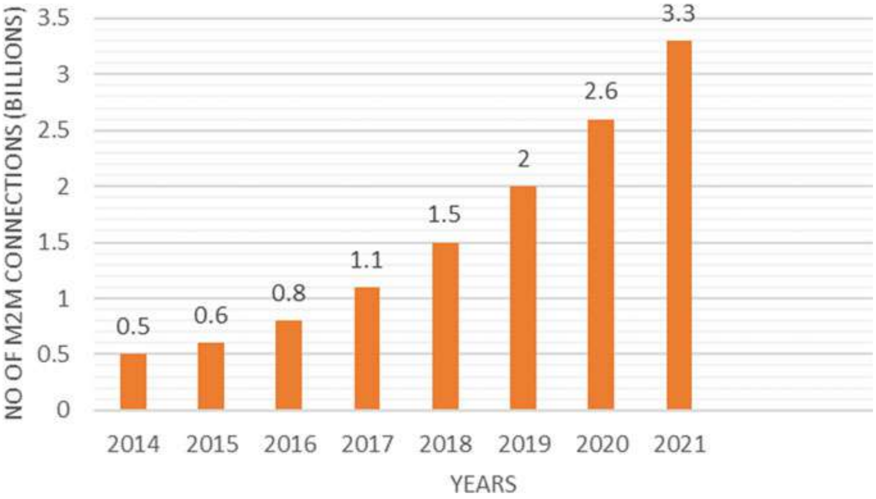


Fig. 3 M2M connections with respect to years [19]

surveyed in June 2012). The predicted 40-fold rise between 2010 and 2015 as shown in Fig. 4 that supports computer communications and information traffic, and a 1000-fold rise over ten years is expected. This degree of development pushes network providers to offer all forms of heterogeneous and updated Internet-based networks and applications with global broadband connectivity [20–22].

However, the control of the standard of service in 5G can be understood in terms of cell spectral performance and latency. The cell's spectral power is shown in Fig. 5 in 5G networks for different transmission channels. Improved spectrum performance of 5G networks can be accomplished in wireless access networks using non-orthogonal

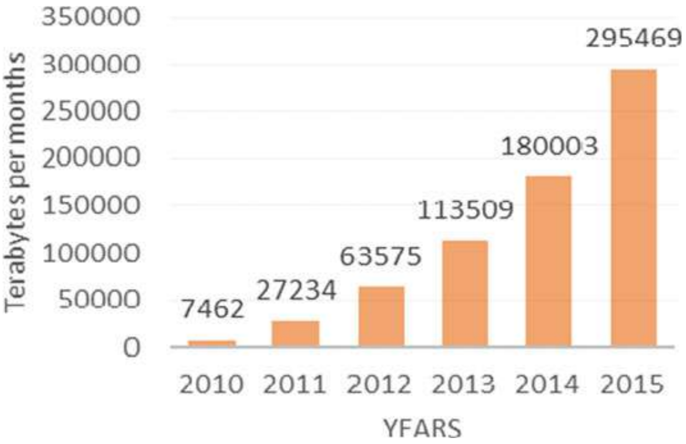


Fig. 4 Machine-to-machine traffic analysis (2010–2015)

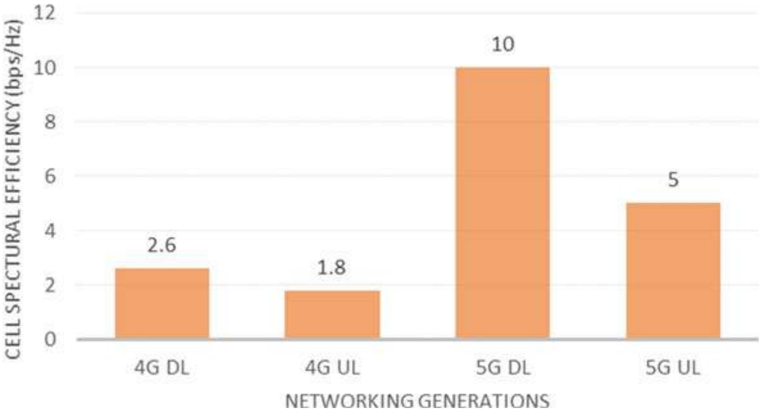


Fig. 5 Cell spectral efficiency in 5G networks [20]

communication technologies by utilizing non-orthogonal signals. Comparison of these factors with the same criteria for 4G networks indicates 3–5 times increase in spectral performance.

Assessment of power delay criteria and customer planes to monitor traffic and system movement is shown in Fig. 6. This figure indicates that the demands imposed on 5G networks are twice as large for user air traffic and ten times higher for customer transport plane, while we will see the introduction of 5G telecommunications networks in the USA has already started. A variety of businesses seeks to leverage on the expected growth [19–22].

The survey conducted by TeleGeography shows that total 69 deployments of 5G have made all over the world (as of April 15, 2020). As we can see highest deployments had been made by the USA. It is predicted that by 2025 most of the parts of country would be 5G enabled [19–22].

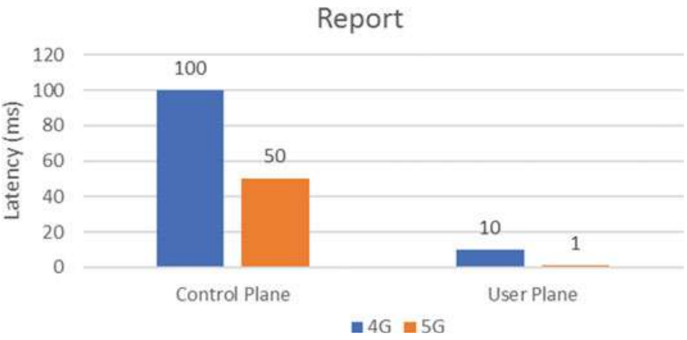


Fig. 6 Latency comparison reports with 4G and 5G

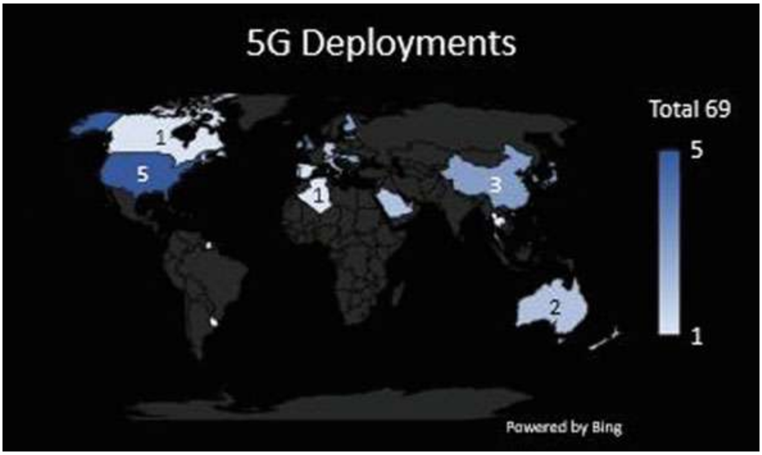


Fig. 7 5G deployments as of 15-04-2020

Although the wide-ranging smartphone 5G rollout is several years away, a variety of firms is currently engaged in the production and manufacturing of 5G infrastructure ,and installation of its network around the world is shown in Fig. 7 [19–22].

7 Conclusion

A comprehensive survey was performed in this paper on power efficiency requirements of 5G wireless cellular communications networks established that are data rate, spectral efficiency, latency, energy consumption, and service quality. Through this paper, we described a 5G cellular network system with massive MIMO technologies. Any short-range networking innovations have been described, such as Wi-Fi, tiny cell, visible light connectivity, and millimeter-wave communication technology, which promises a bright future in terms of improved coverage and higher data speeds for in-house consumers and decreases the demand from outside base stations at the same time. In particular, 5G offers significant changes in pace, responsiveness, and size to help all manner of bandwidth-hungry applications and technologies. 5G is already in use, and we cannot say network providers plan to offer. Several primary new innovations were also addressed that may be included in 5G network networks to fulfill the expected efficiency specifications, such as massive MIMO and device-to-device communication. We tackled convergence on millimeter waves across radio access networks and software-based networks and cloud networking in general. This paper may provide a strong forum for inspiring researchers to further tackle various forms of challenges in the next century networks.

References

1. Baldemair, R., et al.: Evolving wireless communications: addressing the challenges and expectations of the future. IEEE Veh. Technol. Mag. **8**(1), 24–30 (2013)
2. Ong, E.H., Kneckt, J., Alanen, O., Chang, Z., Huovinen, T., Nihtila, T.: IEEE 802.11ac: enhancements for very-high throughput WLANs. In: 2011 IEEE Personal Indoor and Mobile Radio Communications (2011)
3. Perahia, E., Gong, M.X.: Gigabit wireless LANs: an overview of IEEE 802.11 ac and 802.11 ad. ACM SIGMOBILE Mob. Comput. Commun. Rev. **15**(3), 23–33 (2011)
4. Chandrasekhar, V., Andrews, J.G., Gatherer, A.: Femtocell networks: a survey. IEEE Commun. Mag. **46**(9), 59–67 (2008)
5. Rusek, F., et al.: Scaling up MIMO: opportunities and challenges with very large arrays. IEEE Signal Process. Mag. **30**(1), 40–60 (2013)
6. Bleicher, A.: Millimeter waves may be the future of 5G phones. IEEE Spectrum (2013)
7. Jin, S., Wang, X., Li, Z., Wong, K., Huang, Y., Tang, X.: On massive MIMO zero-forcing transceiver using time-shifted pilots. IEEE Trans. Veh. Technol. **65**(1), 59–74 (2016). https://doi.org/10.1109/TVT.2015.2391192
8. Chatzidiamantis, N.D., Lioumpas, A.S., Karagiannidis, G.K., Arnon, S.: Adaptive subcarrier PSK intensity modulation in free space optical systems. IEEE Trans. Commun. **59**(5), 1368–1377 (2011). https://doi.org/10.1109/TCOMM.2011.022811.100078
9. Rusek, F., Persson, D., Lau, B., Larsson, E., Marzetta, T., Edfors, O., Tufvesson, F.: Scaling up MIMO opportunities and challenges with very large arrays. IEEE Signal Process. Mag. **30**(1), 40–60 (2013). https://doi.org/10.1109/msp.2011.2178495
10. Bethanabhotla, D., Bursalioglu, O., Papadopoulos, H.C., Caire, G.: User association and load balancing for cellular massive MIMO. In: Proceedings of ITA, Feb 2014, pp. 1–10
11. Peng, M., Liang, D., Wei, Y., Li, J., Chen, H.-H.: Selfconfiguration and self-optimization in LTE-advanced heterogeneous networks. Commun. Mag. IEEE **51**(5), 36–45 (2013)
12. Peng, M., Wang, C.g., Li, J., Xiang, H., Lau, V.: Recent advances in underlay heterogeneous networks: interference control, resource allocation, and self-organization. Commun. Surv. Tutor. IEEE **17**(2), 700, 729 (2015)
13. Tehrani, M.N., Uysal, M., Yanikomeroglu, H.: Device-to-device communication in 5G cellular networks: challenges, solutions, and future directions. IEEE Commun. Mag. **52**(5), 86–92 (2014)
14. Andrews, J.G., et al.: What will 5G be? IEEE J. Sel. Areas Commun. **32**, 1065–1082 (2014)
15. Singh, S., Andrews, J.G.: Joint resource partitioning and offloading in heterogeneous cellular networks. IEEE Trans. Wirel. Commun. **13**(2), 888–901 (2014)
16. Hong, W., Baek, K.-H., Lee, Y., Kim, Y., Ko, S.-T.: Study and prototyping of practically large-scale mm wave antenna systems for 5G cellular devices. IEEE Commun. Mag. **52**(9), 63–69 (2014)
17. Tikhvinskiy, V., Bochechka, G.: Perspectives and quality of service requirements in 5G networks. J. Telecommun. Inf. Technol. Poland **1**, 23–26 (2015)
18. Park, Y.: In: 5G vision and requirements of 5G forum, Korea, Feb 2014
19. Wunder, G.: 5th generation non-orthogonal waveforms for asynchronous signaling. In: COST Meeting, Ferrara (Italy) (2014)
20. Olsson, M., et al.: 5GrEEn: towards green 5G mobile networks. In: International Conference on Wireless and Mobile Computing, Networking and Communications, Oct 2013, pp. 212–216
21. 4G Americas' summary of global 5G initiatives. White Paper, 4G Americas (2014) [Online]. Available at: http://www.4gamericas.org/documents/20144GASummaryofGlobal5GInitiativesFinal.pdf
22. Eze, K.G., Sadiku, M.N.O., Musa, S.M.: 5G Wireless Technology: A Primer (2018), pp. 62–64

Android Malware Classification Based on Static Features of an Application

S. D. Ashwini, Manisha Pai, and J. Sangeetha

Abstract Android is the most sought-after mobile platform that has changed what mobiles can do. Due to this, a continuous increase in android malware applications has been seen that poses a significant hazard to users. Thus, the detection of malware applications in the Android environment has become a trending research field for cybersecurity researchers. Android malware detection depends on characterizing the Android application's functionalities. Over the years, malware has evolved and has become more sophisticated. Hence, it cannot be detected only using a single static feature as it might result in a high number of false negatives. We propose a detection model in this paper that accurately classifies the samples as malware or benign with fewer false positives and false negatives. We have used string features that include suspicious API calls, used permissions, requested permissions, filtered intents, hardware components, and restricted API calls. We have then employed four machine learning algorithms, namely, Ridge Classifier, XGBoost Classifier, Random Forest, and Support Vector Classifier to evaluate the effectiveness of the binary feature vector formed by the combination of these string features. It was noted that Random Forest achieved the highest score for accuracy, precision, recall, area under curve, and F1 score.

