Evaluation of Strategic Decision taken by Autonomous Agent using Explainable AI

Rendhir R. Prasad
Government Engineering College Barton Hill
Thiruvananthapuram, Kerala, India
rendhirrprasad@gmail.com

Rejimol Robinson R. R.
SCT College of Engineering
Thiruvananthapuram, Kerala, India
ashniya@gmail.com

Ciza Thomas
Directorate of Technical Education
Government of Kerala, India
cizathomas@gmail.com

N. Balakrishnan
Indian Institute of Science, Bangalore
Karnataka, India
balki@serc.iisc.in

2021 4th International Conference on Security and Privacy (ISEA-ISAP) | 978-1-6654-2017-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISEA-ISAP54304.2021.9689715

***Abstract*—Autonomous intrusion detection systems assess the data intelligently and take strategic decision to detect and mitigate cyber-attacks. These decisions have to be explained and evaluated for the transparency and correctness. Explainable Artificial Intelligent (XAI) methods that explore how features contribute or influence a decision taken using an algorithm can be useful for the purpose. XAI method of Testing with Concept Activation Vectors (TCAV) has been used recently to show the importance of high level concepts for a prediction class in order to deliver explanations in the way humans communicate with each other. This work explores the possibility of using TCAV to evaluate the strategic decision made by autonomous agents. A case study in the context of DoS attack is analysed to show that TCAV scores for various DoS attack classes and normal class of KDD99 data set can be used to evaluate the strategic decisions. The proposed method of analysis provides a quantifiable method to justify the current strategy or change in the strategy if required.**

***Index Terms*—Autonomous agents, Explainable AI, Evaluation of decision, TCAV, Information Security**

I. Introduction

An autonomous agent is perceived to be capable of taking independent decision by responding to the situation that it faces. In order to arrive at a decision about the response, the agent has to analyse the data. Both unsupervised and supervised machine learning techniques can be employed for analysing the data. The knowledge obtained from the analysis is utilised for assessing the situation for decision making. If the autonomous agent has to be a completely independent and self-evolving, then it should be driven by specific goals. The idea is apparent from the analysis of living beings in a biological ecosystem. The importance of having specific goals is that it gives a reference point for the evaluation of decision taken by the agent. So, it can be assumed that agent takes a decision that maximises the goal satisfaction. For biological beings, the fundamental goals are preservation of self and preservation of species. The motivation for the action of living beings can be traced to the satisfaction of these fundamental goals. In a similar manner, for artificial autonomous agents, assumption of being driven by specific goals helps to establish the motivation for the decision and will act as a tool to evaluate the decision. The autonomous agent that is considered for a case study in this work is an intelligent intrusion detection system that detects DoS cyber-attack and responds with a suitable mitigation strategy while it is deployed for protecting a server. Since the aim of the DoS attack is to adversely affect the availability of the server, we consider the goal of the autonomous agent as ensuring availability. Hence, the strategy suggested by the autonomous agent should be capable of preserving the availability of the server. For evaluation of the strategy, the criterion to be analysed is how far the suggested strategy helps to maximise the availability of the server.

Explainable Artificial Intelligence (XAI) explores the possibilities of explaining the black box reasoning of machine learning algorithms that are used in these autonomous agents. XAI explores how the input features contribute or influence the decision taken using a machine learning algorithm. In this paper, we propose a novel method based on Testing with Concept Activation Vectors (TCAV) [1] by exploring the relationship between strategy, goals, and the most influencing features, thereby addressing the following questions:

- How far the goals of the decision making agent are satisfied by the applying a particular strategy?
- How can a decision be justified in the case of change of strategy, if needed?

The research contributions of this work are:

- Used TCAV to explore the relationship between strategy, goals, concepts, and features.
- Proposed a novel method based on TCAV to evaluate the strategic decision of autonomous agent.
- Applied the proposed method to a cyber security scenario to analyse the choices of strategic decisions.

The paper is organised as follows. Section II explores the choices of XAI methods to be used in in goal based autonomous agents. In section III, TCAV method for measuring concept influence is explained. Section IV describes the case study scenario. In section V the approach to model the scenario is discussed. Section VI explains the details of the experimental evaluation undertaken in this work. In section VII the

experimental results are analysed. Section VIII discusses the evaluation of the strategic decision based on the experimental results followed by conclusion in section IX.

