Explainable AI Feature Selection in Generative Adversarial Networks System aiming to detect DDoS Attacks

Mateus Komarchesqui
Computer Science Department
State University of Londrina
Londrina, Brazil
komarchesqui.mateus@gmail.com

Daniel Matheus Brandão Lent
Electrical Engineering Department
State University of Londrina
Londrina, Brazil
danbranlent@gmail.com

Vitor Gabriel da Silva Ruffo
Electrical Engineering Department
State University of Londrina
Londrina, Brazil
vitor.gs.ruffo@gmail.com

Luiz Fernando Carvalho
Federal Technology University of Paraná
Apucarana, Brazil
luizfcarvalho@utfpr.edu.br

Jaime Lloret
Integrated Management Coastal Research Institute
Universitat Politecnica de Valencia
Valencia, Spain
jlloret@dcom.upv.es

Mario Lemes Proença Jr.
Computer Science Department
State University of Londrina
Londrina, Brazil
proenca@uel.br

2024 11th International Conference on Software Defined Systems (SDS) | 979-8-3315-1832-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/SDS64317.2024.10883903

***Abstract*—With the rapid expansion of global networks and the proliferation of IoT devices, the complexity and scale of traffic have grown exponentially. This surge in connectivity demands larger, faster, and more serviceable architectures, like Software Defined Networks (SDNs). Motivated by various interests, malicious agents seek to compromise services within the network with different attacks. Intrusion Detection Systems (IDSs) are solutions often implemented in SDN using Deep Learning algorithms. These methods are more challenging to explain as they grow in complexity and become less trustworthy for handling sensitive issues like cyber security. This work uses SHapley Additive exPlanations (SHAP) to explain a consolidated IDS that combines Gated Recurrent Units (GRU) with Generative Adversarial Network's discriminator. We conducted a feature selection based on the SHAP explanation and used its insights to better tune the time series's window size hyperparameter. The optimized model performed similarly to the original, with a margin for improvement upon further hyperparameter tuning. It was also more stable in the training phase and faster to execute. This new version of the model was also explained by SHAP and presented a more consistent behavior.**

***Index Terms*—XAI, GAN, SDN, DDoS, SHAP, IDS.**

I. Introduction

Integrating heterogeneous devices, such as smartphones, laptops, and autonomous vehicles, has introduced efficiency and convenience to society. According to Cisco, 500 billion devices will be connected globally by 2030 [1]. Faster networks with more extensive storage capability and serviceability are needed to meet this expectation [2]. Traditional network architectures become more complex and challenging to manage as they grow in size and variety of equipment. Software Defined Networks (SDNs) are a more scalable and straightforward approach to managing. Centralizing control is a trade-off in security as this architecture intrinsically creates a single point of failure, which malicious agents may target to compromise the network services [3].

Attacks based on flooding, like Distributed Denial of Service (DDoS), have a high potential for compromising SDNs [4]. Systems that continuously monitor traffic and search for suspicious activities are called Network Intrusion Detection Systems (NIDS) and are a solution for defending the network against attackers [5]. Since labeling data is expensive and requires previous knowledge about the invasions, unsupervised solutions based on anomaly detection become promising. These are often implemented based on Deep Learning (DL) algorithms that learn the network's normal behavior and render an alert of any abnormal occurrence [6].

A DL algorithm that has been extensively studied in the context of intrusion detection is Generative Adversarial Network (GAN) and its variations [7] [8] [9]. Due to its complexity, GANs are considered black boxes. These comprise uninterpretable models whose weights and biases cannot be directly understood by humans [10].

Trustworthiness is a product of understandability when dealing with sensitive and essential solutions such as cyber security [11]. Protection systems are essential to maintain network confidentiality, integrity, and availability. Thus, entrusting crucial decisions to IDS lacking explainability or failing to offer insights into the reasoning behind their outputs can be risky [12]. Explainable AI (XAI) techniques employ various strategies to explain how black box models work internally; this can be leveraged to increase reliability and trust, besides enabling continuous model optimization, which may improve detection efficacy. This transparency increases trust in automated security systems and equips cybersecurity professionals with actionable insights for optimizing the

implementation and enabling more informed and effective responses to potential threats. SHapley Additive exPlanations (SHAP) is a widely used method based on cooperative game theory [13] [14]. It is employed externally to any already trained model, offering local and global explanations based on feature relevance. SHAP enhances the understandability of the NIDS, clarifying how the traffic features impact the model's decision-making process.

This paper aims to apply Kernel SHAP, an implementation for approximating SHAP, in a consolidated Generative Adversarial NIDS for model explanation, feature selection, and hyperparameter tuning. The evaluated black box system is optimized to become more reliable and stable at training, lighter to execute, and less prone to controller overhead.

This work's contributions are:

- Explain a black box Generative Adversarial Network Intrusion Detection System using Kernel SHAP.
- Leverage the model's explanation to perform feature selection and hyperparameter tuning, yielding a more stable and lighter model.
- Promote trust in the new optimized version of the NIDS by explaining its decision-making process.

The remainder of this paper is organized as follows. Section II presents the related works. Section III discusses the proposed IDS. In Section IV, we present the interpretation of the proposed model from the perspective of SHAP, selecting the features with more significant roles and time steps, retraining the IDS, and explaining the outcome. Finally, Section V concludes the paper.