Keywords Static analysis · Android malware · Manifest file · Malware classification

S. D. Ashwini · M. Pai · J. Sangeetha (✉)
Department of Computer Science and Engineering, Ramaiah Institute of Technology, Bengaluru, India
e-mail: sangeethakirank@msrit.edu

S. D. Ashwini
e-mail: ashwini5896@gmail.com

M. Pai
e-mail: rdc2011mannu@gmail.com

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_45

1 Introduction

The usage of smartphones is increasing worldwide for both business and personal purposes. They run applications that can have direct access to sensitive information such as user's contacts, messages, emails, location, confidential files, etc. More significantly, these devices can access user's bank accounts either through credentials or digital wallets, hence posing a monetary risk. So far, android is the most popular platform for smartphones because of which they have become a lucrative target for attackers with malicious intent. Android applications can be installed by users from primary stores like Google play store and various other 3rd-party stores. With an increase in the amount of android malware, malware analysis has become a trending field for security researchers.

The principal aim of android security solutions is to defend the device and the data in it from harmful applications. To do that, one must recognize what type of malware it is, analyze their behavior, and classify them into their respective families based on their characteristics. There are 3 types of malware analysis: static, dynamic, and hybrid analysis [1]. Malware detection in static analysis is done based on features extracted from the Android apk file. In this approach, the application code is not executed but only observed i.e. only static features of the application are used to detect malware. The main disadvantage of this analysis is that it generates a high amount of false-positives. The dynamic analysis executes the application and observes how the android system interacts with it. This is comparatively time-consuming than the static approach but is effective in understanding the malware. The hybrid analysis takes the best features of both static and dynamic methods. In our paper, we have used static analysis to detect malware applications with fewer false positives.

The rest of our paper is structured as follows—Sect. 2 addresses the importance of android applications and their components. Section 3 contains a survey of related works. Section 4 discusses the different types of malware that affects the Android platform. In Sect. 5, we describe the implementation of our proposed malware detection system and have then presented the results of our experiment in Sect. 6. Finally, Sect. 7 describes the conclusion and future work.

2 Android Applications

Android applications can be written in Java, Kotlin, or C++. An application developed using Kotlin is executed by Java Virtual Machine (JVM) that was developed by JetBrains and licensed under Apache 2.0 whereas C++ can be integrated as native code. An android application can be published in Google play or any other third-party stores once it's developed. Figure 1 depicts the core components of an android application.

Each component has its functionalities and life cycles. The activity component interacts with users through the user interface (UI). This component is the entry point

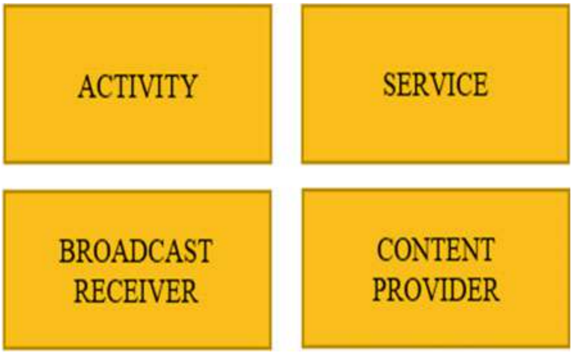


Fig. 1 Core components of an android application

for applications. The service component is concerned about running particular tasks in the background. These tasks do not require any interface and can run for a long time. They do not block any UI threads while waiting for network data or while processing some data. The broadcast receiver component is responsible for communication among android applications to send signals regarding status messages. The data in the application are handled by content providers. They allow read or write operations through certain APIs. There is no single point entry like a main method in Android applications. They use intents to facilitate inter-process communication between components or other applications.

Android package (apk) file is used to install an android application onto a smartphone. This apk file is disassembled to gain access to internal folders present in an application. The features present in AndroidManifest.xml and classes.dex files are used for static-based android malware detection. AndroidManifest.xml is an XML file that contains important data such as the name of the application, its version, and other data relevant to the application. It specifies the minimum API level of the host android system which would be favorable to the application. Features like content providers, services, activities, broadcast receivers, and permissions required by an application can be found in the AndroidManifest.xml file. The classes.dex file holds the android application's bytecode. Information such as restricted API calls, used permissions, suspicious API calls, and network addresses can be obtained from an application's bytecode. In our paper, we have used requested permissions, filtered intents, hardware components, restricted API calls, suspicious API calls, and used permissions taken from the AndroidManifest.xml and classes.dex files.

3 Related Works

This work aims to distinguish benign applications from malicious ones. To demonstrate the achievement of the stated aim, we have identified and surveyed the work done on similar grounds where different detection mechanisms have been used to detect applications with malicious intents.

J. Wang et al. [2] have used permissions and API calls as features to distinguish between malware and benign applications. They used 2 different machine learning algorithms, RandomForest (RF) for feature selection, and the XGBoost algorithm for malware detection. The authors used the accuracy metric to evaluate the efficacy of their proposed method. Similarly, the authors of [3] have also used permissions and API calls for detecting malware applications using Support Vector Machine (SVM), Decision tree (DT), Naïve Bayes (NB), and RF. L. Cen et al. in [4] extracted android API calls from classes.dex files of the sample applications. They used regularized logistic regression machine learning model to train and classify the samples. Their model achieved a high F1 score of 0.95. Inter-component communication (ICC) is the property of Android that is used by applications to communicate with the android system and also with each other. The authors of [5] used the ICC patterns of benign and malware apps to bring out the distinction between the two. In [6], the authors proposed a system called SIGPID that detects malware based on the permissions required by android applications. They identified 22 significant permissions that were found to have a major influence on differentiating between malware and benign applications.

In [7], G. Tao and et al. proposed an automated malware detection system that shortlisted 50 sensitive APIs containing information about the functionalities of an application and this feature set was fed to the RF algorithm for classification. F1 score metric was used to evaluate the effectiveness of the proposed system and the authors achieved a high score of 0.9824. The model proposed by W. Wang and et al. in [8] called DroidEnsemble considered the combination of static and structural features of an android application. The authors noted that they achieved better accuracy with static features alone compared to structural features. But by combining these two features, they achieved the highest accuracy of 0.98. HR in [9] made use of a deep learning model for detecting malware applications. To improve the accuracy of the proposed model, the author used 400 API permissions which included almost all the API permissions that an application can have. The work done in [10] utilized features taken from the manifest and Classes.dex files for malware detection. These features were embedded into a feature vector space that was fed to a residual Long Short-term Memory (LSTM) model. Though their proposed method obtained a high accuracy of 99.32%, it did not perform well with respect to recall, precision, and F1 scores. The security researchers in [11] focused on identifying if the information extracted from the manifest file were sufficient to identify malicious applications. They extracted features from three categories, namely, requested permissions, used permissions, and filtered intents. It was observed that the combined dataset produced the highest accuracy than individual feature categories. Three different algorithms, namely, Linear subspace discriminant, cubic SVM, and weighted k-nearest neighbor (KNN) were used to evaluate the performance on each dataset respectively. The proposed model's effectiveness was also tested against Drebin and AMD datasets.

H. Zhang et al. in [12] proposed an android malware detection technique that mad use of correlation behavior among abstracted API calls of applications. The authors used the Nearest Neighbor, SVM, and RF machine learning algorithms for classification purposes. The arrangement proposed in [13] reverse-engineered the malware datasets depending on their families to obtain the permissions required by applications. To address the constraints developed during classifying the malware families, the authors of [14] proposed a system called Android Malware Capability Annotation (MCA) to handle multi-label categorization that is required to define the abilities of Android malware. Their model achieved an accuracy of 63%, 98%, and 100% in deducing different capabilities of zero-day malware, small-size malware, and known malware, respectively. The state-of-the-art model proposed by Arp, Daniel et al. in [15] called DREBIN is a lightweight system for detecting Android malware applications. The authors extracted a total of 11 features from each application. Features such as hardware component, requested permission, content providers, broadcast receivers, activities services, and filtered intents were obtained from the manifest file. The remaining features i.e. restricted API calls, suspicious API calls, used permissions, and URLs were extracted from the dex file. These features were fed into the SVM machine learning algorithm for classifying samples into malware or benign.

In our paper, we propose a malware detection method that employs six significant string features i.e. suspicious API calls, used permissions, requested permissions, filtered intents, hardware components, and restricted API calls that were extracted from the samples in the DREBIN dataset. Since our proposed method is static based, the feature extraction process is more efficient compared to the process in dynamic analysis. We made a comparative study of the effectiveness of the selected features using four different machine learning algorithm that includes Support Vector Classifier, Random Forest, XGBoost Classifier, and Ridge Classifier.

4 Types of Malware

A snippet of code or software programmed with an intent to destroy a network, or a device or the end-user in any way or form, can be classified as a 'Malware'. The aim of a malicious application depends on what type of malware it is. The DREBIN dataset used in our paper contains 5,560 malware samples that are instances of different types of malware such as [16]:

- Trojan: It performs malicious activities without the user's knowledge or consent by impersonating itself as a benign application. They either steal or leak sensitive or confidential information of the user. Short message service (SMS) trojan is one such example. It sends premium messages without the user's consent and hence costing financial loss to the user.
- Adware: This type of malware steals bookmarks, changes the default settings of search engines, and pushes pointless alerts onto the user's device hence hindering its effectiveness.
- Backdoor: This malicious program is installed in the targeted device unbeknownst to the user, through which attackers can bypass the device's security protocols and gain access to the data inside the targeted device or control the device itself.

- Botnet: Botnets are a network functioning malware which has characteristics of a ‘Backdoor’, as they allow the attacker to gain access to the device. The infected devices would be under the control of the attacker and would await for commands from the Command and Control (C&C) server to execute harmful programs.
- Downloader: Once the downloader is installed onto the target device, it can be used by the attackers to facilitate further attacks by downloading other malicious programs onto the device.
- Kernel-Kit: This type of malware is designed to conceal other malicious programs. A typical kernel suit is often a combination of multiple malwares, for example, backdoor, downloader, etc. These kernel toolkits help the attacker to access the device remotely while itself being undiscoverable to the user.
- Launcher: The launcher is a malware that configures the programs or other similar malware for immediate or future covert operations. The main goal for a launcher is to install programs that allow it to obscure the malicious behavior from the victim. This generally contains a group of malicious programs pre-conditioned to launch other malware.
- Spyware: Spyware is a malware used for industrial espionage. The objective of this malware is to obtain proprietary, confidential, or personal information from the targeted device and pass the information from the compromised device to the attacker’s device without the permission of the user.
- Ransomware: This malware is indirectly installed on the victim’s device. This type of malicious software disables the accessibility of the device or data from the user, by encrypting the files. Further, the attackers extort the victims through intimidation until the ransom is met.
- Worm: Worm is a category of malicious programs with the likeliness of a computer virus. They are capable of self-replication, however, unlike a computer virus, a worm does not have a dependence on any host programs, on that terms they are independent in their execution strategy and infect other devices without the need to attach themselves to a host.

5 Methodology

We propose a malware detection method to assist android market stores in identifying malware applications such that it can be proactively removed before it is published at any of these markets. The method used in our paper is outlined as shown in Fig. 2:

Static analysis: The DREBIN dataset used for this experiment had used static analysis to obtain the features from AndroidManifest.xml and classess.dex files.

Generating feature vector: The features extracted from the DREBIN dataset are linked together to form a binary feature vector that is fed to the classifiers. The machine learning algorithms used in this paper are RandomForest (RF) classifier, Support vector classifier (SVC), Extreme gradient boost classifier (XGB), and Ridge classifier (RC). The RF algorithm generates decision trees based on each sample and obtains the prediction output from every decision tree.

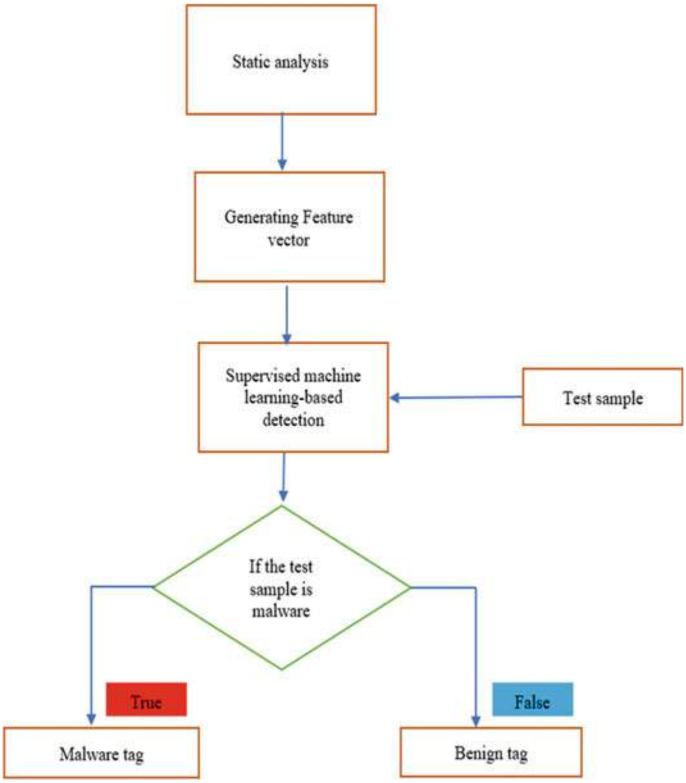
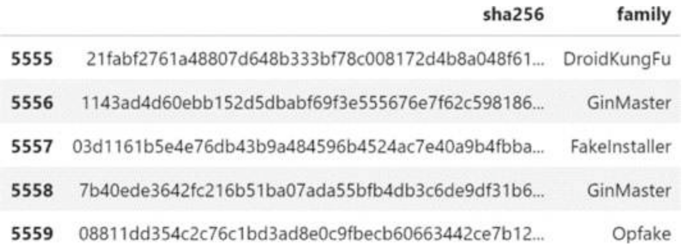

Static analysis
Generating Feature vector
Supervised machine learning-based detection
Test sample
If the test sample is malware
True
False
Malware tag
Benign tag

Fig. 2 Proposed Detection Method

The final prediction result is obtained using the voting method. Support Vector Classifier establishes a hyperplane to separate instances of different classes in a multidimensional space. Ridge Classifier transforms the target values to {−1,1} and then evaluates the problem as a regression task. XGBoost classifier is implemented by constructing multiple decision trees. The objective of XGB is to accurately calculate the predictions of decision tree classifiers iteratively.