II. Explanation of Decision in goal based autonomous agent using XAI methods

The aim of the study is to propose a methodology to explain and evaluate the potential strategic decisions taken by autonomous agent in response to specific scenarios of cyber attacks. By strategic decision, we mean abstract representation of an action or a series of actions, which are possible for the agent to adopt for the mitigation of attacks. The aim of the agent is to satisfy the goal. Goal in this context mean the combination of high level goals of Confidentiality, Integrity and Availability (CIA). For example, for an agent deployed to mitigate DoS attack the goal is availability [2]. Hence, the strategy adopted by the autonomous agent should satisfy this goal. We need to analyse and answer the question of how far the goal is satisfied by the strategy. With a data set used for experimental evaluation, this analysis needs to be done in terms of features. Hence, our analysis technique should enable us to define the goals in terms of features, and provide a quantification for analysis. The XAI method using TCAV, which define high level concepts, is useful in this scenario.
TCAV method was proposed by Kim et al. [1] [3] and further investigated in [4], [5], [6], [7], [8], [9], [10], [11] and [12]. Continuing from the original example from Kim et al., TCAV was actively used to analyse image data for medical decision making [13] [14] [15] [16] [17] and also used in areas such as analysing bias in spoken languages [18], providing counterfactual explanation for trust enhancement [19] etc. We have addressed a problem in the field of cybersecurity using TCAV in order to utilise the potential of defining the goal as the concept, and to generate examples data points of the concept using features.

III. Measuring concept influence using TCAV

TCAV is an interpretability method that makes it easier to understand and comprehend the reason behind the predictions made by the neural networks models. Typical interpretability methods show importance weights in each input feature. TCAV instead shows importance of high-level concepts like the color or gender, for a prediction class. The explanations using TCAV are similiar to the normal humans communication, where prominence is given to the high-level concepts of a prediction class [1] [3].

A. Steps for implementing TCAV

1) Define a concept of interest: This is done by selecting a set of examples, which represent the concept, or by finding an independent data set that has the concept labelled.
2) Concept Activation vectors: By following the linear interpretability approach, a vector has to be sought in the space of activations of layer l that represents the concept of human interest with a set of examples that represents this concept. Such a vector is found by considering the activations in layer l that are produced by input examples in the concept set versus random examples. Concept Activation Vector (CAV) is defined as the normal to a hyperplane that separates examples without a concept and examples with a concept, in the model's activations. Let C be the concept of interest, P_C be the positive set of example inputs, and N be the negative set of example inputs. A binary linear classifier is trained to distinguish between the layer activations of the two sets $f_l(x) : x \in P_C$ and $f_l(x) : x \in N$. This classifier $V^lC \in \mathbb{R}^m$ is a linear CAV for the concept C. The sensitivity of ML predictions to changes in inputs towards the direction of a concept at neural activation layer l can be calculated using CAVs and directional derivatives. If $V^lC \in \mathbb{R}^m$ is a unit CAV vector for a concept C in layer l, and $f_l(x)$ is the activation for input x at layer l, the "conceptual sensitivity" of class k to concept C can be computed as the directional derivative $S_{C,k,l}(x)$: as proposed by Kim et al. [1]

$$\begin{aligned} S_{C,k,l}(x) &= \lim_{\epsilon \to 0} \frac{h_{l,k}(f_l(x) + \epsilon V_C^l) - h_{l,k}(f_l(x))}{\epsilon} \\ &= \nabla h_{l,k}(f_l(x) \cdot V^lC) \end{aligned} \tag{1}$$

where $h_{l,k} \in \mathbb{R}m \to \mathbb{R}$
3) Testing with CAVs (TCAV): TCAV make use of directional derivatives to compute the machine learning model's conceptual sensitivity across the entire classes of inputs. Let k be a class label for a given supervised learning task and let X_k denote all the inputs with that label. TCAV score [1] can be defined as

$$TCAV_{Q_{C,k,l}} = \frac{|\{x \in X_k : S_{C,k,l}(x) > 0\}|}{X_k} \tag{2}$$

This indicates the fraction of k-class inputs whose l-layer activation vector is positively influenced by the concept C, $TCAV_{Q_{C,k,l}} \in [0, 1]$. The decision taken by self-evolving autonomous system is explained by defining the goals as concepts, and analyzing how far the goal gets satisfied by target class based on which the strategy is set. The target classes can be normal or the various attack types. Relative influences of various goals can also be analyzed using TCAV. The analysis needs to use the relevant dataset to verify the method. KDD99 dataset is used for the analysis in this work.