II. Related Works

SHAP is mainly used to explain the feature's importance globally in various model classes but can also be leveraged to explain observations alone.

C. Kumar and Ansari [15] proposed an IDS that combines the Sheep Flock Optimization Algorithm with the Least Absolute Shrinkage and Selection Operator for feature selection. Four Machine Learning algorithms were trained and tested using the SD-IoT and CIC-IoT-2023 datasets. SHAP was employed to interpret the Decision Tree and XGBoost system variations globally and yielded similar feature importances.

Sharma et al. [16] used a filter-based technique to limit the number of features of NSL-KDD and UNSW-NB15 datasets. They applied a Deep Neural Network (DNN) and a Convolutional Neural Network (CNN) as IDS variations. DNN had a better result and was interpreted using SHAP and LIME. SHAP's local and global explanations showed each dataset's most essential features.

Hooshmand et al. [17] proposed a Network Anomaly Detection System that employs the Synthetic Minority Oversampling Technique (SMOTE) and K-means clustering to treat data imbalances. They used a Denoising Autoencoder to select the top 15 features of NSL-KDD and UNSW-NB15 datasets. The implemented NADS was based on XGBoost and explained globally by SHAP. The authors highlighted the relationship between two features that exhibited a roughly linear and positive trend.

Barnard, Marchetti, and DaSilva [18] combined a supervised XGBoost with an unsupervised Deep Autoencoder to tackle zero-day attacks using the NSL-KDD dataset. The authors used SHAP on single instances to produce insights into the former model's decisions. These explanations feed the Deep Autoencoder, which mainly aims to learn the XGBoost's typical behavior during training. It assumes that a deviation occurs when dealing with never-seen attacks.

Arreche, Bibers, and Abdallah [19] developed a comprehensive two-level ensemble learning framework to enhance the performance of Intrusion Detection Systems. In the first level of their framework, the researchers trained multiple base learners and ensemble methods, and their output was used to generate new datasets with prediction probabilities that fed the second level's training. The framework incorporated feature selection at both levels, utilizing SHAP in the first and Information Gain in the second.

Javeed et al. [20] implemented an IDS combining Bidirectional Long Short-Term Memory Networks, a Bidirectional-GRU, fully connected layers, and a Softmax classifier. The model's experimental results are based on the CIC-DDoS2019 dataset, and its feature's importance was elucidated using SHAP, both locally and globally.

Kumar et al. [21] presented a cybersecurity framework integrating blockchain-enabled smart contracts with Digital Twins (DTs) for Zero-Touch Networks. The framework leverages SHAP to explain the Self-Attention-Based LSTM Intrusion Detection System globally, ranking its feature importances. They propose a layered ZTN model, comprising device, edge, cloud, and digital twin layers, to monitor data and enable secure authentication among entities. Simulated DT datasets and the N-BaIoT dataset were used to validate their system.

Hariharan et al. [22] proposed an IDS framework that utilizes four explainability algorithms to address the complexity and lack of transparency associated with machine learning-based IDSs. They applied Permutation Importance, SHAP, LIME, and Contextual Importance and Utility (CIU) on Random Forest, XGBoost, and LightGBM models. This study demonstrated the use of global and local explainability approaches to enhance IDS transparency on the NSL-KDD and Kaggle datasets, focusing on the effectiveness of these explanations in capturing the importance and stability of features across various attack types. The results emphasized the benefits of combining high prediction performance with interpretability to help security analysts understand model decisions, especially in detecting DoS attack variants through both global and local feature impact analyses.

Oseni et al. [23] propose an explainable deep learning framework aimed at improving the transparency and resilience of Intrusion Detection Systems within IoT-enabled transportation networks. This approach leverages SHAP, specifically the Deep SHAP method. The combination of Deep LIFT with Shapley values is used to elucidate the feature importance in CNN-based IDS predictions using both local and global

explanations. They validated their model on the ToN_IoT dataset, which includes diverse attack scenarios relevant to IoV systems. Deep SHAP's feature attributions aided in enhancing transparency, enabling cybersecurity experts to assess the IDS's threat detection capabilities and optimize cyber resilience.

Most works rely on SHAP and the combination of Shapley values with other XAI techniques, such as Deep LIFT, to elucidate the model's decisions. Feature selection is an aspect of feature importance-based XAI algorithms that is often underutilized. Furthermore, model explanation, feature selection, and hyperparameter tuning simultaneously, as proposed by our work, were not found in most recent intrusion detection studies that addressed eXplainable AI. This paper aims to fill this research gap by applying SHAP to a consolidated GAN-GRU NIDS and performing explanations and optimizations.

III. Evaluated System

This section describes the evaluated GAN-GRU system from Lent et al. work depicted in Fig. 1 [24], used as a case study to demonstrate how SHAP can be leveraged for model optimization. This method uses a pre-processed version of the CIC-DDoS2019 dataset, in which only volumetric attacks are present, and the entries labeled as attacks that presented legit behavior were removed. Since the system focuses on detecting anomalies that have a volumetric effect on traffic, attacks that remained were variations of DDoS, namely NetBIOS, LDAP, MSSQL, UDP, and Syn, based on different protocols. Lent et al. [24] described this processing justified by the entropy features used, and the same conditions were replicated here. The analyzed model has two main characteristics: it is a generative adversarial network and has Gated Recurrent Units (GRU) composing one of its layers. Both are explained in this section.