Supervised machine learning-based detection: The classifiers are trained to detect malware by learning to distinguish between malware and benign features. If the trained model detects a test sample as malware then it is tagged as ‘1’, else it is tagged as ‘0’ indicating that the test sample is benign.



	sha256	family
5555	21fabf2761a48807d648b333bf78c008172d4b8a048f61...	DroidKungFu
5556	1143ad4d60ebb152d5dbabf69f3e555676e7f62c598186...	GinMaster
5557	03d1161b5e4e76db43b9a484596b4524ac7e40a9b4fbba...	FakeInstaller
5558	7b40ede3642fc216b51ba07ada55bfb4db3c6de9df31b6...	GinMaster
5559	08811dd354c2c76c1bd3ad8e0c9fbecb60663442ce7b12...	Opfake

Fig. 3 Apks along with its Malware Families

6 Results and Discussions

This section evaluates the effectiveness of our strategy to combine 6 string features namely, requested permissions, hardware components, used permissions, filtered intents, suspicious and restricted API calls for detecting malware applications.

6.1 Dataset

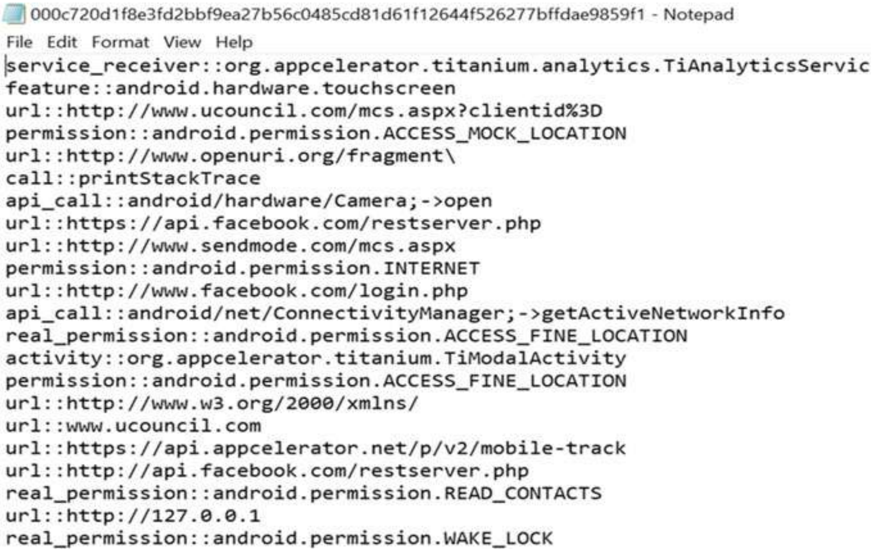
The dataset considered for this project is Drebin [11]. The whole dataset contains 5,560 malicious apps and 1,23,453 lakhs benign apps. For our experiment, we have included the entire 5,560 malware samples but only 10,000 benign samples. The malware samples are obtained from 179 different malware families such as Plankton, DroidKungFu, GinMaster, etc. Figure 3 shows a few malware samples and the malware family it belongs to.

6.2 Feature Selection

Each application's features have been extracted and imported to a text file illustrated in Fig. 4. Drebin has gathered these features from the manifest and classes.dex files. Requested permissions, services, hardware components, activities, broadcast receivers, content providers, and filtered intents were extracted from the AndroidManifest.xml file. Restricted API calls, suspicious API calls, and URLs were extracted from the classes.dex file.

Out of these 11 features, we have used the below 6 features to form the feature vector.

Hardware Components. An application's requests to access hardware components like camera, touchscreen, GPS, etc. are obtained from the manifest file and are stored as 'feature' variable in the Drebin dataset. If access has been requested


000c720d1f8e3fd2bbf9ea27b56c0485cd81d61f12644f526277bffdae9859f1 - Notepad
File Edit Format View Help

```
service_receiver::org.appcelerator.titanium.analytics.TiAnalyticsServic
feature::android.hardware.touchscreen
url::http://www.ucouncil.com/mcs.aspx?clientid%3D
permission::android.permission.ACCESS_MOCK_LOCATION
url::http://www.openuri.org/fragment\
call::printStackTrace
api_call::android/hardware/Camera;->open
url::https://api.facebook.com/restserver.php
url::http://www.sendmode.com/mcs.aspx
permission::android.permission.INTERNET
url::http://www.facebook.com/login.php
api_call::android/net/ConnectivityManager;->getActiveNetworkInfo
real_permission::android.permission.ACCESS_FINE_LOCATION
activity::org.appcelerator.titanium.TiModalActivity
permission::android.permission.ACCESS_FINE_LOCATION
url::http://www.w3.org/2000/xmlns/
url::www.ucouncil.com
url::https://api.appcelerator.net/p/v2/mobile-track
url::http://api.facebook.com/restserver.php
real_permission::android.permission.READ_CONTACTS
url::http://127.0.0.1
real_permission::android.permission.WAKE_LOCK
```

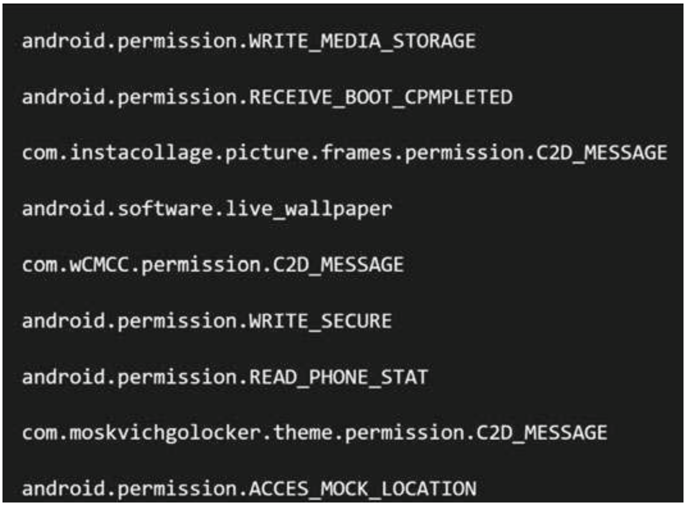
Fig. 4 Sample Text File

for hardware or a combination of hardware that could have some security implications, then it could imply that the application requesting these accesses could have malicious intents.

Requested Permissions. When an apk is installed in a mobile, it asks for permission to be granted by the user to access utilities such as contacts, wallpaper, location, etc. But few applications request permissions that might not be required for their functionality. Such applications with malicious intent usually incline to request a certain combination of permissions more recurrently than benign applications. The entire list of these permissions is cataloged in the manifest file. This list is extracted and stored in the variable called ‘permission’ in the Drebin dataset. Figure 5 shows a snapshot of permissions requested by a few malware samples in the dataset.

Filtered Intents. Intents are used for communication purposes or to request any action from other android components. Implicit intents are sent to other applications requesting certain details [9]. Intents are filtered by the application such that its components can reject unwanted communication or action requests and hence called as filtered intents. These intents are exported from the manifest file and into a variable called ‘intent’ in the DREBIN dataset.

Restricted API Calls. Critical API calls have restricted access as governed by the android permission system. If any of these API calls have to be invoked, then the permission to access it must be granted by the user. These API calls are analyzed by Drebin to have a better perception of the application’s functionality. If an application is accessing a restricted API call without requesting any permission to access it, then this implies that the application might have some malicious intent [8]. The restricted



```
android.permission.WRITE_MEDIA_STORAGE

android.permission.RECEIVE_BOOT_CPMPLETED

com.instacollage.picture.frames.permission.C2D_MESSAGE

android.software.live_wallpaper

com.wCMCC.permission.C2D_MESSAGE

android.permission.WRITE_SECURE

android.permission.READ_PHONE_STAT

com.moskvichgolocker.theme.permission.C2D_MESSAGE

android.permission.ACCES_MOCK_LOCATION
```

Fig. 5 Requested Malware Permissions

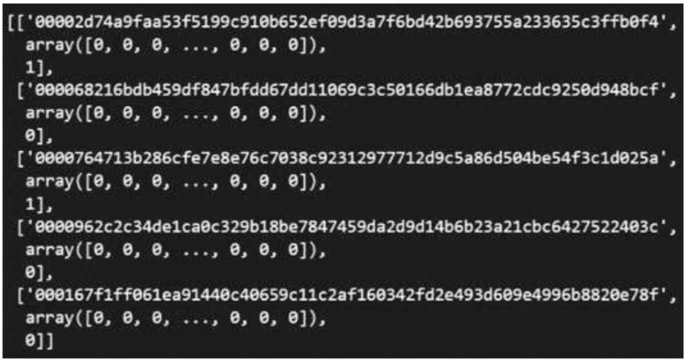
API calls are obtained from the classes.dex file and is stored in a variable called 'api_call' in the DREBIN dataset.

Used Permissions. It's not necessary that an app uses the resources it had previously requested access to. Therefore, the Drebin dataset contains a feature called 'real-permission' which comprises of permissions that the app uses [11].

Suspicious API Calls. Malicious applications use a collaboration of methods, services, and API calls that are not typical to benign applications. The API calls used by these apps usually access sensitive information of the user or the smartphone. Drebin dataset has collected API calls that access sensitive data or API calls that are used to communicate over a network or for sending or receiving messages or for executing external commands or used for obfuscation purposes, etc.

6.3 Feature Vector

A binary feature vector is created with all the values extracted from the 6 string features namely, suspicious API calls, used permissions, requested permissions, filtered intents, hardware components, and restricted API calls. Each apk Y is depicted as a binary vector Y = [y1, y2, y3, … , yn] where yi ∈ 0,1 specifying the existence (i.e. 1) or inexistence (i.e. 0) of a specific feature [13]. The target variable indicates whether the application is benign (0) or malicious (1). Figure 6 shows the binary vector of a few training samples in the dataset.



```
[['00002d74a9faa53f5199c910b652ef09d3a7f6bd42b693755a233635c3ffb0f4',
  array([0, 0, 0, ..., 0, 0, 0]),
  1],
 ['000068216bdb459df847bfdd67dd11069c3c50166db1ea8772cdc9250d948bcf',
  array([0, 0, 0, ..., 0, 0, 0]),
  0],
 ['0000764713b286cfe7e8e76c7038c92312977712d9c5a86d504be54f3c1d025a',
  array([0, 0, 0, ..., 0, 0, 0]),
  1],
 ['0000962c2c34de1ca0c329b18be7847459da2d9d14b6b23a21cbc6427522403c',
  array([0, 0, 0, ..., 0, 0, 0]),
  0],
 ['000167f1ff061ea91440c40659c11c2af160342fd2e493d609e4996b8820e78f',
  array([0, 0, 0, ..., 0, 0, 0]),
  0]]
```

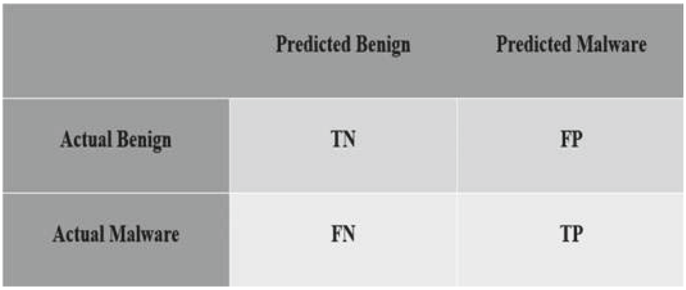
Fig. 6 Binary Vector of Training Samples

6.4 System Setup and Configuration

The demonstration was accomplished utilizing the Anaconda platform which contains a wide range of applications such as Rstudio, Spyder, Jupyter Notebook, etc. We have used Jupyter Notebook and python 3.8 version and a Windows 8 OS laptop powered by Intel® Core™ i5-8265U CPU @ 1.60 GHz and 8 GB of RAM for this experiment.

6.5 Evaluation Metrics

The confusion matrix (CM) or an error matrix as shown in Fig. 7 is specifically used in a classification problem of machine learning. It describes a classifier's performance on the testing data by summarizing its prediction results. Its main purpose is to provide a visualization of the algorithm's performance by providing an insight into



	Predicted Benign	Predicted Malware
Actual Benign	TN	FP
Actual Malware	FN	TP

Fig. 7 Confusion Matrix

Table 1 Confusion matrix variables

Algorithms	TN	FP	FN	TP
RF	1978	22	40	1072
SVC	1973	27	40	1072
XGB	1956	44	36	1076
RC	1941	59	113	999

the types of errors made by the algorithm. One can easily identify how many samples have been correctly or falsely classified.

Here, True Positive (TP) is when malware samples have been identified accurately. True Negative (TN) is when benign samples have been identified accurately. False Positive (FP) is when a given sample is benign but has been identified as malware. False Negative (FN) is when a given sample is malware but has been predicted as benign. Table 1 contains the TN, FP, FN, and TP values obtained in our experiment for RF, SVC, XGB, and RC classifiers.