IV. Case Study Scenario

The case study explained in this section is used to discuss the working of the proposed methodology.
The cyber-attack scenario considered for analysis is the Distributed Denial of Service (DDoS) attack. One of the methods used for this attack is TCP SYN flood attack in which premature termination of initiated three way handshake

for establishing TCP connection causes the wastage of resources that will lead to exhaustion of resource like CPU or bandwidth. Another method which is prevalent due to the growing popularity of IoT devices is HTTP flood attack. In this attack, the attacker exploit seemingly innocuous and legitimate HTTP GET or POST request to attack a web server. HTTP flood attack is usually carried out using botnets, which are the computers taken over by attacker, using malware, to simultaneously send large volume of HTTP requests to exhaust the server resources. In this attack the requests used for attack is very much similar to the genuine user requests. Normally, for any e-commerce business that provides service through Internet, getting large volume of user requests shows a good sign of popularity of any website and hence assessed as a success of the business model. Besides that, in order to boost the business, e-commerce websites announce special occasions of sales where they expect large volume of traffic. In these situations, they allocate more resources to manage the increased traffic. Attacker can exploit the situation by artificially creating traffic so that the website exhausts resources, and will be shut down, or necessitates the allocation of more resources, which results in financial burden. The scenario is as follows: A server is entrusted by a popular e-commerce website to deliver an important service to the user. The continuous availability of the server is very important to ensure profit for the company. If the server is down or not able to deliver the required service, the brand value of the company will be badly affected. Now, the server faces a situation of sudden surge of user requests. There can be two possible reasons for the sudden surge. It can be due to sudden increase in the popularity of website, which will be highly profitable for the company. It can also be due to a DoS attack, which will adversely affect the company.

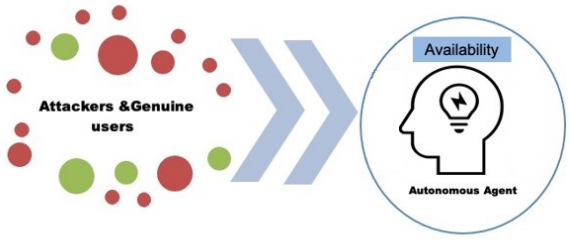


Figure 1: Visualisation scenario for case study

Figure 1 presents a visualisation for the case study scenario. Attackers and genuine users are requesting for service from the server secured using an autonomous agent. The autonomous agent under consideration is an intelligent intrusion detection system, which can detect the incoming attack with suitable mechanism and it responds with the appropriate mitigation strategy. The decision taking mechanism of the agent is analysed here. By analysing the scenario, it can be seen that availability can be considered as the goal of the agent. It is assumed that all the actions of the agents are driven by, and evaluated against, the goal. The autonomous agent has to preserve availability by ensuring the unobstructed access to the server.

While analysing this scenario from the perspective of an autonomous agent, the essential steps required for taking an action are identification of the type and nature of the network traffic, recognition of the effect of the particular traffic situation with a surge in network traffic, and deciding the appropriate strategic response.The potential strategies that can be taken in this case are as follows:

1) $TerminateConnection$ - Terminating the connection from the server.
2) $AllocateMoreResources$- Allocating more resource to keep server available.

The aim of the analysis is to evaluate these decisions. The goal of the autonomous agent is availability. The idea is to define availability as a concept and find the TCAV score of target classes, which are defined in KDD99 data set including:

1) Normal – class representing normal(benign) traffic
2) Smurf – DoS attack
3) Neptune -DoS attack
4) Land - DoS attack
5) Back - DoS attack
6) Teardrop - DoS attack
7) Pod - DoS attack

V. Modeling Approach

The approach for analysis of the scenario is explained by considering, say, $TerminateConnection$, as the decision taken by the autonomous agent. Through this section we try to explain how that decision is justified. We assume that the decision can be justified if the phenomenon of flash crowd is correctly presumed as a DoS attack. It means in TCAV terms, if we get significant TCAV score for DoS attack classes, we can assume that the traffic contains DoS attack. For analysis purpose, we consider what will happen to the traffic if we take the decision $TerminateConnection$. The resultant scenario can be described as follows:

1) Since the server terminates the connections, the traffic from the server will be zero. In terms of features, the condition dst_bytes=0 will hold.
2) The incoming traffic to the server will still be there, in large numbers (since our case is that of a traffic burst). Hence, the feature $src\_bytes > 50000$, also holds (The number 50000 is chosen after manually analyzing the data set).
3) The goal of the autonomous agent being 'availability' will get adversely affected. The server is now 'non available'.

Now, we define the concept $Nonavailability\_dst\_src$ to name the condition $dst\_bytes = 0 | src\_bytes > 50000$. In fact, our concept is a condition of features itself as in the case of stripes for the class zebra [1]. We assign human

understandable names (like availability or non-availability) for that condition as the concepts. This helps to connect the strategy ($TerminateConnection$) to the features (dst_bytes and src_bytes) in a meaningful manner. It also allows us to connect goal of the autonomous agent to strategy and features. Hence, the relation of strategy $TerminateConnection$ to the features dst_bytes and src_bytes and the concept $Nonavailability\_dst\_src$ can be visualized in Figure 2.