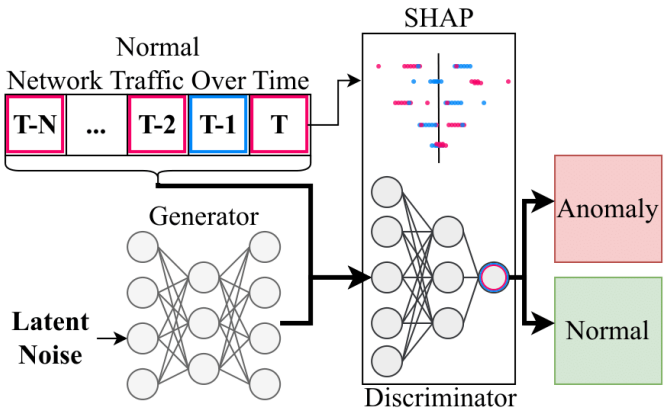

Fig. 1. Overview of the evaluated system adapted to incorporate SHAP explanations; adapted from [24].

A. Generative Adversarial Network

A generative adversarial model is a paradigm in which a generator network is trained to maximize the error of a discriminator network, which is also trained to minimize the same error. Those networks alternate in training until they excel in their respective purpose such that there is no further improvement to any of them. In this stage, both are independent networks that can be used to either generate realistic samples of certain dataset or to act as an anomaly detection system.

Due to the deterministic nature of neural networks inference, the generator requires random noise as input to produce different data on each iteration. Thus, this network acts as a mapping function of the noise set to the original data set. It never receives real samples directly; otherwise, it would have an unfair advantage over the discriminator, hindering its improvement.

The evaluated system's discriminator receives 10 seconds of network traffic to return the probability of the last second containing an anomaly. Consequently, the generator is required to create 10 realistic seconds per sample. Since the considered model uses 6 features for each traffic second, a total of 60 elements must be generated. This amount slows the training and detection process since more layers and neurons on each network are needed.

B. Gated Recurrent Unit

Gated recurrent units are special neurons and an evolution from Recurrent Neural Networks (RNN). RNN are networks that, alongside their input, they receive their previous output, which served as context to be considered in the network's decision. However, some problems may require this context to be "remembered" for longer than a single iteration. Thus the GRU neurons are adapted to store context from previous iterations to improve their outputs.

Structures called gates control how each iteration affects the neuron's internal state, as well as how this state influences the current output. This allows the neuron to store information for several iterations until it becomes relevant or is no longer useful. At this point, the neuron can replace or adjust the information. These gates have their own set of weights that are also tuned during training.

GRU neurons are especially efficient when applied in problems that involve sequential data such as time series. In network traffic, for example, a high packet count might be considered normal when isolated, but, when the previous seconds had significantly fewer packets, an anomaly can be detected. For this reason, 10 seconds of previous traffic are fed as a context for the network to its output. The evaluated study experimented with 5, 10, 15, and 20 seconds of input, with 10 being superior to others.

C. Shannon's Entropy

The showcased model utilizes 6 features: bits per second, packets per second, entropies of source and destination IP, and source and destination port. Bits and packets are straightforward metrics to measure amount of traffic. However, the entropies are a way of summarizing how well distributed the traffic is among the hosts. They allow for IP and port occurrences to be interpreted by neural networks.

Each entropy is a measurement of randomness and is calculated based on how many times an IP or port appear on the collected traffic. When a single element becomes significantly more frequent than the others, the entropy decreases. Similarly, when the amount of different observed elements grows, the entropy also increases.

Equation 1 represents the calculation of Shannon's entropy H in which p_i is the probability of occurrence of the event i and N the number of different events [25]. In this case, each individual IP or port is considered an occurrence, and the probability of it is calculated by the amount of times it appears divided by the total number of flows collected.

$$H = -\sum_{i=1}^{N} p_i \log_2(p_i) \tag{1}$$

IV. SHAP Experimental Analysis

SHAP is a technique used to explain machine learning models proposed by Lundberg and Lee in 2017 [26]. It is based on a cooperative game theory called Shapley value. This theory is a way to distribute a payoff among players according to their contribution to the game's outcome. In the context of XAI, the payoff is the model's output, and players are the features.

SHAP is classified as model-agnostic, post-hoc, additive, and feature importance-based. Model-agnostic refers to the fact that this technique can be applied to any already trained model. Post-hoc XAI methods are those employed after the model training phase, external to the actual model, seeking to explain its decision-making process [27].

This algorithm combines the contributions of individual features in an additive manner, where the explanation is a sum of the features' effects. Let $f(x)$ be the model's prediction for a given input x. Explanation models, such as LIME [28], often use simplified inputs x' that map the original input x through a mapping function $x = h_x(x')$. The additive approach can be expressed by the equation 2:

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i, \tag{2}$$

where ϕ_0 refers to an expected value of the model's prediction across the entire dataset, called a baseline value. The contribution of the i^{th} feature is expressed by ϕ_i. $z' \in \{0,1\}^M$ refers to a binary indicator that expresses the presence of the i^{th} feature in the simplified input z'. M is the number of simplified input features. Local methods try to ensure $g(z') \approx f(h_x(z'))$ whenever $z' \approx x'$.

