A deep learning anomaly detection framework with explainability and robustness

Manh-Dung Nguyen, Anis Bouaziz, Valeria Valdés, Ana Rosa Cavalli, Wissam Mallouli and Edgardo Montes de Oca
firstname.lastname@montimage.com
Montimage EURL
Paris, France

ABSTRACT

The prevalence of encrypted Internet traffic has resulted in a pressing need for advanced analysis techniques for traffic analysis and classification. Traditional rule-based and signature-based approaches have been hindered by the introduction of network encryption methods. With the emergence of machine learning (ML) and deep learning (DL), several preliminary works have been developed for anomaly detection in encrypted network traffic. However, complex Artificial Intelligence (AI) models like neural networks lack explainability, limiting the understanding of their predictions. To address this limitation, eXplainable Artificial Intelligence (XAI) has emerged, aiming to provide users with a rationale for understanding AI system outputs and fostering trust. However, existing explainable frameworks still lack comprehensive support for adversarial attacks and defenses.

In this paper, we present Montimage AI Platform (MAIP), a new GUI-based deep learning framework for malicious traffic detection and classification combined with its ability of explaining the decision of the model. We employ popular XAI methods to interpret the prediction of the developed deep learning model. Furthermore, we perform adversarial attacks to assess the accountability and robustness of our model via different quantifiable metrics. We perform extensive experiments with both public and private network traffic. The experimental results demonstrate that our model achieves high performance and robustness, and its outcomes align closely with the domain knowledge.

KEYWORDS

Network Security, Encrypted Traffic Analysis, Malware Detection, Deep Learning, Explainable AI, Adversarial Attacks

ACM Reference Format:
Manh-Dung Nguyen, Anis Bouaziz, Valeria Valdés, Ana Rosa Cavalli, Wissam Mallouli and Edgardo Montes de Oca. 2023. A deep learning anomaly detection framework with explainability and robustness. In *The 18th International Conference on Availability, Reliability and Security (ARES 2023), August 29–September 01, 2023, Benevento, Italy*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3600160.3605052

1 INTRODUCTION

Nowadays, the widespread adoption of HTTPS and Virtual Private Networks (VPN) has led to an increasing amount of encrypted Internet traffic. As of 2022, 95% of all internet traffic is encrypted, and over 85% of attacks occur within encrypted traffic [17]. While encryption serves as a safeguard for user privacy, it simultaneously presents challenges for security tools tasked with traffic analysis and classification. Consequently, there is a growing demand for advanced analysis techniques that leverage alternative criteria, such as behavior analysis, to overcome these obstacles. The introduction of network encryption methods, like the Transport Layer Security (TLS) protocol, has significantly diminished the accuracy and efficiency of traditional Network Intrusion Detection Systems (NIDS), which heavily relied on rule-based and signature-based monitoring detection approaches. As a result, research efforts have shifted towards exploring AI-based analysis methods for malicious traffic detection, aiming to enhance accuracy and effectiveness in the face of encryption [2, 4, 7].

The development of AI-based systems must consider not only accuracy and performance but also additional requirements such as trustworthiness, transparency, unbiasedness, privacy, and robustness. However, complex AI methods like Deep Neural Networks are often perceived as black boxes, lacking explainability. This limitation hampers the understanding of how datasets, input features, and selected models contribute to the predicted classifications. Consequently, there is a pressing need to enhance and optimize anomaly detection and other traffic analysis applications, focusing not only on performance but also on the aforementioned properties. Addressing this need, eXplainable Artificial Intelligence (XAI) [1, 9] has emerged as a prominent research area, aiming to provide users with a rationale for understanding the output produced by AI systems and fostering trust among end-users. Various approaches, such as the ones proposed in [8, 11], have been developed to enhance global and local interpretability, shedding light on what the models have learned and how they make individual predictions.

Existing AI frameworks, such as *Shapash*[1], *explainerdashboard*[2], and *DataRobot*[3], have made valuable contributions to the field. However, they also have certain limitations that need to be addressed. *Shapash* and *explainerdashboard* are open-source frameworks that support explainability in AI models. However, they lack comprehensive support for adversarial attacks and defenses, which is crucial for developing robust AI systems. On the other hand, *DataRobot*, although offering documentation and explainability features, is a commercial product that restricts access and customization options for users and, like the other frameworks, does not provide built-in support for attacks and defenses, hindering its ability to address the growing challenges posed by adversarial ML.