The performance metrics used to evaluate the algorithms in this experiment are accuracy, recall, precision, F1-score, and AUC score. The ratio of the samples predicted correctly to the total predictions made is defined as accuracy. Recall is described as the ratio of the samples correctly classified as malware to the overall malware samples in the dataset. Precision is described as the ratio of samples correctly classified as malware to the overall predicted malware samples. F1 score is the harmonic mean of precision and recall. The area under the Receiver operating characteristic (ROC) curve (AUC) score is calculated by measuring the area covered by the curve plotted using 2 factors i.e. True Positive rate (TPR) and False Positive rate (FPR). TPR is nothing but the recall score whereas FPR is defined as the probability that a machine learning algorithm classifies the benign samples incorrectly. The result analysis shown in the next section clearly shows that all the algorithms that have been considered for this experiment have produced high precision and high recall which is an optimal score and implies that the number of false positives and false negatives are low.

7 Results

The extracted features present in the Drebin dataset are embedded into a binary vector. The binary vector for each application is fed into the classifiers along with the target variable that indicates whether the application is malware or benign. The effectiveness of the trained models is evaluated based on the prediction results made against the test sample. The predicted result is either 0 or 1 indicating that the application is benign or malware respectively. The dataset is split in a stratified manner such that the training and the testing samples contain sufficient instances from both malware and benign classes. This is done to avoid overfitting of the training data. We have employed GridSearch to tune the significant hyperparameters of the machine learning algorithms used in this experiment. For SVC, we mainly focused on 'C', 'gamma', and 'kernel' parameters. We fine-tuned 'n_estimators' and 'max_features' for the RF algorithm. For XGB, we considered 'n_estimators' and 'learning_rate' and for RC, we only focused on tuning the 'alpha' hyperparameter. DREBIN [11] had used a total of 11 features and achieved an accuracy of 94% using SVM whereas, in our method, we have used only salient feature categories and have achieved a better accuracy of 98%. The performance metrics considered to evaluate these algorithms are the weighted average of precision, recall, F1 score, accuracy, and AUC. In Table 2, it can be observed that RF has the best overall performance, followed closely by SVC and XGB.

A ROC curve is particularly useful in determining the overall effectiveness of an algorithm since it plots TPR against FPR. The area under the ROC curve is directly proportional to the potency of an algorithm i.e. if an algorithm covers more area under the curve then it is more effective compared to other algorithms. The performance of all the algorithms with respect to the ROC curve can be seen in Fig. 8. They clearly show that RF, SVC, and XGB have performed similarly whereas RC covers less AUC compared to the rest. The AUC scores for RF, SVC and XGBoost classifiers are 0.98, 0.98, and 0.97. Since these scores are very similar, the lines for these classifiers has overlapped in the ROC curve.

Table 2 Algorithm Comparison

Algorithms	Accuracy	Precision	Recall	AUC	F1_score	Training time in seconds
RF	0.98	0.98	0.98	0.976	0.98	171.15
SVC	0.978	0.978	0.978	0.975	0.978	240.49
XGB	0.975	0.975	0.975	0.972	0.975	349.11
RC	0.944	0.944	0.944	0.934	0.944	1.56

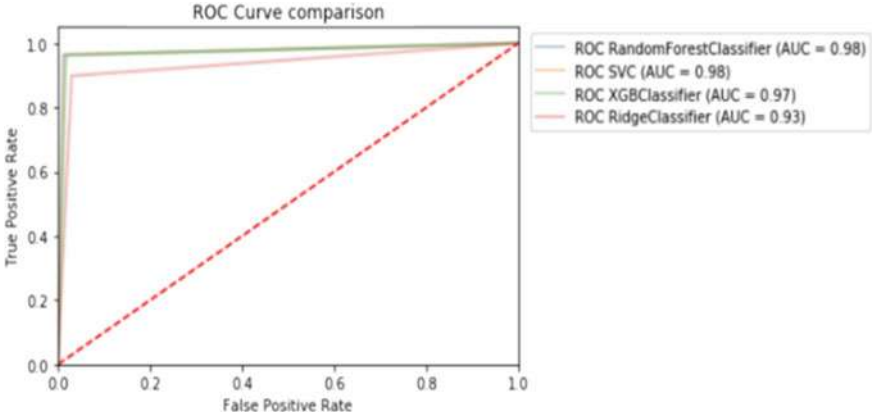


Fig. 8 Roc Curve Comparison

8 Conclusions

Scrutinizing and detecting malware android applications helps in proactively removing the malware apps before it is published in Google store or other third-party stores. The model proposed in this paper distinguishes the static behavior of malware and benign Android applications. We employ four machine learning algorithms, namely, RF, XGB, RC, and SVC for detecting malware. We have used 5,560 malware instances and 10 K benign instances from the DREBIN dataset to assess our model's efficacy. We noted that RF has achieved the highest score of 98% for accuracy, precision, recall, and F1 score. In the future, we aim to assess the effectiveness of our model using recent application samples and explore the usage of hybrid analysis in characterizing an application's behavior.

References

1. Damodaran, A., Di Troia, F., Visaggio, C.A., Austin, T., Stamp, M.: A comparison of static, dynamic, and hybrid analysis for malware detection. J. Comput. Virol. Hacking Tech. (2015). https://doi.org/10.1007/s11416-015-0261-z
2. Wang, J., Li, B., Zeng, Y.: XGBoost-based android malware detection. 13th International Conference on Computational Intelligence and Security (CIS), Hong Kong, pp. 268–272 (2017)
3. Koli, J.D.: RanDroid: Android malware detection using random machine learning classifiers. Technologies for Smart-City Energy Security and Power (ICSESP), Bhubaneswar, pp. 1–6 (2018). https://doi.org/10.1109/icsesp.2018.8376705.s
4. Cen, L., Gates, C.S., Si, L., Li, N.: A probabilistic discriminative model for android malware detection with decompiled source code. IEEE Trans. Dependable Secure Comput. **12**(4), 400–412 (2015). https://doi.org/10.1109/TDSC.2014.2355839
5. Xu, K., Li, Y., Deng, R.H.: ICCDetector: ICC-based malware detection on android. IEEE Trans. Inf. Forensics Secur. **11**(6), 1252–1264 (2016). https://doi.org/10.1109/TIFS.2016.2523912
6. Li, J., Sun, L., Yan, Q., Li, Z., Srisa-an, W., Ye, H.: Significant permission identification for machine-learning-based android malware detection. IEEE Trans. Industr. Inf. **14**(7), 3216–3225 (2018). https://doi.org/10.1109/TII.2017.2789219
7. Tao, G., Zheng, Z., Guo, Z., Lyu, M.R.: MalPat: Mining patterns of malicious and benign android apps via permission-related APIs. IEEE Trans. Reliab. **67**(1), 355–369 (2018). https://doi.org/10.1109/TR.2017.2778147
8. Wang, W., Gao, Z., Zhao, M., Li, Y., Liu, J., Zhang, X.: DroidEnsemble: Detecting android malicious applications with ensemble of string and structural static features. IEEE Access **6**, 31798–31807 (2018). https://doi.org/10.1109/ACCESS.2018.2835654
9. HR: Static analysis of android malware detection using deep learning. International Conference on Intelligent Computing and Control Systems (ICCS), Madurai, India, pp. 841–845 (2019) https://doi.org/10.1109/iccs45141.2019.9065765
10. Alotaibi, A.: Identifying malicious software using deep residual long-short term memory. IEEE Access **7**, 163128–163137 (2019). https://doi.org/10.1109/ACCESS.2019.2951751
11. Kumaran, M., Li, W.: Lightweight malware detection based on machine learning algorithms and the android manifest file. IEEE MIT Undergraduate Research Technology Conference (URTC), Cambridge, MA, pp. 1–3 (2016) https://doi.org/10.1109/urtc.2016.8284090
12. Zhang, H., Luo, S., Zhang, Y., Pan, L.: An efficient android malware detection system based on method-level behavioral semantic analysis. IEEE Access **7**, 69246–69256 (2019). https://doi.org/10.1109/ACCESS.2019.2919796

13. Alswaina, F., Elleithy, K.: Android malware permission-based multi-class classification using extremely randomized trees. IEEE Access **6**, 76217–76227 (2018). https://doi.org/10.1109/ACCESS.2018.2883975
14. Qiu, J., et al.: A3CM: Automatic capability annotation for android malware. IEEE Access **7**, 147156–147168 (2019). https://doi.org/10.1109/ACCESS.2019.2946392
15. Arp, D., Spreitzenbarth, M., Hübner, M., Gascon, H., Rieck, K.: DREBIN: Effective and explainable detection of android malware in your pocket. symposium on network and distributed system security (NDSS) (2014) https://doi.org/10.14722/ndss.2014.23247
16. Han, W., Xue, J., Wang, Y., Zhu, S., Kong, Z.: Review: Build a roadmap for stepping into the field of anti-malware research smoothly. IEEE Access **7**, 143573–143596 (2019). https://doi.org/10.1109/ACCESS.2019.2945787

Performance Evaluation of Cross-Layer Routing Metrics for Multi-radio Wireless Mesh Network

D. G. Narayan, Mouna Naravani, and Sumedha Shinde

Abstract Wireless Mesh Networks (WMNs) are emerging as future-generation technologies for back-haul connectivity of different types of networks. They are multi-hop networks consisting of mesh nodes, mesh clients and mesh routers. To achieve high performance, these networks use Multi-Channel Multiple Radio (MCMR) capabilities of mesh routers. However, QoS degrades due to the introduction of inter-flow interference and intra-flow interference by these nodes. Thus, there is a need to design a routing protocol that considers interference in the network. Furthermore, as the traditional approaches of protocol design lacks in improving QoS, the cross-layer metrics are designed by using the information from physical, MAC and network layer. In this paper, we investigated and analysed the performance of multi-radio cross-layer routing metrics. We performed extensive quantitative analysis in NS-2 using OLSR protocol. The performance differentials and trade-off analysis is carried out using five QoS parameters.

Keywords Multi-radio · Cross-layer · QoS · OLSR · P-IDA · CATT · iAWARE · MIND

D. G. Narayan (✉)
School of Computer Science and Engineering, KLE Technological University, Hubballi, Karnataka 580031, India
e-mail: narayan_dg@kletech.ac.in

M. Naravani
Department of Electrical and Electronics Engineering, KLE Technological University, Hubballi, Karnataka 580031, India
e-mail: mouna.naravani@kletech.ac.in

S. Shinde
Department of Mathematics, KLE Technological University, Hubballi, Karnataka 580031, India
e-mail: sumedha@kletech.ac.in

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_46

1 Introduction

Wireless mesh networks [1] are one of the emerging technology which has gained a lot of popularity and are in demand by making significant improvements in recent years. WMNs are becoming more popular with its minimum infrastructures and minimum investments abilities to provide all-over broadband internet access covering larger areas. WMNs consists of mesh gateways, routers and clients. End users use mesh clients like mobile phones, laptops, personal digital assistants, smart TVs, etc. Mesh routers acts a backbone which provides services to clients by forwarding client traffic details to gateways. These routers are static and equipped with multi-radio capabilities to deal with heavy traffic. Gateways acts as a bridge to connect other wired and wireless devices with the mesh networks.

WMNs uses routing protocols that consists of set of rules which plays the key role in the network performance. The main aim of routing protocols is to discover, establish and maintain routes for transmission. Until now, various routing protocols have been proposed in the literature. Routing metrics define the QoS performance of these routing protocols. To improve the performance in routing protocol there is a need for cross layer optimization in the design of routing metrics by using the information from physical layer, MAC layer and Network layer [2].

Design of new routing metric requires meticulous study, which involves motivation for new metric, their suitability to the types of WMNs, their merits and demerits. This study can be done using simulation or test bed. In this paper, we carry out through extensive simulation to study the performance of five cross layer routing metrics: P-IDA (Passive Interference and Delay Aware Routing Metric) [3], MIND (Metric for Interference and Channel Diversity) [4], CATT (Contention Aware Transmission Time) [5], iAWARE (Interference Aware Routing Metric) [6] and MIC (Metric of Interference and Channel Switching) [7]. We choose OLSR [8] routing protocol to implement these metrics. To evaluate the performance of these metrics the parameters assessed are: throughput, delay, routing overhead, route flaps and average path length. To carry out these performance evaluation NS2 simulator is used.

The paper is organised as follows. Section 2 discusses related work on performance evaluation of routing metrics. Section 3 refers to the design of evaluated routing metrics. Section 4 discusses the quantitative analysis of five popular routing metrics. Finally, in Sect. 5 we discuss the conclusion.

2 Related Work

Authors in [9] evaluate hop count and ETX routing metrics for community WMNs. They use 3 types of internet traffic for comparison of 3 QoS parameters. In [10], authors evaluate primary and composite routing metrics using RPL protocols. In [11], authors evaluate three single radio routing metrics in static wireless networks. Using OLSRV2 protocol, authors show the problem of routing metrics incoherence in heterogeneous networks. Similar study is carried out in [12] using RPL. In [13], authors evaluate passive and active metrics using iterative multipath protocol for wireless sensor networks. Authors conclude that passive metrics like MIND performs better than ETX based active metrics. However, all these studies are based on ETX and does not consider interference.

In [14], authors evaluate different multi-radio routing metrics with their proposed airtime link metric. Authors implement these metrics in hybrid wireless mesh protocol. The extensive simulation of voice over IP (VoIP) application was carried out by the authors of [15]. The study involved cross-layer routing metrics where the evaluation of these metrics were carried out with respect to user, performance of the network and stability of routes. Authors also discussed that metrics, stability and performance are co-related with each other to evaluate the network performance. However, some of the non-isotonic routing metrics are not considered for performance evaluation. There are lot of studies which have carried out qualitative performance comparison considering the characteristics of various multi-channel multi-radio metrics [2]. Performance evaluation of In-network caching is carried out in [16].

3 Cross-Layer Routing Metrics

This section discusses the routing metrics considered for the evaluation.