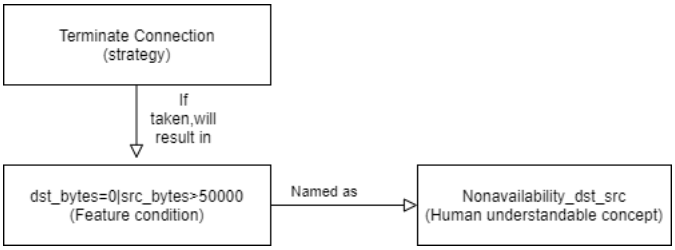


Figure 2: Relationship between strategy, features and concept

VI. Details of Experimental Evaluation

The KDD99 dataset is used for the experimental evaluation. The KDD cup 99 dataset collected by MIT Lincoln Laboratory under DARPA and AFRL sponsorship is the first standard dataset for the evaluation of Intrusion Detection systems. The attack types in the dataset are categorized into normal, R2L, U2R, Probe and DoS. We consider only DoS attacks in this work. This dataset is considered as a benchmark in the study of Intrusion Detection Systems. Though often criticised as an old dataset, which barely represents the current complex attacking scenarios, as demonstrated in [20], it is very much relevant for serving the comparison purpose.

The process of experimentation is detailed here with the example of the concept $Nonavailability\_dst\_src$ and the target class normal in the following steps.

1) We select rows from KDD99 dataset using the condition $dst\_bytes = 0|src\_bytes > 50000$ to generate the data samples for the concept $Nonavailability\_dst\_src$. There are a total of 4,94,021 data points in the data set (KDD99), We select 4,10,439 data points using the condition $dst\_bytes = 0|src\_bytes > 50000$ for the concept $Nonavailability\_dst\_src$.
2) We select samples of all target classes (normal and all six DoS attacks) using the feature 'labels'. The normal and all six DoS attack are selected separately. The number of data points are given in the Table I.
3) Random samples are generated from the full dataset. There are 10 data points in the random dataset. 10 such random data sets are used for analysis. Random datasets are randomly chosen from the data set, including all target classes.
4) We train the feed forward network for the target class. Here, we consider target class as normal for the explanation. Our concept of interest is $Nonavailability\_dst\_src$ and we generate the positive data points for the concept using the condition $dst\_bytes = 0|src\_bytes > 50000$. We also generate the negative set as the random sets. A binary linear classifier is trained to distinguish between the layers of activation of the two sets belonging to the positive and negative samples. This classifier is the linear CAV for the concept $Nonavailability\_dst\_src$. We calculate the directional derivative, which is the conceptual sensitivity of class normal to concept $Nonavailability\_dst\_src$ at layer say, $dense\_1$. Using the directional derivative, $TCAV_Q$ score, which is the fraction of normal-class inputs whose $dense\_1$ layer activation vector was positively influenced by concept $Nonavailability\_dst\_src$ can be computed.
5) Apply repeated statistical significance testing (p-value test) to check whether a valid CAV is learned or not. We repeat the testing 100 times for the calculation of each $TCAV_Q$ Score. The average time taken for calculating $TCAV_Q$ Score is found to be 0.0414 seconds. We calculate six $TCAV_Q$ Scores for each target class. Same data set is used for all the repeated trials.
6) Interpret the $TCAV_Q$ score. In order to interpret the TCAVQ score, we follow the methods used by Kim et al. [1].

Table I: Distribution of data points in KDD99. Total data points 4,94,021

Target Class	No. of data points
normal	97,278
smurf	2,80,790
neptune	1,07,201
back	2203
teardrop	979
pod	264
land	21

The process flow of the proposed methodology for evaluation of strategic decision is shown in figure 3.

VII. Result Analysis

The $TCAV_Q$ scores obtained for target classes are shown in the Figures 4a, 4b, 4c, 4d, 4e, 4f, and 5. Details of $TCAV_Q$ obtained for the middle bottleneck layer $dense\_1$ for both the strategies are shown in Tables II and III for both the strategies $TerminateConnection$ and $AllocateMoreResources$.