To overcome these limitations, we propose Montimage AI Platform (MAIP), a comprehensive and adaptable framework that outshines existing frameworks to ensure the robustness and reliability

[1] https://github.com/MAIF/shapash
[2] https://github.com/oegedijk/explainerdashboard
[3] https://www.datarobot.com/

Table 1: Comparison between our MAIP and some existing GUI frameworks.

Framework	Open-source	Documentation	XAI	Attacks & Defenses	Metrics	
					Accountability	Resilience
Shapash	✓	✓	✓	✗	✓	✗
explainerdashboard	✓	✓	✓	✗	✓	✗
DataRobot	✗	✓	✓	✗	✓	✗
MAIP	✓	✓	✓	✓	✓	✓

of AI models in different practical applications. We adopt popular XAI methods, named SHAP [8] and LIME [11], to explain the prediction of our malicious traffic detection and classification model. Additionally, MAIP ensures accountability through its quantifiable metrics and exhibits resilience in detecting and mitigating attacks. As shown in Table 1, MAIP is open-source, offers comprehensive documentation, includes XAI functionalities for explainability, and stands out by providing built-in support for attacks and defenses.

Contributions. Our contributions are as follows.

- We develop an open-source GUI-based deep learning framework designed for anomaly detection in encrypted network traffic. Our framework takes into account crucial aspects of an AI system including performance, explainability, and resilience.
- We evaluate the effectiveness of our detection and classification model, providing concrete evidence of its performance through extensive experimentation.
- Furthermore, we assess the accountability and resilience metrics of our model using both public and private datasets, further validating its robustness and reliability.

The rest of the paper is organized as follows. Section 2 discusses our study on existing approaches for Intrusion Detection Systems (IDS). In Section 3, we introduce a deep learning model specifically designed for anomaly detection in encrypted network traffic. Section 4 presents MAIP, a GUI-based framework for deep learning anomaly detection that emphasizes explainability and robustness. We then provide an illustration of these technical concepts within the context of our AI-based framework in Section 5. Finally, Section 6 concludes the paper and offers insights into potential future directions for research and development.

2 RELATED WORK

The field of Intrusion Detection Systems (IDS) has become indispensable in today's Internet-dependent systems due to the increasing threats. IDSs employ various strategies to detect potential attacks in network traffic. We can classify IDSs into three main categories based on the algorithms they use for detection [5].

Rule-based approaches involve monitoring events and comparing them against patterns and signatures based on known attacks and policy rules. These methods exhibit high accuracy and low false positives for established attacks. However, they are ineffective when new or modified attacks occur as each modification of the attack requires completely new or adaptation of already existing rules. Consequently, rule-based IDSs need much more maintenance and regular updates in order to remain relevant [14].

Static-based approaches detect attacks by creating statistical distributions of intrusion patterns and then identifying deviations from the expected distribution. Nevertheless, similarly to rule-based approaches, they need regular updates to be able to accurately investigate traffic. Moreover, in order to create a statistical model to compare with, data needs to be clean and not noisy so that the distributions are representing real-world traffic [5].

Machine Learning-based approaches employ different classifiers to distinguish and classify attacks from normal traffic. These methods require substantial amounts of training data, but with the growing volume of data exchanged over the Internet, they have gained popularity and are the focus of current research. Machine learning-based IDSs aim to detect anomalies and are particularly suitable for identifying unknown attacks [6].

Within the ML-based methods, the classical ML algorithms require less data and computational power, they also often are not able to model the complexity of the problem. On the other hand, DL techniques, including Autoencoders, Deep Belief Networks (DBNs) and Convolutional Neural Networks (CNN), are more adapted to model complex problems and handle large volumes of unstructured data, such as network traffic [6]. Most importantly, DL has the potential for building IDS that are more accurate, faster in prediction and easier in terms of feature engineering preparation for the training of the model. Moreover, DL techniques have demonstrated the ability to detect new and unidentified intruders [5].

3 DL MODEL FOR ANOMALY DETECTION

Considering these latest advancements, we employ a deep learning technique that combines Convolutional Neural Networks (CNN) and Stacked Autoencoders (SAE). This hybrid approach allows us to benefit from the dimensionality reduction of SAE and the high accuracy of CNN classification. Figure 2 depicts an overview of our proposed DL model for malicious traffic detection.