The classic Shapley Value estimation is given by the equation 3:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|! \cdot (|F| - |S| - 1)!}{|F|!} \cdot [f_{S \bigcup \{i\}}(x_{S \bigcup \{i\}}) - f_S(x_S)], \tag{3}$$

where ϕ_i is the Shapley value for the feature i. F is the set of all features. $S \subseteq F \setminus \{i\}$ expresses that S is any subset of the feature set F that does not include the i^{th} feature. $|S|$ is the number of features in subset S, and $|F|$ the number of features in F. $f_{S \bigcup \{i\}}(x_{S \bigcup \{i\}})$ represents the model's prediction using the features in S combined with the i^{th} feature. $f_S(x_S)$ is the model's prediction over the S subset. The first part of the summation is a weighted factor to determine how much each subset S contributes to the overall Shapley value, and the second part is the marginal contribution of the feature i when added to the subset S.

The classical approximation of Shapley values involves calculating values based on evaluating all possible subsets of features, which can be computationally infeasible for models with many features. This complexity limits the practical application of Shapley values to large-scale problems. Thus, Kernel SHAP was employed since it combines Linear LIME and Shapley values to offer a more computationally efficient approach while still providing accurate and interpretable feature importance. This is achieved through a weighted linear regression model acting as the local surrogate model and a weighting kernel used to estimate the Shapley values.

LIME determines ϕ, from additive equation 2, by minimizing the objective function:

$$\xi = \underset{g \in G}{\arg\min}\, L(f, g, \pi_{x'}) + \Omega(g), \tag{4}$$

where ξ represents the optimal explanation model within the set of interpretable models G. $L(f, g, \pi_{x'})$ is the loss function that measures how well the candidate model g approximates the black box model f. $\pi_{x'}$ is a kernel that assigns weights to the perturbed instances based on their proximity to the instance x'. $\Omega(g)$ is the regularization term applied to g, penalizing more complex candidate models from G.

The key to the Kernel SHAP approach is the weighted kernel, denoted as:

$$\pi_{x'}(z') = \frac{(M-1)!}{(M \; choose \; |z'|)|z'|!(M-|z'|)!}, \tag{5}$$

where $|z'|$ is the number of non-zero features in the input z'. The kernel $\pi_{x'}(z')$ represents a Shapley kernel that determines the weight assigned to each perturbed instance z' when fitting the local surrogate model. It ensures that subsets of features are weighted according to their relevance to the instance x' being explained.

Kernel SHAP combines LIME's local approximation capabilities with SHAP's properties of local accuracy, missingness, and consistency. It provides local explanations that can be aggregated across multiple instances to offer a global perspective of feature importance.

A. Kernel SHAP Analysis

The evaluated system was subjected to Kernel SHAP analysis. Since the dataset's evaluation day includes 8 hours of traffic, a sample of 120 random seconds was extracted for each data category, seeking to reduce the computation time of the explanation. Since the instances labeled as attacks that presented benign behavior were removed from the dataset in the pre-processing phase, random subsets of 120 entries

sampled from the remaining attack registries presented a consistent representation of its entire class. The effects of randomness in the sampling strategy are minimal, making the trade-off between representativity and computational cost justifiable. These samples were concatenated to express the entirety of the dataset, while their separate state was also analyzed.

Fig. 2 expresses the average impact of each feature on the model's output across all selected seconds. The entropy features of *destination port*, *destination IP*, *source port*, and *source IP* are represented by *H(dst port)*, *H(dst ip)*, *H(src port)*, and *H(src ip)*, respectively. All of the features present its time step. For instance, "packets_t-8" is equivalent to the packets per second feature at step 8 of the 10-second window.

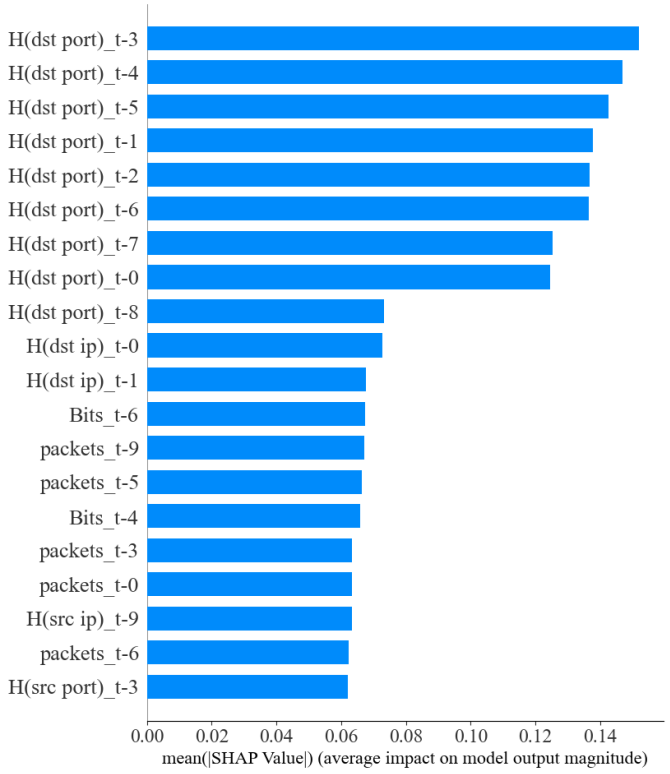

Fig. 2. Summary plot for all data categories.