A. Interpretation of $TCAV_Q$ scores for the normal class

The general interpretation of the $TCAV_Q$ score for a target class w.r.t to a particular concept is that, it is the percentage of time classifier looked at the concept to make the decision. When analysing the case of the concept $Nonavailability\_dst\_src$ as is shown in the figure 5, CAV learned at the bottleneck layer $dense$ could not pass the p-test and hence was discarded. It is stated in [1] that $TCAV_Q$s in layers close to the logit layer represent more direct influence

Table II: $TCAV_Q$ Score of the target classes for the strategy $TerminateConnection$

Concept	Class	Bottleneck	p-value	$TCAV_Q$ Score
$Nonavailability\_dst\_src$	normal	dense_1	$0.015(significant)$	0.72
$Nonavailability\_dst\_src$	smurf	dense_1	$0.001(significant)$	0.97
$Nonavailability\_dst\_src$	land	dense_1	$0.001(significant)$	1.00
$Nonavailability\_dst\_src$	back	dense_1	$0.000(significant)$	0.97
$Nonavailability\_dst\_src$	neptune	dense_1	$0.049(significant)$	0.20
$Nonavailability\_dst\_src$	teardrop	dense_1	$0.025(significant)$	0.71
$Nonavailability\_dst\_src$	pod	dense_1	$0.002(significant)$	0.76

Table III: $TCAV_Q$ Score of the target classes for the5 strategy $AllocateMoreResources$

Concept	Class	Bottleneck	p-value	$TCAV_Q$ Score
$availability$	normal	dense_1	$0.015(significant)$	0.72
$availability$	smurf	dense_1	$0.001(significant)$	0.97
$availability$	land	dense_1	$0.006(significant)$	0.92
$availability$	back	dense_1	$0.042(significant)$	0.78
$availability$	neptune	dense_1	$0.229(notsignificant)$	0.31
$availability$	teardrop	dense_1	$0.077(notsignificant)$	0.66
$availability$	pod	dense_1	$0.107(notsignificant)$	0.63

on the prediction than lower layers. Following this argument, $dense\_2$ which is the highest layer is considered. So, we can argue that the concept $Nonavailability\_dst\_src$ (ie. the feature condition $dst\_bytes = 0|src\_bytes > 50000$) is highly relevant and influential in predicting the class normal because of a $TCAV_Q$ score of 1.00 for the layer $dense\_2$. We can extend the argument in our context as follows:
If we take the strategy $TerminateConnection$, the concept it results from, will be relevant for normal class. ie. classifier depends 100% of the time on this concept to reach the decision. This argument is supported by the ground truth because if we take data from dataset using the feature condition $dst\_bytes = 0|src\_bytes > 50000$, out of 410439 data points, 14211 are normal as shown in the Table IV. So, the argument is, if $TerminateConnection$ is taken as a strategy, there exists the risk of loosing genuine customers.
When analysing in the case of the concept $availability$, $TCAV_Q$ scores follows a similar pattern. $TCAV_Q$ score for the $dense\_2$ layer is high, and the normal is relevant for this concept also. When we argue in the context of the strategic decision $AllocateMoreResources$, this gives the confidence that allocating more resources will serve customers better.

B. Interpretation of $TCAV_Q$ scores for the DoS classes

Six DoS attack classes present in KDD99 dataset are used in this study. The $TCAV_Q$ score of smurf attack is shown in the Figure 4a. Considering both the concepts $Nonavailability\_dst\_src$ and $availability$, though significant $TCAV_Q$ score for the $dense\_2$ layer could not be obtained, concept $Nonavailability\_dst\_src$ has got a significant support for smurf attack ($TCAV_Q$ score of 0.97 for both the concepts). Similar to the case of normal class already discussed, it can be argued that the traffic has the presence of smurf attack in both cases. In the case of land attack shown in the Figure 4b, $TCAV_Q$ score of layer $dense\_2$ is insignificant in both concepts. $TCAV_Q$ scores of the layers $dense$ and $dense\_1$ shows significant scores. It can be argued that land attack is present in the traffic even when considering both concepts. When analysing $TCAV_Q$ scores of back attack 4c, it can be seen that significant $TCAV_Q$ scores are present for both the concepts, though the score of the concept $availability$ is lower than that of the concept $Nonavailability\_dst\_src$. Hence, it can be argued that presence of back cannot be ignored while taking the decision $AllocateMoreResources$. The $TCAV_Q$ score of the attack neptune shown in 4d shows a different pattern. The concept $Nonavailability\_dst\_src$ is having a positive but low support for neptune ($TCAV_Q$ score of 0.20) but significant $TCAV_Q$ score for the concept $availability$ could not be obtained. The $TCAV_Q$ scores of the attacks teardrop and pod shown in Figures 4e, 4f show almost similar pattern as neptune. This is because they also receive support from the concept $Nonavailability\_dst\_src$ and significant $TCAV_Q$ scores for the concept $availability$ could not be obtained.
The distribution of target classes in the concepts $Nonavailability\_dst\_src$ and $availability$ are shown in the Tables IV and V respectively. When analysing the obtained $TCAV_Q$ scores in general, it can be seen that $TCAV_Q$ scores agree with the ground truth except in the cases of classes smurf, land, and back for the concept $availability$. This can be due to model error as argued in [1]. In that case experts can use the analysis to fix the model.