Feature engineering module employs the open source Montimage monitoring tool[4] (MMT) to parse raw network traffic, extract the needed information, compute the features required by the DL module and translate them into a numeric form. Technically, MMT-Probe is a monitoring and data extraction software that parses network traffic to extract network and application-based events, such as protocol field values and statistics. It allows parsing of a variety of network protocols (e.g., TCP, UDP, HTTP, and more than 700) for the purpose of extracting metadata. Furthermore, the architecture of the MMT-Probe is modular, allowing for the integration of new protocols for parsing purposes. The features consist of multiple parameters that are always computable on raw traffic independently of whether the traffic is encrypted or not, such as,

[4] https://github.com/Montimage/mmt-probe

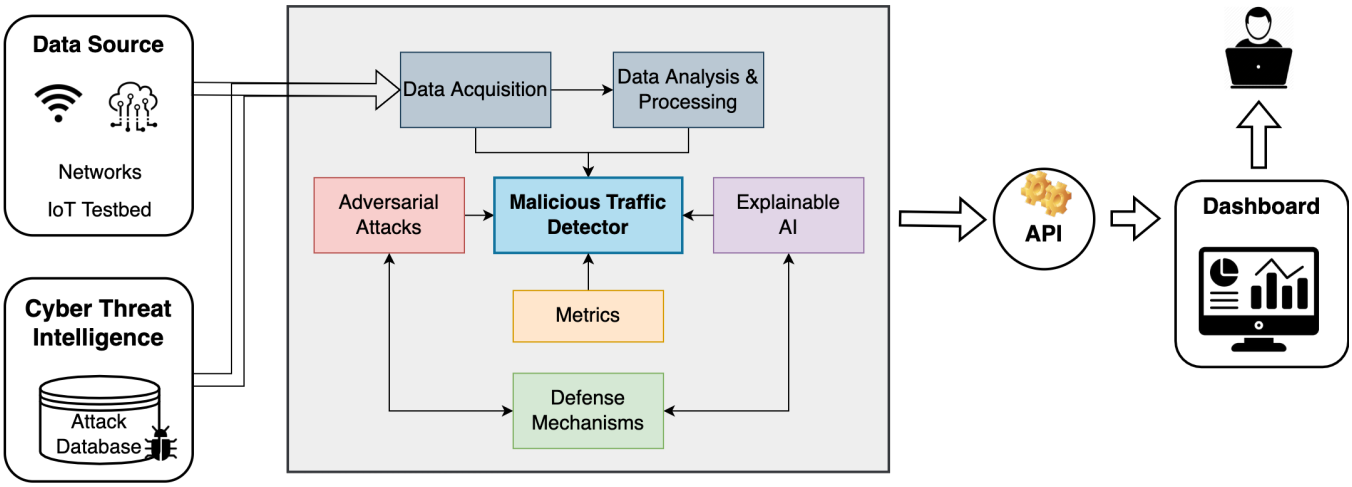

Figure 1: Architecture of our AI-based malicious traffic detector.

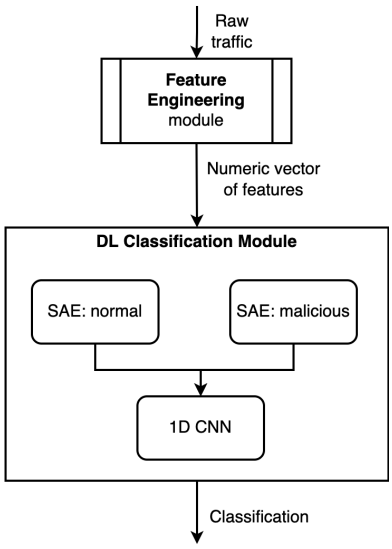

Figure 2: Overview of our deep learning model.

the statistics involving byte and time information. In particular, we extract 59 features, including basic features in packet headers and statistical features after performance traffic aggregation into flows. Finally, the restructured and computed data is transformed into a numeric vector so that can be easily processed by our AI model.

Deep learning malicious traffic detector module is responsible for creating and utilizing a ML model able to classify the vectorized form of network traffic data. Our proposed approach involves a hybrid model that combines two DL techniques: Stacked Autoencoders (SAE) and Convolutional Neural Networks (CNN). We first train two SAEs, one for each class of data (normal or malicious). Each SAE is designed with one hidden dense layer and trained separately with their respective data. The output of each SAE is then concatenated to form a single vector, which is then passed as input to the CNN. The CNN structure is based on the well-known VGG16 model [12], which consists of three repeated segments that are built from two convolutional layers (Conv1D) followed by a MaxPool layer (MaxPooling1D). After three blocks of such a structure, a flatten layer (Flatten) followed by two dense layers (Dense) are used in order to provide the final classification.