3.1 Metric of Interference and Channel Switching (MIC)

MIC depicts how load balancing can be achieved in WMNs to improve the network performance. It considers both inter-flow and intra-flow interference without creating uncertainty in the network. Thus, balancing both load and interference of the network. MIC is derived from ETT and is calculated as follows:

$$\mathrm{MIC}_p = \frac{1}{(N * \min(\mathrm{ETT}))} \sum_{\text{link } l \in p} \mathrm{IRU}_l + \sum_{i \in p} \mathrm{CSC}_i \tag{1}$$

and

$$\mathrm{IRU}_l = \mathrm{ETT}_l + N_l \tag{2}$$

IRU is the interference aware resource usage. Channel switching cost (CSC_i) estimates intra-flow interference as follows:

$$\mathrm{CSC}_i = \left\{ \frac{w1 \text{ if } \mathrm{CH}(\mathrm{prev}(i)) \neq \mathrm{CH}(i)}{w2 \text{ if } \mathrm{CH}(\mathrm{prev}(i)) = \mathrm{CH}(i)} \right. \tag{3}$$

$1 \leq w1 \leq w2$.

MIC prefers nodes with minimum number of neighbours so that traffic can be routed towards edges of the network. MIC becomes isotonic metric in order to consider intra-flow and inter-flow interferences in a better way when it is combined with virtual networks. MIC requires to have up to date information about ETT of each link which may lead to the overhead that reduces the performance of the network.

3.2 Interference Aware Routing Metric (iAWARE)

The first routing metric that took into account physical interference model (signal strength) is iAWARE. It computes optimal paths using inter and intra flow interferences. The metric is designed to suit the properties of infrastructure mesh networks. It can be calculated as follows

$$\mathrm{iAWARE} = (1-\alpha)\sum_{i=1}^{n}\mathrm{iAWARE}_i + \alpha \max_{1\le j\le k} X_j \tag{4}$$

where k is number of communication channels, n is number of connections, p is route of the network, iAWARE is the estimated interflow interference, α is used as adjusting factor between inter-flow and intra-flow interference.

Inter flow of a node is calculated as

$$\mathrm{iAWARE}_i = \mathrm{ETT}_i/\mathrm{IR}_i \tag{5}$$

IR_i is calculated as follows

$$\mathrm{IR}_i = \mathrm{SINR}_i/\mathrm{SNR}_i \tag{6}$$

where SINR_i and SNR_i is calculated as follows.

$$\mathrm{SINR}_i = \frac{P_i}{N} \tag{7}$$

$$\mathrm{SNR}_i = \frac{P_i}{N + \sum_{w\in N_{i-v}.\tau_w P_w}} \tag{8}$$

where P is signal strength, N is background noise, τ_w is amount of time for which node w makes channel busy. X_j is calculated as follows:

$$X_j = \sum_{\text{conflicting links } i \text{ on channel } j} \mathrm{iAWARE}_i \tag{9}$$

where $1 \le j \le k$.

3.3 Contention Aware Transmission Time (CATT)

An isotonic routing metric CATT, considers both contention and rate diversity. CATT selects a path which minimizes the time required for packet transmission. Thus taking into account both inter-flow and intra-flow interference. It avoids congested paths by using protocol interference model along with transmission time that captures interfering links between one and two hop neighbours, hence providing better performance. It is calculated as:

$$\mathrm{CATT}_i = \mathrm{ETX}_i * \sum_{j \in N_i} \left(\left(\sum_{k \in N_j} \frac{R_k}{L_k} \right) \right) \cdot \tau_j \cdot \frac{R_j}{L_j} \tag{10}$$

where N_i is the number of interfering connections with i, N_j interfering connections with transmission on the connection j. R_k and R_j are the packet sizes of 1-hop and 2-hop neighbours. The physical interference is not considered in this metric.

3.4 Metric for Interference and Channel Diversity (MIND)

MIND Routing metric uses passive approach to estimate link quality. It takes into account INTERLOAD which estimates load and inter-flow interference and CSC calculates the intra-flow interference. MIND is computed on each link as,

$$\mathrm{MIND} = \sum_{\mathrm{link}_{i \in p}}^{n} \mathrm{INTERLOAD}_i + \sum_{\mathrm{node}_{j \in p}}^{m} \mathrm{CSC}_j \tag{11}$$

3.5 Passive Interference and Delay Aware (P-IDA) Metric

In order to find least congested routes accurately, P-IDA estimates interference and delay of routes. Contention delay and transmission delay are equipped to compute delay using passive mechanism and to estimate interference using logical and physical interference model. P-IDA is computed as:

$$\begin{aligned} \text{P-IDA}(p) = \alpha & \sum_{\mathrm{link}\, i \in p}^{n} \mathrm{Delay}_i \cdot (1 - \mathrm{IR}_i) \\ & + (1 - \alpha) \sum_{\mathrm{link}\, j \in m}^{n} \mathrm{ICD}_j \end{aligned} \tag{12}$$

where n is the number of links. P-IDA estimates the important parameters like frame error rate based on SNR values, channel utilization, available bandwidth using passive mechanism. The metric is derived from 802.11 DCF model.

4 Performance Evaluation

An extensive performance evaluation of five multi-radio routing metrics is carried out in this section. It mainly consists of two parts namely simulation environment and result analysis of routing metrics.

4.1 *Simulation Model*

We discuss the simulation environment and performance evaluation parameters.

4.1.1 Simulation Environment

We use NS2 [17] for performance evaluation of routing metrics. We use the static nodes scenarios with OLSR protocol. We use 30 node topology with CBR traffic of packet size of 512 bytes. The other simulation parameters are presented in Table 1.

4.1.2 Performance Metrics

The following parameters are used to analyze the metrics.
Throughput: It is total packets received per second by the destination node.
Average End-to-End Delay: It is the difference in time between packets sent and received.
Routing overhead: It defines the total number of routing packets used in computing optimal route.
Route Flaps: The total number of route changes as part of routing interval.

Table 1 Simulation parameters

Parameters	Values
MAC protocol	802.11b
Data rate	11 Mbps
Routing protocol	OLSR
Number of flows	30–50 flows
Number of interfaces	2, 3, 4
Simulation time	100 s

4.2 Results and Discussion

We compare 5 metrics namely MIND, CATT, iAWARE, P-IDA and MIC in a heavily congested networks. We consider 30 node scenario where interfering flows are varied.

- Throughput and delay

Throughput of P-IDA routing metric is better than other metrics as shown in Fig. 1. P-IDA considers both logical and physical interference and hence chooses the least congested routes. MIND throughput is better than iAWARE and CATT as it computes the routes using load, logical and physical interference. iAWARE metric does not consider the logical interference thus the less performance.

Figure 2 shows that the average delay of P-IDA is less compared to all routing metrics as it estimate the delay accurately. iAWARE performs poorly as link computation does not consider backoff delay.

- Routing Overhead

Figure 3 shows that routing overhead of MIND and P-IDA are almost same as these metrics does not use probe packets. The probe packets in other three metrics adds a control overhead as they use ETX metric.

- Route Flaps

The route flaps contribute to route instability. As shown in Fig. 4, the route flaps for P-IDA and MIND is less compared to other routing metrics. This indicates that link costs are not changing frequently as due to small changes in passive parameters. The iAWARE and CATT metrics use probe packets and hence are very sensitive to traffic changes.

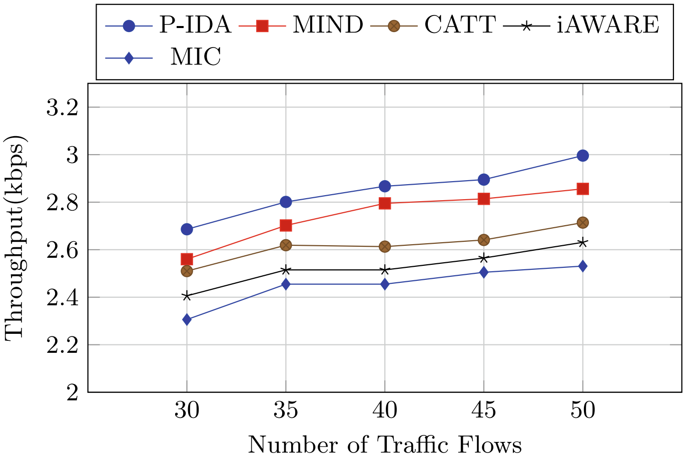


Fig. 1 Throughput versus number of interfering flows

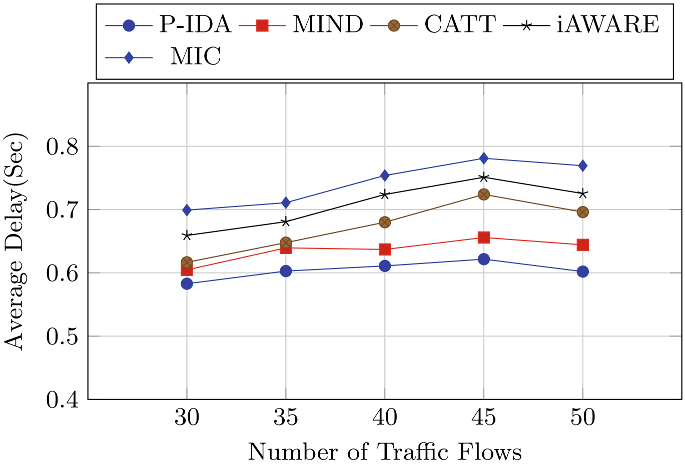


Fig. 2 Average end-to-end delay versus number of interfering flows

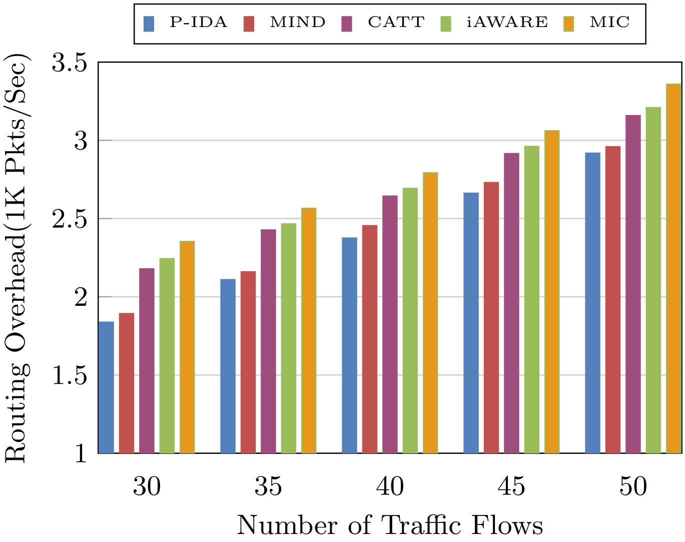


Fig. 3 Routing overhead versus number of interfering flows

- Effect of Multiple Interfaces

Multiple radios help in increasing the throughput of the networks. Figure 5 shows the throughput improvement of evaluated routing metrics with varying number of interfaces. The results reveal that P-IDA throughput is increase by 74.5% with Ni = 2.

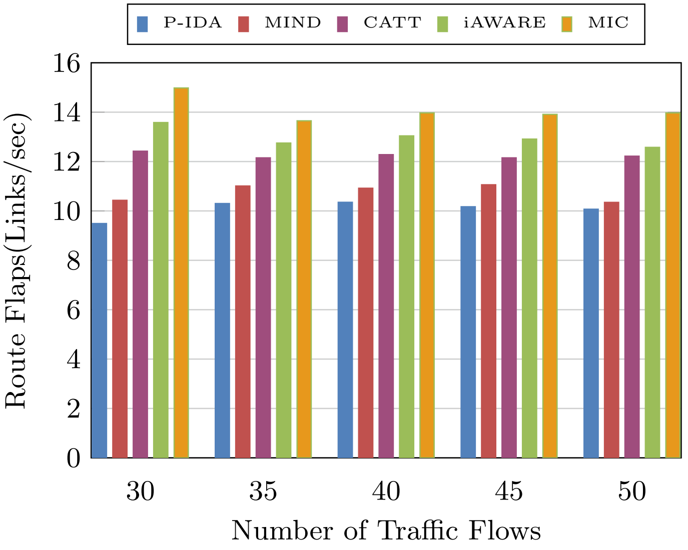


Fig. 4 Route flaps versus number of interfering flows

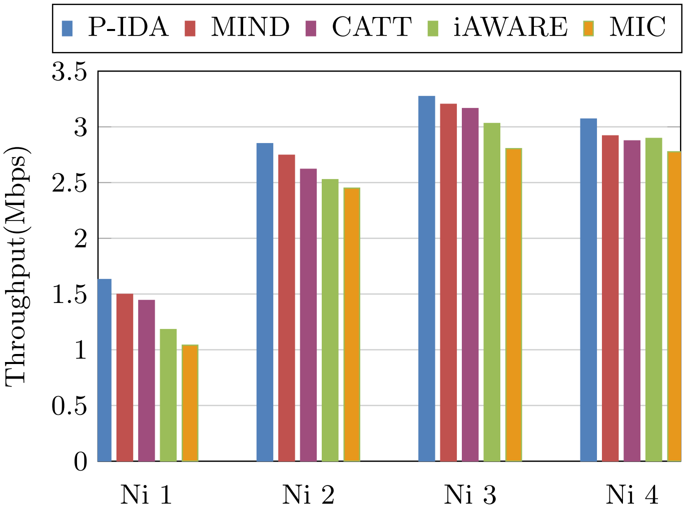


Fig. 5 Effect of multiple interfaces

Similarly, with Ni = 3, increase is 15%. However, with Ni = 4, the throughput is decreased by 6 %. The similar performance is observed with all other routing metrics.

The results clearly indicate that, with more number of interfaces, interference increases and hence throughput decreases.