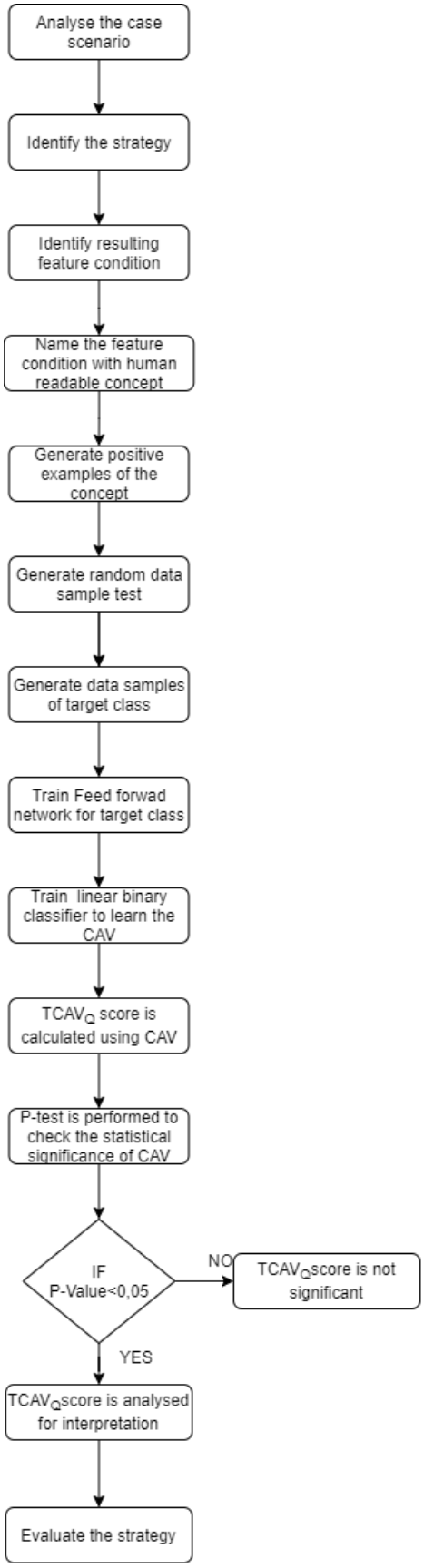


Figure 3: Process flow of the evaluation method

Table IV: Distribution of data points in the concept $Nonavailability\_dst\_src$.Total data points 4,10,439

Target Class	No. of data points
normal	14,211
smurf	2,80,790
neptune	1,07,201
back	2166
teardrop	977
pod	264
land	21

Table V: Distribution of data points in the concept $availability$. Total data points 85,763

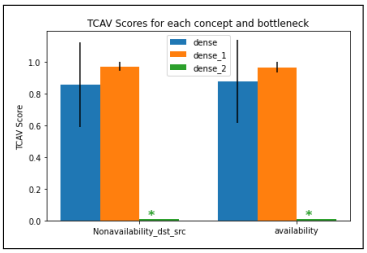
Target Class	No. of data points
normal	83,083
smurf	0
neptune	0
back	2202
teardrop	2
pod	0
land	0

VIII. Evaluation of strategic decision

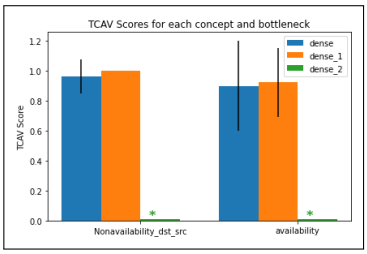
Evaluation of a strategy is essentially checking whether presumption on which the decision is taken is right or not. In the case of strategy $TerminateConnection$, the presumption is that the incoming traffic is a DoS attack. So, the decision to take that strategy is correct if traffic burst is a DoS attack. The presumption of the decision to take the strategy $AllocateMoreResources$ is that the incoming traffic is normal to flash crowd. In order to analyse the decision, we analyse the resultant scenarios of taking the decision. As explained in our method, we define concepts based on resultant scenarios and test with target classes using TCAV.