In the learning phase, the model is fed with a dataset used for training and tuning the model in order to obtain the best performance and highest accuracy of the final classification. This phase requires multiple experiments with the use of different structures of the particular DL model and its hyperparameters. The second mode of operation, used during the prediction phase, requires an already trained model that is used on the incoming data from the feature engineering module. This mode allows quick prediction of the membership class of the incoming flows. The output of the DL module consists primarily of the classified flows, such as malicious or benign traffics.

4 FRAMEWORK FOR ANOMALY DETECTION WITH XAI AND ROBUSTNESS

4.1 Architecture

In this section, we propose Montimage AI Platform (MAIP), an AI-based framework for anomaly detection in encrypted traffic with high performance, explanation and robustness against adversarial attacks. Figure 1 shows the architecture of our anomaly detection framework.

Data acquisition module collects raw traffic data from networks or IoT testbed in either online or offline mode. It can also use Cyber Threat Intelligence (CTI) sources, e.g., deployed honeypots, to learn and continuously train our model using attack patterns and past

Figure 3: A succinct Metrics page of MAIP showing different accountability and resilience metrics.

anomaly information in the database. As discussed in Section 3, *Data analysis & processing* and *AI-based malicious traffic detector* module of the MAIP correspond to the feature engineering and building DL models step, respectively.

Because of a complex hard-to-interpret DL model with a large number of features used for malicious traffic detection purposes, *interpretability* is crucial to earn trust of its end user. *Explainable AI* module aims at producing post-hoc global and local explanations of predictions of our model. Concretely, we employ popular model agnostic post-hoc XAI techniques, like SHAP [8] and LIME [11] to explain predictions of our proposed malicious traffic detection and classification DL model.

Adversarial machine learning [15] is a significant challenge for the security and reliability of machine learning models, particularly in high-stakes applications such as autonomous vehicles, medical diagnosis, and financial fraud detection. Therefore, *Adversarial attacks* module aims at injecting various evasion and poisoning adversarial attacks for robustness analysis of our system. Additionally, *Defense mechanisms* module provides countermeasures to prevent attacks against both AI and XAI models.

The relationship between XAI and adversarial attacks is intricate and multi-dimensional. On one hand, increasing the transparency and interpretability of a model can make it more vulnerable to adversarial attacks. By leveraging the explanations provided by XAI methods, attackers can identify model weaknesses and create more potent adversarial examples. On the other hand, if a model lacks transparency and interpretability, it becomes difficult not only to understand the reasoning behind its predictions but also to detect and address adversarial attacks. Hence, it is crucial to find a balance between XAI and adversarial attacks to develop secure and robust machine learning models. As we need to consider the tradeoff between explainability, robustness and performance of our AI system, *Metrics* module allows to measure quantifiable metrics for its accountability and resilience.

We design our malicious traffic detection framework to be easily accessible for users or developers through a *Dashboard*. It provides a range of ML services, including extract features, build or retrain the model, inject adversarial attacks, produce explanations and evaluate our model using different metrics. Each of these services is exposed through dedicated *APIs* that can be accessed through the server, making it easy to integrate with other applications and systems.

4.2 Implementation

Our framework is designed with a server written in ExpressJS, that employs the MMT-Probe tool written in C for feature extraction and leverages popular Python libraries for DL and XAI. As shown in Table 2, MAIP offers a complete set of features and APIs that cover various aspects of AI and XAI. The client is built in React and accessible via Swagger APIs, offering users an intuitive and user-friendly interface to interact with the DL services. For instance, Figure 3 illustrates the metrics page, which provides users with a comprehensive overview of accountability and resilience metrics for a specific model. This page presents a wide range of metrics that enable users to evaluate the model's performance and assess its robustness against potential adversarial attacks. The code of our framework is publicly available at https://github.com/Montimage/maip.

5 EXPERIMENTAL EVALUATION

5.1 Evaluation Setup

Datasets. In order to evaluate the performance of our AI-based framework, we use below datasets mentioned in Table 3. We first

Table 2: Some important APIs.