The most impactful feature was the destination port feature over every time step except the last. Below the eighth destination port, no particular order of features takes place, and their overall impact is mostly less than half of the destination port.

Fig. 3 illustrates the interaction between all time steps of the destination port entropy feature and H(dst ip)_t-0, identified as the tenth most important feature on average (as depicted in Fig. 2). The destination port entropy presents well-defined SHAP value distributions for every time step except for the last and loses partial definition on the penultimate. Time steps 0 through 7 display a positive trend, indicating that it influences the model mainly to predict instances as attacks, which is expected since most analyzed instances are anomalies. A minority of points fall below the zero SHAP value, indicating

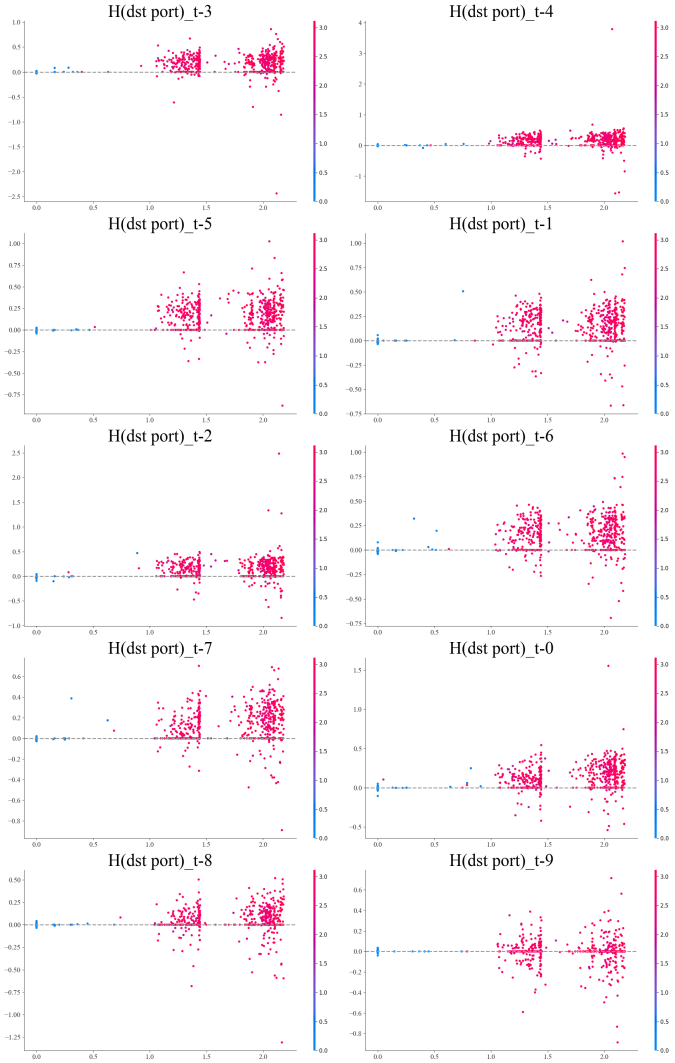

Fig. 3. Interaction between destination port entropies and H(dst ip)_t-0.

regular traffic, which aligns with the data's actual distribution. This design occurs for every interaction between destination port entropy and other features, suggesting that this feature is consistently influential across the model's decision-making process.

Less pronounced patterns emerge when analyzing all other feature interactions that do not comprise the destination port entropy, as exemplified in Fig. 4. The observations scatter across SHAP values, denoting a lack of consistent relationships or clear trends between these features and the model's output.

The separate categories were analyzed to investigate the behavior of the evaluated system further. Fig. 5 displays a grid of the six types of traffic present in the dataset, listed from "a" to "f:"

a) The most influential feature is the entropy of the destination port across time steps 2, 6, and 4. Most features are centered around zero, indicating that normal traffic does not strongly impact the model's predictions. The

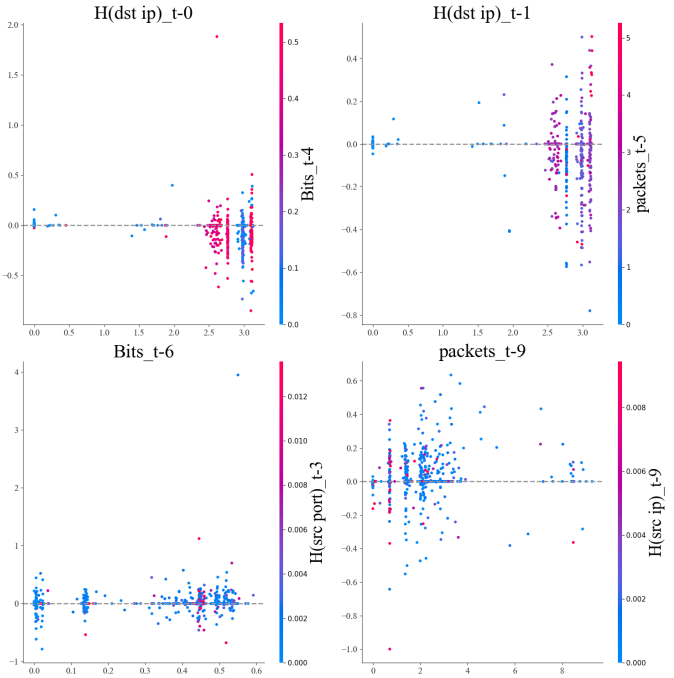

Fig. 4. Interaction between remaining features.