A. Strategy-$TerminateConnection$

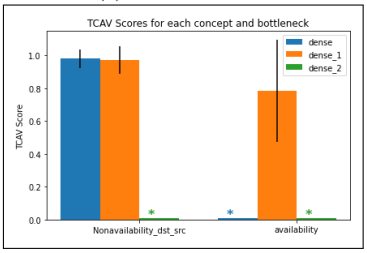
The concept $Nonavailability\_dst\_src$ is defined to represent those data instances that exemplify the strategy $TerminateConnection$. The aim of the analysis is to check how much the given concept is supported by various classes in terms of TCAV scores. For the concept $Nonavailability\_dst\_src$, the condition $dst\_bytes = 0|src\_bytes > 50000$ is used for generating the example data items, which represent the strategy $TerminateConnection$, and hence the concept $Nonavailability\_dst\_src$. The rationale for using this condition is that, when server terminates the connection, no data will be sent from server and hence the condition $dst\_bytes = 0$. Meanwhile traffic from the source will be large compared to the usual traffic and hence the condition $src\_bytes > 50000$.



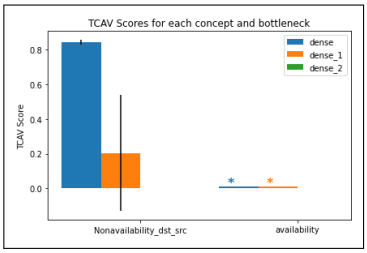
(a) Class = smurf



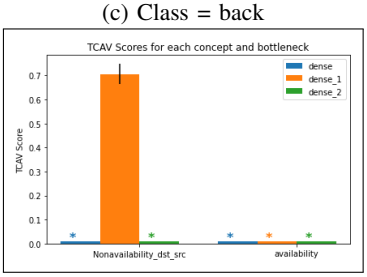
(b) Class = land



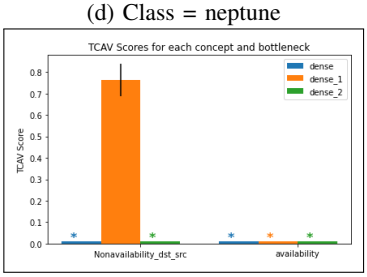
(c) Class = back



(d) Class = neptune



(e) Class = teardrop



(f) Class = pod

Figure 4: $TCAV_Q$ scores of DoS attacks for the bottleneck layers $dense$, $dense\_1$ and $dense\_2$ with the concepts $Nonavailability\_dst\_src$ and $availability$. The concepts represent strategies $TerminateConnection$ and $AllocateMoreResources$ respectively.

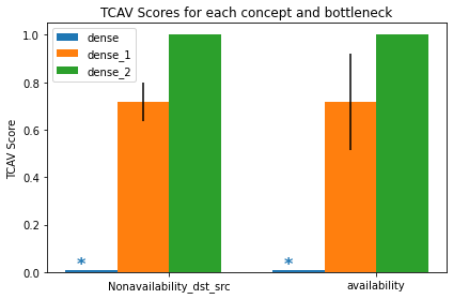


Figure 5: TCAV scores for normal with the concept $Nonavailability\_dst\_src$ and $availability$ for the strategies $TerminateConnection$ and $AllocateMoreResources$.

By interpretation of the $TCAV_Q$ scores, it can be argued that this strategy is moderately effective in the sense that, all six DoS attacks are giving positive TCAV score. However, normal class is also getting a positive TCAV score. So if server use $TerminateConnection$, the service provided to genuine users will be affected.

B. Strategy -$AllocateMoreResources$

The concept $availability$ is defined to represent those data instances which exemplify the strategy $AllocateMoreResources$. For the concept $availability$, the condition $dst\_bytes > 0$ is used. It is assumed that when server adopts the strategy $AllocateMoreResources$, the server remains available even by allocating more resources, if required. Hence, the condition $dst\_bytes > 0$ is used here to represent the strategy $AllocateMoreResources$.
The analysis based on $TCAV_Q$ scores shows that the strategy may not be effective because apart from normal, three DoS attacks such as smurf, land and back are also giving positive $TCAV_Q$ scores. So, the presumption of incoming traffic as benign, may not be correct. At the same time, the decision boundary is not completely clear as there exists the possibility of error in the model. There is also another possibility that instead of exercising $TerminateConnection$, server can $AllocateMoreResources$ first and change the decision later. In that case the decision changing point has to be identified. The further scope of this work is to investigate on these problems.

IX. Conclusion

The XAI method of TCAV is used to evaluate the decision taken by autonomous IDS. A novel method based on TCAV is used to define goal of the autonomous agent as concept. The method explores the relationship between the strategic decision, goal of the autonomous agent, and the features of the dataset. A case study of DoS attack is undertaken to show the implementation of the proposed method. $TCAV_Q$ scores are obtained for various DoS attacks and normal traffic in KDD99 dataset. The $TCAV_Q$ scores are used to establish the relationship between the goal $availability$ and the strategies $TerminateConnection$ and $AllocateMoreResources$. The analysis is done to justify the selection of the strategy or a change in the strategy if needed, in the event of cyber attacks.