5 Conclusion

This paper focuses on the routing metrics of multi-radio wireless network. The development of routing metrics has led to an increased difficulty in route computation by adopting new estimation techniques. Previous studies have considered few routing metrics for comparison focussing on single-radio. Our work carries out the performance evaluation of popular multi-radio metrics in NS-2.34 using OLSR. The result analysis reveal that P-IDA and MIND metrics performed better then other metrics. P-IDA is better than MIND due to accurate delay computation using Markov model. The analysis results also showed that the CATT, MIC and iAWARE performed the least.

References

1. Akyildiz, I.F., Wang, X.: A survey on wireless mesh networks. Commun. Mag. **43**(9), S23–S30 (2005)
2. Ngadi, M.A., Ali, S., Abdullah, A.H., Khokhar, R.H.: A taxonomy of cross layer routing metrics for wireless mesh networks. EURASIP J. Wirel. Commun. Netw. **2012**, 177 (2012)
3. Narayan, D.G., Mudenagudi, U.: A cross layer framework for joint routing and rate adaptation in multi-rate multi-radio infrastructure wireless mesh networks. Comput. Electr. Eng. **29**(4), 22–32 (2018)
4. Borges, V.C.M., Pereira, D., Curado, M., Monteiro, E.: Routing metric for interference and channel diversity in multi-radio wireless mesh networks. In: Ruiz, P.M., Garcia-Luna-Aceves, J.J. (eds.) Ad-Hoc, Mobile and Wireless Networks. ADHOC-NOW 2009. Lecture Notes in Computer Science, vol. 5793 (2009)
5. Genetzakis, M., Siris, V.A.: A contention-aware routing metric for multi-rate multi-radio mesh networks. In: Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON'08. 5th Annual IEEE Communications Society Conference on, pp. 242–250. IEEE (2008)
6. Subramanian, A., Buddhikot, M., Miller, S.: Interference aware routing in multiradio wireless mesh networks. In: 2nd IEEE Workshop on Wireless Mesh Networks, WiMesh 2006, pp. 55–63 (2006)
7. Yang, Y., Wang, J., Kravets, R.: Interference-aware load balancing for multihop wireless networks. http://hdl.handle.net/2142/10974
8. Clausen, T., Jacquet, P., Adjih, C., Laouiti, A., Minet, P., Muhlethaler, P., Qayyum, A., Viennot, L.: Optimized link state routing protocol (OLSR) (2003)
9. Liu, N., Seah, W.K.G.: Performance evaluation of routing metrics for community Wireless Mesh Networks. In: 2011 Seventh International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Adelaide, SA, 2011, pp. 556–561. https://doi.org/10.1109/ISSNIP.2011.6146613
10. Karkazis, P., Trakadas, P., Leligou, H.C. et al.: Evaluating routing metric composition approaches for QoS differentiation in low power and lossy networks. Wirel. Netw. **19**, 1269–1284 (2013). https://doi.org/10.1007/s11276-012-0532-2
11. Romanik, J., Bryś, R., Zubel, K.: Performance analysis of OLSRv2 with ETX, ETT and DAT metrics in static wireless networks. In: 2018 International Conference on Military Communications and Information Systems (ICMCIS), Warsaw, 2018, pp. 1–7. https://doi.org/10.1109/ICMCIS.2018.8398722
12. Thomson, C., Wadhaj, I., Romdhani, I., Al-Dubai, A.: Performance evaluation of RPL metrics in environments with strained transmission ranges. In: IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA). Agadir, 2016, pp. 1–8 (2016). https://doi.org/10.1109/AICCSA.2016.7945687

13. Maimour, M.: Impact of interference aware metrics on iterative multipath routing for industrial WSN. Internet Technol. Lett. **3**, e159 (2020). https://doi.org/10.1002/itl2.159
14. Ghannay, S., Gammar, S.M., Filali, F., Kamoun, F.: Multi-radio multi-channel routing metrics in IEEE 802.11s based wireless mesh networks. Ann. Telecommun. **67**(5–6), 215–226 (2012)
15. Borges, V.C., Curado, M., Monteiro, E.: Cross-layer routing metrics for mesh networks: Current status and research directions. Comput. Commun. **34**(6), 681–703 (2011)
16. Rajendra, H., Kammar, A., Budihal, S.: Performance evaluation of in-network caching: a core functionality of information centric networking. In: International Conference on Data Science and Communication, IconDSC 2019, pp. 1–8. https://doi.org/10.1109/IconDSC.2019.8817041
17. Network Simulator Ns-2. http://www.isi.edu/nsnam/ns

Modelling a Plain N-Hypercube Topology for Migration in Fog Computing

Pedro Juan Roig, Salvador Alcaraz, Katja Gilly, and Carlos Juiz

Abstract Fog Computing deployments need consolidated Data Center infrastructures in order to get optimal performances in those special environments. One of the key points in attaining such achievements may be the implementation of Data Center topologies with enhanced features for a relatively small number of users, although ready for dealing with occasional traffic peaks. In this paper, a plain N-Hypercube switching infrastructure is modelled in different ways, such as using arithmetic, logical and algebraic ways, focusing on its capabilities to manage VM migrations among hosts within such a topology.

Keywords Algebraic modelling · Fog computing · IoT · Live VM migration · N-hypercube

1 Introduction

Fog computing deployments are going to permit IoT devices to take advantage of remote computing services, as Cloud requirements do not properly fit into the needs of those devices, such as latency or bandwidth usage. Fog environments bring the computing assets to the edge of the network [1–13], thus improving the performance of such devices, whilst keeping the Cloud as a backup solution.

P. J. Roig · S. Alcaraz · K. Gilly
Miguel Hernández University, Elche, Spain
e-mail: salcaraz@umh.es

K. Gilly
e-mail: katya@umh.es

P. J. Roig (✉) · C. Juiz
University of the Balearic Islands, Palma de Mallorca, Spain
e-mail: proig@umh.es

C. Juiz
e-mail: cjuiz@uib.es

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0_47

Furthermore, Fog infrastructure may play a significative role in incorporating the moving IoT devices into the network, as they need to have their associated computing assets as close as possible, hence, as they may move around, so do their associated virtual machines (VMs) trying to follow them. In order to do so, a proper Data Center design is required, in a way that VM migrations take the shortest path to go from one host to another within the Fog domain [14–16].

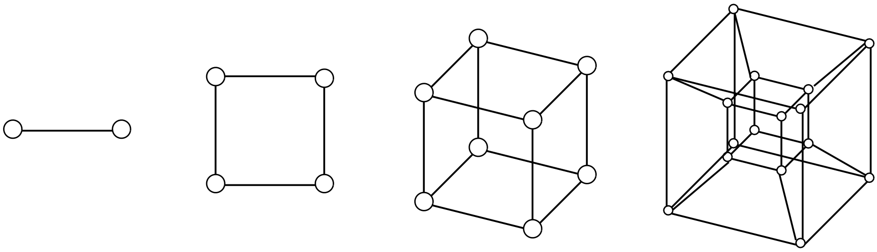
There are different types of topologies when it comes to designing Data Centers, but some of the most populars may be tree like layouts, such as Fat Tree or Leaf and Spine, or otherwise, graph like diagrams, such as plain or folded N-Hypercube. The aim of this paper is to present models for the VM migration process in a plain N-Hypercube topology using arithmetic, logical and algebraic ways, as this kind of topology may work properly in Fog environments [17–25].

The organization of this paper is as follows: first off, Sect. 2 introduces the plain N-Hypercube topology, next, Sect. 3 brings the model fundamentals, then, Sect. 4 shows the models for a generic host, afterwards, Sect. 5 presents the models for a generic switch, later on, Sect. 6 carries out a verification of the model, and finally, Sect. 7 draws some final conclusions.

2 Plain N-Hypercube

The N-Hypercube topology is a graph-like structure where the nodes are distributed according to the vertices of the shape of an N-Hypercube, those being connected by edges such that square faces are formed. N is the dimension of such a shape and it is the parameter imposing the number of nodes and its corresponding links among them, in a way that there must be 2^N nodes equally distributed, and $N \cdot 2^{N-1}$ links among those nodes. Furthermore, each node has N links connecting to other nodes [26, 27]. Figure 1 depicts the shapes of N-Hypercube for the first values of N.

Analyzing deeper the N-Hypercube structure as a generalization of a cube, some parameters may be deduced such that the number of j-dimensional faces (F_j) being found in a N-hypercube, where $j = 0$ gives the total number of vertices in the struc-



1-Hypercube 2-Hypercube 3-Hypercube 4-Hypercube

Fig. 1 N-Hypercubes of dimension N going from 1 to 4

Table 1 Number of j-dimensional faces and generalized Euler's characteristic

Figure	Number of j-dimensional faces (F_j)	Generalized Euler's characteristic for convex polyhedra
N-Hypercube	$\binom{N}{N-j} \cdot 2^{N-j}$	$\sum_{j=0}^{N-1} (-1)^j \cdot \binom{N}{N-j} \cdot 2^{N-j} = 1 - (-1)^N$

Table 2 Particular expressions to get the amount of relevant geometric places

Dimension	Vertices	Edges	Squares	Cubes	4-Hypercubes
N	$\binom{N}{0} \cdot 2^N$	$\binom{N}{1} \cdot 2^{N-1}$	$\binom{N}{2} \cdot 2^{N-2}$	$\binom{N}{3} \cdot 2^{N-3}$	$\binom{N}{4} \cdot 2^{N-4}$

ture, $j = 1$ states the number of edges, $j = 2$ accounts for the number of squares or $j = 3$ does it for the number of cubes. Furthermore, the generalizing Euler's characteristic may be obtained, whose original formulation stated that adding the number of faces and vertices and subtracting the number of edges in a polyhedron ($N = 3$), the result would always be 2 (Table 1).

From those general expressions, it is immediate to get the particular expressions for the number of switches and links among them, by just considering that the former are located in the vertices, as nodes are 0-dimensional, and the latter are located in the edges, as links are 1-dimensional (Table 2).

Regarding the overall number of hosts, it may depend on how many hosts are hanging on each of the switches, which will be located in each node. Moreover, the number of hosts will impose the number of links per switch, as each of them must have N links connecting to its neighboring switches, and additionally, the necessary links connecting to the corresponding hosts hanging on it.

The maximum distance between any two nodes is N, namely the dimension of the N-Hypercube, that being the case of opposite nodes. Therefore, there is just one node being at a distance N from a given node. It is to be said that taking one given node i, its opposite node may be calculated as its 1's complement in binary form, or otherwise, the result of the expression $2^N - 1 - i$ in decimal form.

Furthermore, a given node i is the only one at a distance 0 of itself, which is trivial. As per the rest of the nodes, they are at a distance from node i within a range between 0 and N, where the number of nodes being at a certain distance is given by the Nth row of the Pascal's triangle. Specifically, each column shows the number of nodes whose distance to node i is the same as the order of that kth column, starting from 0, and going sequentially from left to right. This may be seen in Fig. 2.

The main feature of Pascal's triangle is that a value located at any row N and at any column k is the addition of its two upper values, this is, the one at row $N - 1$ and column $k - 1$, along with the one at row $N - 1$ and the column k, whenever those addends exist. Alternatively, each value might be achieved by applying the binomial coefficient: $\binom{N}{k} = \frac{N!}{k! \cdot (N-k)!}$, where N and k have the same meaning as explained above:

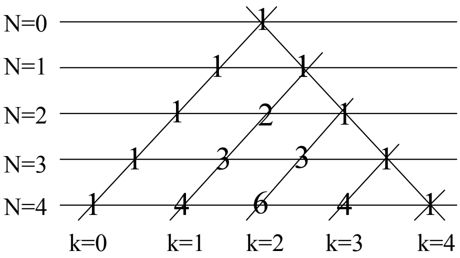


Fig. 2 Pascal's triangle for the first rows

$$W_{k=0}^{N}\binom{N}{k} = \frac{N!}{k!\cdot(N-k)!} = \binom{N}{0},\binom{N}{1},\binom{N}{2},\ldots,\binom{N}{N-1},\binom{N}{N}$$

Here they are the first values obtained with the aforesaid expression, which coincides with those attained with the Pascal's triangle.

- $N = 2 \rightarrow W_{k=0}^{2}\binom{2}{k} \rightarrow w_k = \binom{2}{0},\binom{2}{1},\binom{2}{2} \rightarrow w_k = 1, 2, 1$
- $N = 3 \rightarrow W_{k=0}^{3}\binom{3}{k} \rightarrow w_k = \binom{3}{0},\binom{3}{1},\binom{3}{2},\binom{3}{3} \rightarrow w_k = 1, 3, 3, 1$
- $N = 4x \rightarrow W_{k=0}^{4}\binom{4}{k} \rightarrow w_k = \binom{4}{0},\binom{4}{1},\binom{4}{2},\binom{4}{3},\binom{4}{4} \rightarrow w_k = 1, 4, 6, 4, 1$
- $N = 5 \rightarrow W_{k=0}^{5}\binom{5}{k} \rightarrow w_k = \binom{5}{0},\binom{5}{1},\binom{5}{2},\binom{5}{3},\binom{5}{4},\binom{5}{5} \rightarrow w_k = 1, 5, 10, 10, 5, 1$

As per the interpretation of those results, if $N = 2$, a given node i has 1 node at 0 hops away (just itself), 2 nodes at 1 hop away (its neighbors) and 1 node at 2 hops away (just the opposite node). Otherwise, if $N = 3$, i has 1 node at 0 hops away (itself), 3 nodes at 1 hop away (its neighbors), 3 nodes at 2 hops away and 1 hop at 3 hops away (the opposite node). And likewise with any other value of N.

On the other hand, it may be interesting to obtain expressions to calculate the average number of hops away for a given node i to reach the rest of the nodes within the infrastructure. Additionally, the average number of hops away between any pair of hosts may also be calculated, considering that all nodes have the same amount of hosts hanging on. In this case, that would be as easy as increment the former amount by 2 in order to account for the link from the source host to the source node, as well as the link from the destination host to the destination node.