Shapley values are less spread and tend to concentrate around zero.

b) *H(dst port)* from time steps 0 to 7 is among the most critical features, showing a significant spread away from zero towards positive SHAP values. This spread indicates that this feature has a pronounced positive influence on the model's predictions during this attack, strongly suggesting the presence of this type of attack. *H(dst ip)* at time step 0 shows a spread toward negative SHAP values, indicating that this feature hinders the model's prediction of an attack. The remaining features exhibit a more symmetrical spread around zero, suggesting a more balanced or less significant impact on the model's decisions.

c) *H(dst port)* from time steps 0 to 7 tends to spread towards positive SHAP values, impacting model prediction mainly to the positive spectrum. The remaining features are either more symmetrically spread around zero or tend to the negative side, hindering the prediction.

d) *H(dst port)* across time steps 0 to 8 shows behavior similar to the previous attack.

e) The Shapley values of this attack vary less on both sides but show a tendency from destination port entropies towards positive values, especially for time steps 5, 6, 4, 7, and 3.

f) This attack shows a more extensive spread for both sides. Feature *H(dst port)* from time steps 0 to 7 is among the most crucial, showing a tendency for positive values. *H(src ip)* at time step 3 shows a concentration near value zero and more to the negative than positive side, revealing a hindrance to the model's decision.

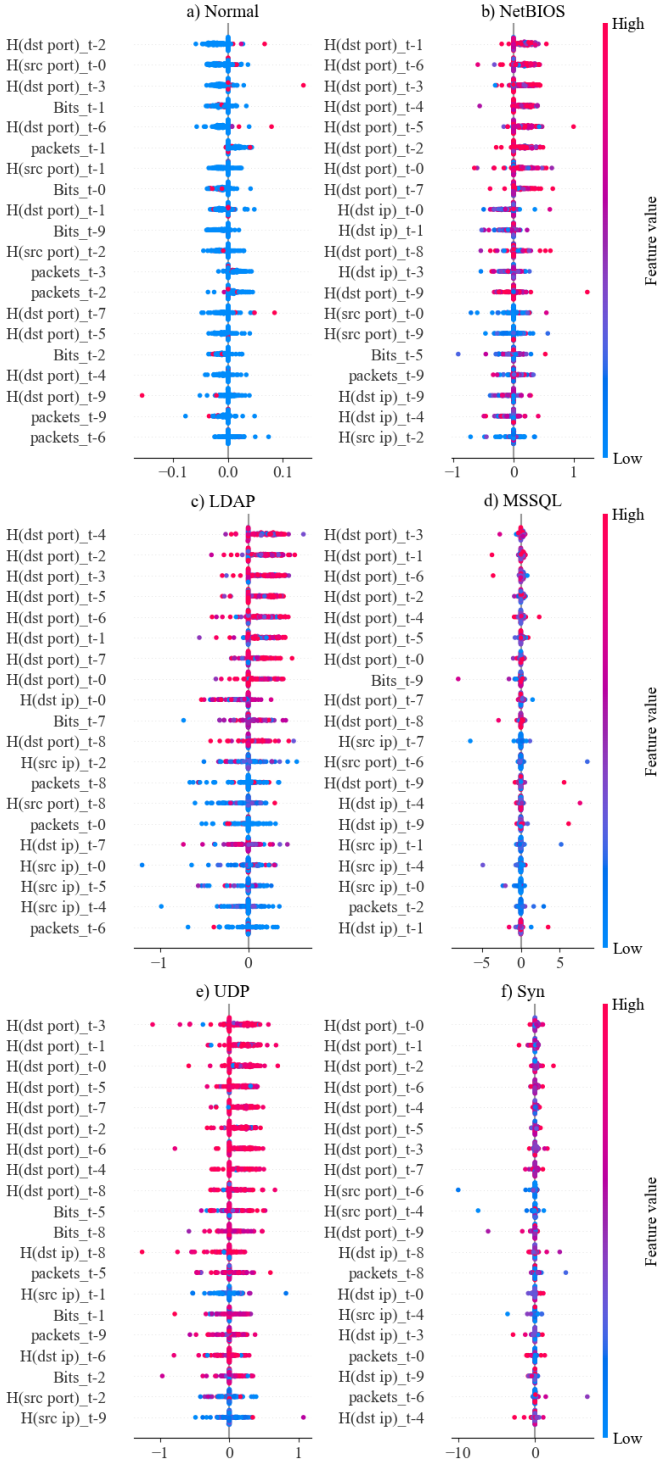

Fig. 5. Summary Plot for Each Type of Traffic.

The analysis of the SHAP summary plots across different traffic types consistently highlights the critical importance of destination port entropy. This feature, particularly from 0 to 7, shows a significant spread toward positive SHAP values in most attack scenarios.