References

[1] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, *et al.*, "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)," in *International conference on machine learning*, pp. 2668–2677, PMLR, 2018.

[2] S. R. Islam, W. Eberle, S. K. Ghafoor, A. Siraj, and M. Rogers, "Domain knowledge aided explainable artificial intelligence for intrusion detection and response," *arXiv preprint arXiv:1911.09853*, 2019.

[3] B. Kim, J. Gilmer, M. Wattenberg, and F. Viégas, "Tcav: Relative concept importance testing with linear concept activation vectors," 2018.

[4] S. Verma, C. Wang, L. Zhu, and W. Liu, "A compliance checking framework for dnn models," in *Twenty-Eighth International Joint Conference on Artificial Intelligence {IJCAI-19}*, International Joint Conferences on Artificial Intelligence Organization, 2019.

[5] A. Ghorbani, J. Wexler, J. Zou, and B. Kim, "Towards automatic concept-based explanations," *arXiv preprint arXiv:1902.03129*, 2019.

[6] A. Ghorbani, J. Wexler, and B. Kim, "Automating interpretability: Discovering and testing visual concepts learned by neural networks," *arXiv preprint arXiv:1902.03129*, 2019.

[7] R. Soni, N. Shah, C. T. Seng, and J. D. Moore, "Adversarial tcav–robust and effective interpretation of intermediate layers in neural networks," *arXiv preprint arXiv:2002.03549*, 2020.

[8] A. Janik, J. Dodd, G. Ifrim, K. Sankaran, and K. Curran, "Interpretability of a deep learning model in the application of cardiac mri segmentation with an acdc challenge dataset," in *Medical Imaging 2021: Image Processing*, vol. 11596, p. 1159636, International Society for Optics and Photonics, 2021.

[9] J. Pfau, A. T. Young, J. Wei, M. L. Wei, and M. J. Keiser, "Robust semantic interpretability: Revisiting concept activation vectors," *arXiv preprint arXiv:2104.02768*, 2021.

[10] H. Fischel, "Relevance-tcav: Explaining deep neural nets in human concepts," 2021.

[11] W. Coenraads, *Infusing Causal Knowledge Into Deep Neural Networks*. PhD thesis, 2021.

[12] T. Hammarström, "Towards explainable decision-making strategies of deep convolutional neural networks: An exploration into explainable ai and potential applications within cancer detection," 2020.

[13] S. Singla, S. Wallace, S. Triantafillou, and K. Batmanghelich, "Using causal analysis for conceptual deep learning explanation," *arXiv preprint arXiv:2107.06098*, 2021.

[14] M. Graziani, J. M. Brown, V. Andrearczyk, V. Yildiz, J. P. Campbell, D. Erdogmus, S. Ioannidis, M. F. Chiang, J. Kalpathy-Cramer, and H. Müller, "Improved interpretability for computer-aided severity assessment of retinopathy of prematurity," in *Medical Imaging 2019: Computer-Aided Diagnosis*, vol. 10950, p. 109501R, International Society for Optics and Photonics, 2019.

[15] A. Lucieri, M. N. Bajwa, S. A. Braun, M. I. Malik, A. Dengel, and S. Ahmed, "On interpretability of deep learning based skin lesion classifiers using concept activation vectors," in *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10, IEEE, 2020.

[16] P. Gamble, R. Jaroensri, H. Wang, F. Tan, M. Moran, T. Brown, I. Flament-Auvigne, E. A. Rakha, M. Toss, D. J. Dabbs, *et al.*, "Determining breast cancer biomarker status and associated morphological features using deep learning," *Communications Medicine*, vol. 1, no. 1, pp. 1–12, 2021.

[17] K. Blumer, S. Venugopalan, M. P. Brenner, and J. Kleinberg, "Using a cross-task grid of linear probes to interpret cnn model predictions on retinal images," *arXiv preprint arXiv:2107.11468*, 2021.

[18] X. Wei, M. J. Gales, and K. M. Knill, "Analysing bias in spoken language assessment using concept activation vectors," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7753–7757, IEEE, 2021.

[19] A. R. Akula, K. Wang, C. Liu, S. Saba-Sadiya, H. Lu, S. Todorovic, J. Chai, and S.-C. Zhu, "Cx-tom: Counterfactual explanations with theory-of-mind for enhancing human trust in image recognition models," *arXiv preprint arXiv:2109.01401*, 2021.

[20] C. Thomas, V. Sharma, and N. Balakrishnan, "Usefulness of darpa dataset for intrusion detection system evaluation," in *Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2008*, vol. 6973, p. 69730G, International Society for Optics and Photonics, 2008.