For each case, either the average distance between any two switches or any two hosts, two different expressions are going to be quoted, one being the full expression (which is longer) and another one being a consolidated expression (which bears some simplifications), although both achieve the same results. The former expression gets the overall amount of hops multiplied by the number of switches being that far apart, divided by the total amount of nodes. The latter expression increases the hops considered by 2, to take into account the links going to both source and destination host (Table 3).

Table 3 Average distance among switches and among hosts

	Full expression	Consolidated expression
Average hops away between any two **nodes**	$\frac{\sum_{i=0}^{N}\binom{N}{i}\cdot i}{2^N}$	$\frac{N\cdot 2^{N-1}}{2^N} = \frac{N}{2}$
Average hops away between any two **hosts**	$\frac{\sum_{i=0}^{N}\binom{N}{i}\cdot(i+2)}{2^N}$	$\frac{(N+4)\cdot 2^{N-1}}{2^N} = \frac{N+4}{2} = \frac{N}{2} + 2$

3 Model

Nodes may be called by using N binary digits, each bit representing one of the dimensions. First of all, a node is selected as the initial one, where all of its N bits are set to 0, and from there on out, a 0 bit value means that a given node has the same value in such a dimension as that of the initial node, whilst a 1 bit means that the node in question has a different value in that dimension, as shown in Fig. 3.

Nodes may be also identified with decimal digits, just by translating each binary number, composed by N bits, into its decimal format, thus making those identifiers to be managed in an easier manner, as shown in Fig. 4.

On the other hand, the model proposed [28–35] is going to be composed by hosts h and switches i, those taking the place of each node within the N-Hypercube shape. As

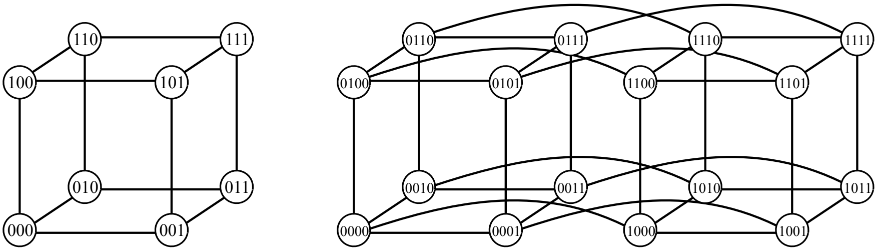


Fig. 3 Node nomenclature using N binary values: $N = 3$ (left); $N = 4$ (right)

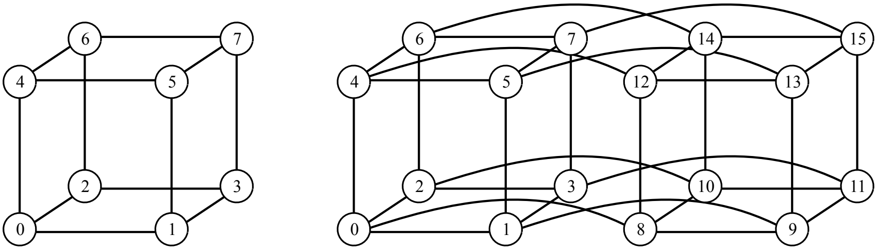


Fig. 4 Node nomenclature using decimal values: $N = 3$ (left); $N = 4$ (right)

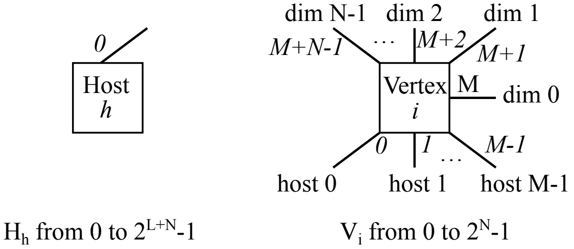


Fig. 5 Links on each host H_h and each vertex V_i within the N-Hypercube architecture

stated previously, the number of switches depends on N, according to the expression 2^N, whereas the number of hosts are going to be equally distributed along all switches, being up to M hosts connected per switch, ranging from $h_{i \cdot M}$ to $h_{(i+1) \cdot M-1}$, each one having a unique port 0. With respect to the nomenclature of the ports belonging to each node, the first M ports are going to be connected to its corresponding hosts, this is, ports from 0 to $M - 1$ will be dedicated to hosts, whereas the last N ports are going to be connected to its neighboring nodes, this is, ports from M to $M + N - 1$. Those entities and their links are exhibited in Fig. 5. Additionally, M may be a power of 2, such that $M = 2^L$, where the exponent L is the natural number which accounts for the logarithmic expression $\log_2 M$. Anyway, 2^L will be referred to in expressions, whereas M will be used in port identifiers.

4 Host

4.1 Preamble

For each host (2^L hosts per each of the 2^N switches, thus 2^{L+N}), check if the VM associated to any user u (given by the variable VM(u)) out of the total amount of users is hosted within (given by the variable *TotalUsers*). If that is not the case, then get ready to receive it at any time, and if that is it, get ready to send it over at any time whilst keeping that VM working.

Arithmetic and logical models are given by means of snippets of code. Focusing on the coding for host, it is the same for both approaches, and it is shown by using a C-style coding for clarity purposes, as exhibited in Algorithm 1, where a and b are source and destination hosts, respectively, whilst algebraic models are built up using Algebra of Communicating Processes (ACP), as seen in 1.

ACP recursive expressions use process terms, being defined by send and read as atomic actions carrying a generic message, with two arguments stating the device and its corresponding port. In case that the atomic action is send, it means that the VM associated to user u is sent by host h through its port 0, and likewise for the receive action. Furthermore, the relationships among those actions in the algebraic expressions are given by sequential operators, denoted by the multiplication sign, and addition operators, done by the addition sign [36–41].

As per the parameters of such actions, it may either be generically expressed by d, as in the algebraic expressions, or otherwise, described in a detailed manner, as in the arithmetic and logical expressions. In that case, three parameters are quoted, such as the initial point where the VM was located (source host), the final point where the VM is going to be located (destination host) and the identifier of the *VM* being moved. Therefore, atomic actions may be expressed as $s(d)$ and $r(d)$ in the former case and $s(x, y, \mathrm{VM}(u))$ and $r(x, y, \mathrm{VM}(u))$ in the latter case.

4.2 Coding Model

```
for (h = 0; h < $2^{L+N}$; h++) {
 while (1) {
  for (u = 0; u < TotalUsers; u++) {
   if NOT (VM(u) in h)
    $\text{receive}_{\text{HOST}\{h\},\text{PORT}\{0\}}$(a,h,VM(u));
   else
    $\text{send}_{\text{HOST}\{h\},\text{PORT}\{0\}}$(h,b,VM(u));
  }
 }
}
```

Algorithm 1 HOST().

4.3 ACP Model

$$H_h = \sum_{h=0}^{2^{L+N}-1} \left(s_{H_h,0}(d) + r_{H_h,0}(d) \right) \cdot H_h \tag{1}$$

5 Switch

5.1 Preamble

For each node whatsoever (2^N nodes), if a VM is received in any port, check whether that node i is the destination node, this is, the node where the destination host b is hanging on, which is given by the expression $\left\lfloor \frac{b}{2^L} \right\rfloor$, as there are 2^L hosts connected to each node. Therefore, this checking point is made by inspecting if $i = \left\lfloor \frac{b}{2^L} \right\rfloor$.

If this is the case, send the received VM over straight to destination host, or otherwise, search for the mismatching dimensions between that particular node and the destination node, and send the received VM over through all the corresponding ports in order to cover all possible redundant paths.

For this purpose, the values for each dimension j in both nodes are compared, although this may be done by identifying the nodes in its decimal or in its binary form. On the one hand, the arithmetic and algebraic models do it by applying the modulo 2 operator to the integer divisions of the aforesaid expressions for nodes in its decimal form by 2 to the power of each dimension j, where the complete expression is given by $\left\lfloor \frac{i}{2^j} \right\rfloor_{|2} \neq \left\lfloor \frac{\lfloor b/2^L \rfloor}{2^j} \right\rfloor_{|2}$. On the other hand, the logical model does it by applying the logical XOR operator to both nodes expressed in its binary form.

5.2 Arithmetic Model

Algorithm 2 presents the model for a generic node, where mismatching dimensions are found out by using *arithmetic* operations.

```
 for (i = 0; i < 2$^N$; i++) {
  while (1) {
   for (p = 0; p < M + N; p++) {
    if (receive$_{NODE\{i\},PORT\{p\}}$(a,h,VM(u))) {
     if (i == int (b / 2$^L$))
      send$_{NODE\{i\},PORT\{b\ mod\ M\}}$(a,h,VM(u));
     else
      for (j = 0; j < N; j++)
       if (((int (i / 2$^j$)) mod 2) != ((int (int (b / 2$^L$) / 2$^j$))
mod 2))
        send$_{NODE\{i\},PORT\{M+j\}}$(a,h,VM(u));
    }
   }
  }
 }
```

Algorithm 2 VERTEX_a().

5.3 Logical Model

Algorithm 3 exhibits the model for a generic node, where mismatching dimensions are found out by using *logical* operations.

```
for (i = 0; i < 2$^N$; i++) {
 while (1) {
  for (p = 0; p < M + N; p++) {
   if (receive$_{NODE\{i\},PORT\{p\}}$(a,h,VM(u))) {
    if (i == int (b / 2$^L$))
     send$_{NODE\{i\},PORT\{b\ mod\ M\}}$(a,h,VM(u));
    else {
     distance = 0;
     iNodeID = DecimalToBinary(i);
     dNodeID = DecimalToBinary(int(b / 2$^L$));
     logicalXOR = iNodeID XOR dNodeID;
     for (j = 0; j < N; j++)
      if (logicalXOR[j] == 0) {
       index[distance] = j;
       distance = distance + 1;
      }
     }
     for (x = 0; x < distance; x++)
      send$_{NODE\{i\},PORT\{M+index[x]\}}$(a,h,VM(u));
    }
   }
  }
 }
}
```

Algorithm 3 VERTEX_b().

5.4 ACP Model

Expression (2) is based on searching mismatching dimensions by means of *algebraic* operations. It is to be noted that atomic actions send ($s(d)$) and read ($r(d)$) bring the node (V_i) and the port identifiers as arguments. Furthermore, the conditions are represented as (*True* $\triangleleft$ *condition* $\triangleright$ *False*), the empty set ($\emptyset$) means nothing happens, $\lfloor x/y \rfloor$ stands for the integer division, whereas $x_{|2}$ and $x_{|M}$ does for the modulo operator [42, 43].

$$V_i = \sum_{i=0}^{2^N-1} \left(\sum_{p=0}^{M+N-1} \left(r_{V_i,p}(d) \cdot \left(s_{V_i,b_{|M}}(d) \triangleleft i = \left\lfloor \frac{b}{2^L} \right\rfloor \triangleright \left(\sum_{j=0}^{N-1} s_{V_i,M+j}(d) \triangleleft \left\lfloor \frac{i}{2^j} \right\rfloor_{|2} \neq \left\lfloor \frac{\lfloor b/2^L \rfloor}{2^j} \right\rfloor_{|2} \triangleright \emptyset \right) \right) \right) \right) \cdot V_i \qquad (2)$$

5.5 Redundant Paths

Algorithm 4 obtains the list of movements available to go from a source node to a destination node, where the source and destination hosts are hanging on.

The first point here is to distinguish whether the source node $\left\lfloor \frac{a}{2^L} \right\rfloor$ and the destination node $\left\lfloor \frac{b}{2^L} \right\rfloor$ are the same node, so it may forward the message straight to the destination host. Otherwise, a set of movements to reach the destination is sought after, no matter which permutation among them is selected, where each one involves its dimension and its direction, the former being a mismatched dimension between the present node and the destination node, and the latter being the way to get there.

```
Paths(a,b) {
 items[ ];
 items += [ $H_a V_{int(a/2^L)}$ ];
 t = 0;
 if (int (a / $2^L$) == int (b / $2^L$)) {
  items += [ Hb ];
  topology = "INTRASWITCH";
 }
 else {
  items += [ " ( PERMUTATIONS OF THESE MOVEMENTS: " ];
  for (j = 0; j < N; j++) {
   if (((int (int (a / $2^L$) / $2^j$)) mod 2) != ((int (int (b / $2^L$) / $2^j$)) mod 2)) {
    dimension[t] = j;
    exponent = ((int (int (a / $2^L$) / $2^j$)) mod 2);
    direction[t] = $(-1)^{exponent}$;
    movement[t] = direction[t]. $2^{dimension[t]}$;
    items += [ movement[t] ];
    items += [ " , " ];
    t = t + 1;
   }
  }
  items += [ " ) " ];
  items += [ $V_{int(b/2^L)} H_b$ ];
  topology = "INTERSWITCH";
 }
 print("Topology: %s\n",topology);
 print("Number of Hops: %d\n",t+1);
 if (t < 2)
  print(“Redundant Paths: 1\n”);
 else
  print("Redundant Paths: %d\n",(t)!);
 print("‘Items: %s\n", items);
}
```

Algorithm 4 Paths(a,b).

6 Verification

The plain N-Hypercube represents a N-dimensional geometric cube, where all nodes have N edges going to its neighboring nodes and all edges make part of that shape, permitting redundant paths to communicate any pair of nodes. Thus, a message going from a certain node always get to any other node, regardless how long it takes or the amount of edges it traverses, hence, by the application of the fairness condition, there will not be any message not reaching its expected destination if a sufficient amount of time is granted to make as many moves as necessary along this closed topology.