Motivated by the dominant role of *H(dst port)* from time steps 0 to 7 and the less impactful or even detrimental influence of other features, we removed all features except destination port entropy. The window's length was also changed from 10 to 8 seconds, as the contributions of the last two steps of the kept feature were less pronounced. The GAN-GRU model was retrained using only this feature, yielding similar results without modifying any hyperparameters. The model displayed a more consistent training across epochs and was faster to execute.

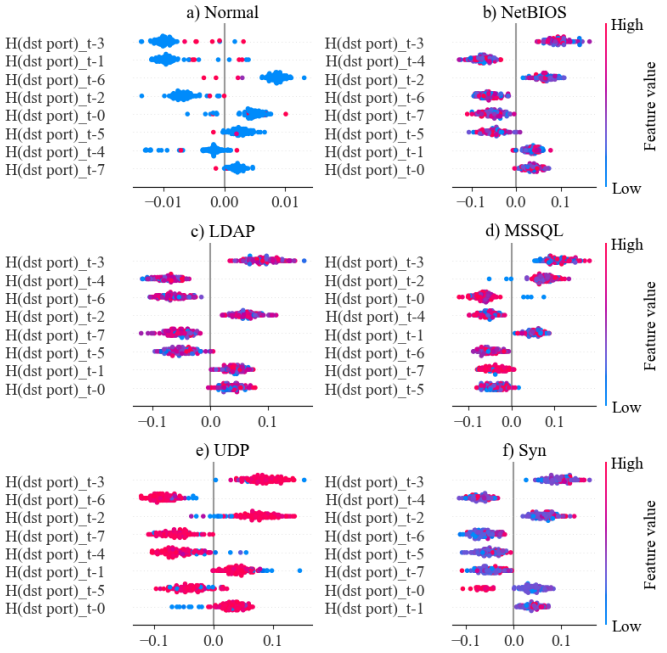

Fig. 6. Summary Plots for Each Type of Traffic after Analysis.

Fig. 6 illustrates the new optimized model's behavior. The features presented a more defined displacement, laying less over the zero SHAP and spreading less to the negative and positive spectrum simultaneously. Initially, the model acquired a MCC of 0.97, with 34 false positives and 44 false negatives. After the optimization, the model with the same hyperparameters, except for the window size, scored an MCC of 0.95, presenting 6 false negatives and 158 false positives on the same training set. Further exploration of new hyperparameters for the optimized model should be conducted for a fairer comparison.

V. Conclusion

In this paper, we evaluated a consolidated black box GAN-GRU Network Intrusion Detection System using Kernel SHAP. The model used a 10-second sliding window to infer the last second. Since the dataset's evaluation day comprises 8 hours of network traffic, each data category was sampled randomly and concatenated for Figures 2, 3, and 4. Each data type was analyzed separately in Figures 5, and 6.

The SHAP explanations suggest that the most impactful feature for anomaly detection is the entropy of the destination port, while the remaining ones present inconclusive behavior. We, therefore, retrained the NIDS black box model, considering only the single influential feature, and measured the change in performance using the MCC metric. After retraining, the model's score was reduced from 0.97 to 0.95. This reduction is coupled with more stability and less computational cost in training. As a result of this optimization, a lighter model was produced, minimizing controller overhead. The hyperparameters and threshold from the original model were kept; therefore, there is potential for improvement through hyperparameter and threshold re-tuning, aiming to address the increased number of false positives. The destination port entropy presented a more concentrated behavior, displaying less distribution across the Shapley values, as illustrated in Fig. 6. Our examination shows how SHAP can be employed for model explanation, feature selection, and hyperparameter tuning. This process yielded a lighter model with more straightforward explanations, less prone to controller overhead, and more accessible for network administrators to understand and trust.

Future work includes exploring other XAI techniques, such as LIME and Deep SHAP, applied to the GAN-GRU NIDS and other state-of-the-art approaches. This investigation would assess whether these alternative explainable methods offer additional benefits regarding model optimization, detection performance, and reliability.

Acknowledgment

This work has been partially supported by the National Council for Scientific and Technological Development (CNPq) of Brazil under the grant of Project 306397/2022-6 and concession of scholarships.

References

[1] Y. B. Zikria, R. Ali, M. K. Afzal, and S. W. Kim, "Next-generation internet of things (iot): Opportunities, challenges, and solutions," *Sensors*, vol. 21, no. 4, 2021.

[2] A. A. Toony, F. Alqahtani, Y. Alginahi, and W. Said, "Multi-block: A novel ml-based intrusion detection framework for sdn-enabled iot networks using new pyramidal structure," *Internet of Things*, vol. 26, p. 101231, 2024.

[3] F. Wahab, A. Shah, I. Khan, B. Ali, and M. Adnan, "An sdn-based hybrid-dl-driven cognitive intrusion detection system for iot ecosystem," *Computers and Electrical Engineering*, vol. 119, p. 109545, 2024.

[4] C. Gkountis, M. Taha, J. Lloret, and G. Kambourakis, "Lightweight algorithm for protecting sdn controller against ddos attacks," in *2017 10th IFIP Wireless and Mobile Networking Conference (WMNC)*, 2017, pp. 1–6.

[5] V. Hnamte, H. Nhung-Nguyen, J. Hussain, and Y. Hwa-Kim, "A novel two-stage deep learning model for network intrusion detection: Lstm-ae," *IEEE Access*, vol. 11, pp. 37 131–37 148, 2023.