Additionally, considering the fact that ACP is an abstract algebra, an N-Hypercube topology may be seen as a bunch of interconnected internal nodes, with some external hosts getting connected to such an infrastructure, and this way, model verification may be easily achieved. Therefore, all nodes V_i may be presumed to be working concurrently, from $i = 0$ to $i = 2^{N-1}$, as their behavior is non deterministic. This fact may bring along a myriad of terms, depending on how many nodes are involved, although that number gets significantly reduced when applying the encapsulation operator (∂_H), which takes out all atomic actions in the nodes by getting them to deadlock, even though it allows communication among nodes. Subsequently, the process keeps on with application of the abstraction operator (τ_I) in order to mask internal communications and actions taking part internally, in a way that just external communications prevail.

At that moment, the external behavior of the model is shown, which may be compared with that of the real system, and if they both carry out the same string of actions and share the same branching structure, then it may be considered that they both are rooted branching bisimilar, which is a sufficient condition to verify a model.

- On the one hand, (3) shows the external behavior of the model:

$$\tau_I\left(\partial_H\left(||_{i=0}^{2^{N-1}} V_i\right)\right) = r_{H_h,0}(d) \cdot s_{H_h,0}(d) \cdot \tau_I\left(\partial_H\left(||_{i=0}^{2^{N-1}} V_i\right)\right) \tag{3}$$

- On the other hand, (4) shows the external behavior of the real system: A message departs from a given source host and arrives to a given destination host, after having traversed the N-Hypercube architecture:

$$X = r_{H_h,0}(d) \cdot s_{H_h,0}(d) \cdot X \tag{4}$$

Hence, it is evident that both recursive expressions are multiplied by the same factors, which make them both rooted branching bisimilar, as they have the same string of actions and they also share the same branching structure. Hence, (5) applies:

$$X \quad \longleftrightarrow \quad \tau_I\left(\partial_H\left(\|_{i=0}^{2^{N-1}} V_i\right)\right) \tag{5}$$

In conclusion, that is a sufficient condition to get a model verified, thus, the model presented is verified.

7 Conclusions

In this paper, there has been undertaken a study about modelling the VM migration process between any pair of given hosts being interconnected through a switching infrastructure whose topology has the shape of an N-Hypercube.

To start with, the most important features of this geometric figure have been introduced, and in turn, the basis of the model have been shown, such as the nomenclature used for the switches and the ports related to hosts and switches.

Afterwards, different models have being presented for both entities, such as the arithmetic ones, the logical ones and the algebraic ones. Additionally, an algorithm aimed at establishing the guidelines to get all redundant paths between any two given hosts has been proposed.

Finally, a model verification has been performed according to the algebraic model by comparing the external behavior of such a model with that of the real system, resulting in rooted branching bisimilar, with leads to the conclusion that the model gets verified.

References

1. Wang, S., Zhou, A., Komarov, M., Yau, S.: Services and communications in Fog computing. China Commun. Mag. **14**(1), iii–iv (2017). https://doi.org/10.1109/CC.2017.8233645
2. Yousefpour, A., Ishigaki, G., Jue, J.P.: Fog computing: towards minimizing delay in the internet of things. In: Narhstedt, K., Zhu, H. (eds.) 2017 IEEE 1st International Conference on Edge Computing, pp. 17–24. https://doi.org/10.1109/IEEE.EDGE.2017.12
3. Iorga, M. et al.: Fog Computing Conceptual Model—Recommendations of the National Institute of Standards and Technology. US Department of Commerce (2018). https://doi.org/10.6028/NIST.SP.500-325
4. Lee, H.G., Chang, L.: Powering the IoT: storage-less and converter-less energy harvesting. In: 20th Asia and South Pacific Design Automation Conference 2015, vol. 1, pp. 124–129. https://doi.org/10.1109/ASPDAC.2015.7058992
5. Stojmenovic, I., Wen, S.: The Fog Computing paradigm: scenarios and security issues. In: Ganzha, M., Maciaszek, L., Paprzycki (eds.) Proceeding of the 2014 Conference on Computer Science and Information Systems, ACSIS, vol. 2, pp. 1–8. Polish Information Processing Society, Warsaw (2014). https://doi.org/10.15439/2014F503
6. Luan, T.H., Gao, L., Li, Z., Xiang, Y., Sun, L.: Fog computing: focusing on mobile users at the edge. In: Computing Research Repository 2015. Cornell University, New York (2015). https://arxiv.org/abs/1502.01815

7. Khan, S., Parkinson, S., Qin, Y.: Fog computing security: a review of current applications and security solutions. J. Cloud Comput. Adv. Syst. Appl. **6**, 19 (2017). https://doi.org/10.1186/s13677-017-0090-3
8. Munir, A., Kansakar, P., Khan, S.U.: IFCIoT: Integrated Fog cloud IoT architectural paradigm for future internet of things. In: Computing Research Repository 2017. Cornell University, New York (2017). https://arxiv.org/abs/1701.08474v1
9. Stojmenovic, I.: Fog computing: a cloud to the ground support for smart things and machine-to-machine networks. In: Gregory, M. (ed.) Australasian Telecommunication Networks and Applications Conference 2014, pp. 117–122. ATNAC, Melbourne (2014). https://doi.org/10.1109/ATNAC.2014.7020884
10. Kamath, R.: Fog computing—a practical overview. Int. J. Innov. Res. Comput. Commun. Eng. **5**(4), 7819–7822 (2017). https://doi.org/10.15680/IJIRCCE.2017.05040234
11. Okano, M.T.: IOT and Industry 4.0: the industrial new revolution. In: International Conference on Management and Information Systems 2017, vol. 17, pp. 75–82. https://doi.org/10.1080/00207543.2018.1444806
12. Kumar, P.: Identities in the future internet of things. In: International Conference on Innovations in Electrical Engineering and Computational Technologies 2017, vol. 1, p. 1. https://doi.org/10.1109/ICIEECT.2017.7916594
13. Popentiu-Vladicescu, F., Albeanu, G.: Software reliability in the Fog computing. In: International Conference on Innovations in Electrical Engineering and Computational Technologies 2017, vol. 1, pp. 1–4. https://doi.org/10.1109/ICIEECT.2017.7916578
14. Kaur, P., Rani, A.: Virtual machine migration in cloud computing. Int. J. Grid Distrib. Comput. **8**(5), 337–342 (2015). https://doi.org/10.14257/ijgdc.2015.8.5.33
15. Filiposka, S., Mishev, A., Juiz, C.: Community-base VM placement framework. J. Supercomput. **71**(12), 4504–4528 (2015). https://doi.org/10.1007/s11227-015-1546-1
16. Osanaiye, O., Chen, S., Yan, Z., Lu, R., Choo, K.R., Dlodlo, M.: From cloud to Fog computing: a review and a conceptual live VM migration framework. In: Pecht, M. (ed.) IEEE Access 2017, vol. 5, pp. 8284–8300. https://doi.org/10.1109/ACCESS.2017.2692960
17. Adda, M., Peratikou, A.: Routing and fault tolerance in Z-Fat tree. IEEE Trans. Parallel Distrib. Syst. **28**(8), 2373–2386 (2017). https://doi.org/10.1109/TPDS.2017.2666807
18. Lebiednik, B., Mangal, A., Tiwari, N: A Survey and Evaluation of Data Center Network Topologies. Arxiv 2016. https://arxiv.org/abs/1605.01701
19. Okafor, K.C., Achumba, I.E., Chukwudebe, G.A., Ononiwu, G.C.: Leveraging Fog computing for scalable IoT datacenter using Spine-Leaf network topology. In: Senani, R. (ed.) J. Electr. Comput. Eng. **2017**, Article ID 2363240, 11 p. (2017). https://doi.org/10.1155/2017/2363240
20. Jyothi, S.A., Dong, M., Godfrey, P.B.: Towards a flexible data center fabric with source routing. In: Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research, Article No. 10. ACM, New York (2015). https://doi.org/10.1145/2774993.2775005
21. Alizadeh, M., Edsall, T.: On the data path performance of leaf-spine datacenter fabrics. In: Proceedings of the 2013 IEEE 21st Annual Symposium on High-Performance Interconnects, pp. 71–74. IEEE Conference Publishing Services, San Jose (2013). https://doi.org/10.1109/HOTI.2013.23
22. Al-Fares, M., Loukissas, A., Vahdat, A.: A scalable, commodity data center network architecture. ACM SIGCOMM Comput. Commun. Rev. **38**(4), 63–74 (2008). https://doi.org/10.1145/1402946.1402967
23. Guo, Z., Duan. J, Yang, Y.: Oversubscription bounded multicast scheduling in fat-tree data center networks. In: 2013 IEEE 27th International Symposium on Parallel and Distributed Processing (IPDPS), pp. 598–600. https://doi.org/10.1109/IPDPS.2013.30
24. Jain, N., Bhatele. A., Howell, L., Böhme, D., Karlin, I., Leon, E., Mubarak, M., Wolfe, N., Gamblin, T., Leiniger, M.: Predicting the performance impact of different fat-tree configurations. In: SC '17 Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, Article 50. ACM, New York (2014). https://doi.org/10.1145/3126908.3126967

25. Singla, A.: Fat-free topologies. In: 2016: Fifteenth ACM Workshop on Hot Topics in Networks (HotNets), pp. 1–7. ACM, New York (2016). https://doi.org/10.1145/3005745.3005747
26. Harary, F., Hayes, J.P., Wu, H.J.: A survey of the theory of hypercube graphs. Comput. Math. Appl. **15**(4), 277–289 (1988). https://doi.org/10.1016/0898-1221(88)90213-1
27. Disanto, F., Frosini, A., Rinaldi, S.: Square involutions. J. Integer Seq. **14**, Article 11.3.5, 1–15 (2011). https://cs.uwaterloo.ca/journals/JIS/VOL14/Rinaldi/square.html
28. Roig, P.J., Alcaraz, S., Gilly, K., Juiz, C.: Modelling VM migration in a Fog computing Environment. Elektron. Elektrotech. **25**(5), 75–81 (2019). https://doi.org/10.5755/j01.eie.25.5.24360
29. Roig, P.J., Alcaraz, S., Gilly, K., Juiz, C.: Modelling a leaf and spine topology for VM migration in Fog computing. In: Proceedings of the 24th International Conference Electronics 2020, Palanga (Lithuania), pp. 1–5. https://doi.org/10.1109/IEEECONF49502.2020.9141611
30. Roig, P.J., Alcaraz, S., Gilly, K., Filiposka, S., Aknin, N.: Formal Algebraic Specification of an IoT/Fog Data Centre for Fat Tree or Leaf and Spine architectures. In: Proceedings of the 2nd International Conference ICECCE 2020, Istanbul (Turkey), pp. 1–6. https://doi.org/10.1109/ICECCE49384.2020.9179445
31. Roig, P.J., Alcaraz, S., Gilly, K.: Arithmetic study on energy saving for some common data centre topologies. In: Proceedings of the 8th European Conference ECRES 2020, Istanbul (Turkey), pp. 1–7 (accepted by not yet published)
32. Mizuno, T., Shiratori, N., Higashino, T., Togashi, A.: Formal description techniques and protocol specification, testing and verification: FORTE X/PSTV XVII '97. In: IFIP Advances in Information and Communication Technology, New Generation Networks and Applications. Springer, Berlin, Heidelberg (1998). https://doi.org/10.1007/978-0-387-35271-8
33. Quemada, J.: Formal description techniques and software engineering: some reflections after 2 decades of research. In: de Frutos-Escrig, D., Núñez, M. (eds.) Formal Techniques for Networked and Distributed Systems. FORTE 2004. Lecture Notes in Computer Science, vol. 3235, pp. 33–42. Springer, Berlin, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30232-2
34. Vörös, A., Darvas, D., Hajdu, A., Klenik, A., Marussy, K., Molnár, V., Bartha, T., Majzik, I.: Industrial applications of the PetriDotNet modelling and analysis tool. In: Kordon, F., Moldt, D. (eds.) Special Issue from PETRI NET 2016. Science of Computer Programming, vol. 157, pp. 17–40. Elsevier, Amsterdam (2018). https://doi.org/10.1016/j.scico.2017.09.003
35. Chen, S., Fu, H, Miao, H.: Formal verification of security protocols using Spin. In: 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), pp. 1–6. Okayama, Japan (2016). https://doi.org/10.1109/ICIS.2016.7550830
36. Padua, D.: Encyclopedia of Parallel Computing. Springer, Heidelberg (2011). https://doi.org/10.1007/978-0-387-09766-4
37. Bergstra, J.A., Klop, J.W.: Algebra of communicating processes with abstraction. Theor. Comput. Sci. **37**, 77–121 (1985). https://doi.org/10.1016/0304-3975(85)90088-X
38. Groote, J.F., Mousavi, M.R.: Modeling and Analysis of Communicating Systems. The MIT Press, Cambridge (2014). https://dl.acm.org/citation.cfm?id=2628007
39. Fokkink. W.: Modelling Distributed Systems. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73938-8
40. Lockefeer, L., Williams, D.M., Fokkink, W.: Specification and verification of TCP extended with the Window Scale Option. In: Lang, F., Flammini, F. (eds.) Formal Methods for Industrial Critical Systems 2014. Science of Computer Programming, vol. 118, pp. 3–23. Elsevier, Amsterdam (2016). https://doi.org/10.1016/j.scico.2015.08.005
41. Fokkink, W.: Introduction to Process Algebra. Springer, Heidelberg (2000). https://doi.org/10.1007/978-3-662-04293-9
42. Neeman, A.: Algebraic and Analytic Geometry. Cambridge University Press (2007). https://doi.org/10.1017/CBO9780511800443
43. Caldwell, J.: Structural induction principles for functional programmers. In: Proceedings of Trends of Functional Programming in Education 2013, EPTCS 136, pp. 16–26. https://doi.org/10.4204/EPTCS.136.2

Author Index

S. M. Thampi et al. (eds.), *Advances in Computing and Network Communications*,
Lecture Notes in Electrical Engineering 736,
https://doi.org/10.1007/978-981-33-6987-0