[6] G. F. Scaranti, L. F. Carvalho, S. Barbon, J. Lloret, and M. L. Proença, "Unsupervised online anomaly detection in software defined network environments," *Expert Systems with Applications*, vol. 191, p. 116225, 2022.

[7] V. G. da Silva Ruffo, D. M. Brandão Lent, M. Komarchesqui, V. F. Schiavon, M. V. O. de Assis, L. F. Carvalho, and M. L. Proença, "Anomaly and intrusion detection using deep learning for software-defined networks: A survey," *Expert Systems with Applications*, vol. 256, p. 124982, 2024.

[8] U. Sabeel, S. S. Heydari, K. El-Khatib, and K. Elgazzar, "Unknown, atypical and polymorphic network intrusion detection: A systematic survey," *IEEE Transactions on Network and Service Management*, vol. 21, no. 1, pp. 1190–1212, 2024.

[9] M. Ozkan-Okay, E. Akin, Ö. Aslan, S. Kosunalp, T. Iliev, I. Stoyanov, and I. Beloev, "A comprehensive survey: Evaluating the efficiency of artificial intelligence and machine learning techniques on cyber security solutions," *IEEE Access*, vol. 12, pp. 12 229–12 256, 2024.

[10] D. Manivannan, "Recent endeavors in machine learning-powered intrusion detection systems for the internet of things," *Journal of Network and Computer Applications*, vol. 229, p. 103925, 2024.

[11] G. Andresini, A. Appice, F. P. Caforio, D. Malerba, and G. Vessio, "Roulette: A neural attention multi-output model for explainable network intrusion detection," *Expert Systems with Applications*, vol. 201, p. 117144, 2022.

[12] E. S. Ortigossa, T. Gonçalves, and L. G. Nonato, "Explainable artificial intelligence (xai)—from theory to methods and applications," *IEEE Access*, vol. 12, pp. 80 799–80 846, 2024.

[13] N. Moustafa, N. Koroniotis, M. Keshk, A. Y. Zomaya, and Z. Tari, "Explainable intrusion detection for cyber defences in the internet of things: Opportunities and solutions," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 3, pp. 1775–1807, 2023.

[14] M. T. Hosain, J. R. Jim, M. Mridha, and M. M. Kabir, "Explainable ai approaches in deep learning: Advancements, applications and challenges," *Computers and Electrical Engineering*, vol. 117, p. 109246, 2024.

[15] C. Kumar and M. S. A. Ansari, "An explainable nature-inspired cyber attack detection system in software-defined iot applications," *Expert Systems with Applications*, vol. 250, p. 123853, 2024.

[16] B. Sharma, L. Sharma, C. Lal, and S. Roy, "Explainable artificial intelligence for intrusion detection in iot networks: A deep learning based approach," *Expert Systems with Applications*, vol. 238, p. 121751, 2024.

[17] M. K. Hooshmand, M. D. Huchaiah, A. R. Alzighaibi, H. Hashim, E.-S. Atlam, and I. Gad, "Robust network anomaly detection using ensemble learning approach and explainable artificial intelligence (xai)," *Alexandria Engineering Journal*, vol. 94, pp. 120–130, 2024.

[18] P. Barnard, N. Marchetti, and L. A. DaSilva, "Robust network intrusion detection through explainable artificial intelligence (xai)," *IEEE Networking Letters*, vol. 4, no. 3, pp. 167–171, 2022.

[19] O. Arreche, I. Bibers, and M. Abdallah, "A two-level ensemble learning framework for enhancing network intrusion detection systems," *IEEE Access*, vol. 12, pp. 83 830–83 857, 2024.

[20] D. Javeed, T. Gao, P. Kumar, and A. Jolfaei, "An explainable and resilient intrusion detection system for industry 5.0," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 1342–1350, 2024.

[21] R. Kumar, A. Aljuhani, D. Javeed, P. Kumar, S. Islam, and A. N. Islam, "Digital twins-enabled zero touch network: A smart contract and explainable ai integrated cybersecurity framework," *Future Generation Computer Systems*, vol. 156, pp. 191–205, 2024.

[22] S. Hariharan, R. Rejimol Robinson, R. R. Prasad, C. Thomas, and N. Balakrishnan, "Xai for intrusion detection system: comparing explanations based on global and local scope," *Journal of Computer Virology and Hacking Techniques*, vol. 19, no. 2, pp. 217–239, 2023.

[23] A. Oseni, N. Moustafa, G. Creech, N. Sohrabi, A. Strelzoff, Z. Tari, and I. Linkov, "An explainable deep learning framework for resilient intrusion detection in iot-enabled transportation networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 1, pp. 1000–1014, 2023.

[24] D. M. Brandão Lent, V. G. da Silva Ruffo, L. F. Carvalho, J. Lloret, J. J. P. C. Rodrigues, and M. Lemes Proença, "An unsupervised generative adversarial network system to detect ddos attacks in sdn," *IEEE Access*, vol. 12, pp. 70 690–70 706, 2024.

[25] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[26] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[27] A. Theissler, F. Spinnato, U. Schlegel, and R. Guidotti, "Explainable ai for time series classification: A review, taxonomy and research directions," *IEEE Access*, vol. 10, pp. 100 700–100 724, 2022.

[28] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1135–1144.