Category	API	Description
Feature extraction	POST /mmt/offline	Start analyzing a pcap file
	POST /mmt/online	Start monitoring a network interface in real-time
DL models	POST /build	Start building a DL model
	POST /retrain	Start retraining a model
	GET /models	Obtain the list of all models
	GET /models/{modelId}	Obtain detailed information of a specific model
	POST /predict	Start a prediction
XAI	POST /xai/shap	Perform SHAP method to produce explanations
	POST /xai/lime	Perform LIME method to produce explanations
Attacks	POST /attacks/ctgan	Perform CTGAN attack to generate synthetic tabular samples
	POST /attacks/poisoning/ctgan	Perform a poisoning attack with CTGAN
	POST /attacks/poisoning/rsl	Randomly choosing two samples of the training dataset and swapping their labels
	POST /attacks/poisoning/tlf	Flip labels of some samples from one class to the target class
Metrics	GET /metrics/{modelId}/accuracy	Obtain accuracy metric of a specific model
	GET /metrics/{typeAttack}/{modelId}/impact	Obtain impact metric of a model under a specific attack

Table 3: Overview of datasets with the number of normal and malicious traffic flows.

Dataset	Number of flows	
	Normal traffic	Malicious traffic
Botnet	14204	1014
Infiltration	10176	2980
Honeypot	2610	4516

Table 4: Accuracy of our framework.

Dataset	#Training	#Testing	Accuracy
Botnet	31704	13586	0.99
Infiltration	7000	3000	0.97
Honeypot	5273	2260	0.94

use the open-source database CSE-CIC IDS2018[5], that includes different attack scenarios, such as botnet and infiltration of the network from inside. Additionally, we also use the network traffic captured from our infrastructure to distinguish between human and bot-generated traffic. While human traffic was captured on a local network, bot traffic was captured by our honeypots deployed on different cloud services (e.g., Amazon, Gandi).

Threat model. Here we consider a *white-box* attack model, meaning that the attacker has a complete knowledge about the DL model. The attacker's objective is to compromise the integrity of our framework leading to a significant degradation in the model's accuracy by injecting following popular attacks:

- *GAN-based poisoning* attack generates or adds synthetic data that looks very similar to the real data. Concretely, we use CTGAN [16] for modelling tabular data to generate synthetic samples.
- *Random swapping labels* attack chooses randomly two samples of the training dataset and swaps their labels.
- *Target label flipping* attack flips the labels of some samples from one class to the target class (e.g., malicious class).

Evaluation metrics. We measure the performance and robustness of our model using following evaluation metrics:

- *Performance* measures the model's performance, including accuracy, F1-score, precision and recall.
- *Poisoning rate* (or *attacker's capability*) is the percentage of poisoned samples S_p out of a training dataset ($S + S_p$).
- *Accuracy decrease* measures the decrease of a performance score between benign model F and poisoned model F_p.

Research questions. In the next section, we investigate four main research questions as follows:

- **RQ1**: Is our AI-based framework effective in detecting different types of malicious traffics?
- **RQ2**: Is our AI-based malicious traffic detector robust under different adversarial attacks? How similar are the CTGAN-generated samples and the real ones?
- **RQ3**: Are XAI methods helpful in providing explanations of the prediction results of our detector and in detecting the existence of adversarial attacks?
- **RQ4**: Can the adversarial training method be effective as a countermeasure?

5.2 Experimental results

RQ1. For all the three datasets, the testing dataset is created using 30% of the total data. Table 4 shows the performance metric of our system. Overall the proposed method yields high detection performance with the accuracy 0.99, 0.97 and 0.94 for detecting botnets, infiltration and honeypot, respectively.

> *Our model achieves very good performance.*

RQ2. Figure 4 is a graphical representation of the cumulative sum of the real data and generated by CTGAN. Clearly, these plots of the synthetic and real data are quite similar, which implies that

[5] https://www.unb.ca/cic/datasets/ids-2018.html

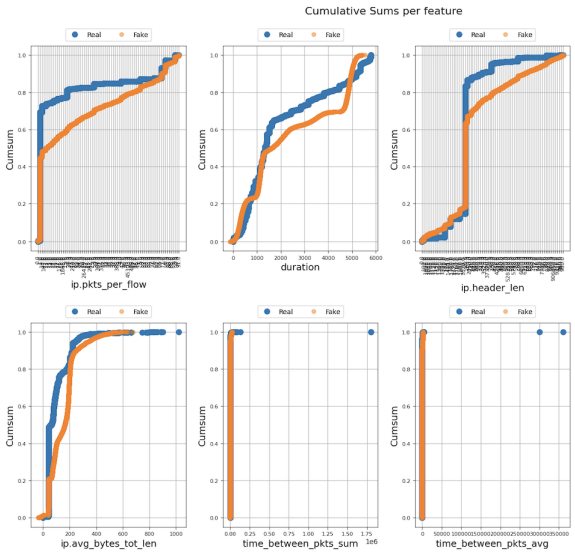

Figure 4: Cumulative sum plot for the infiltration dataset (real data in blue and synthetic data in orange).

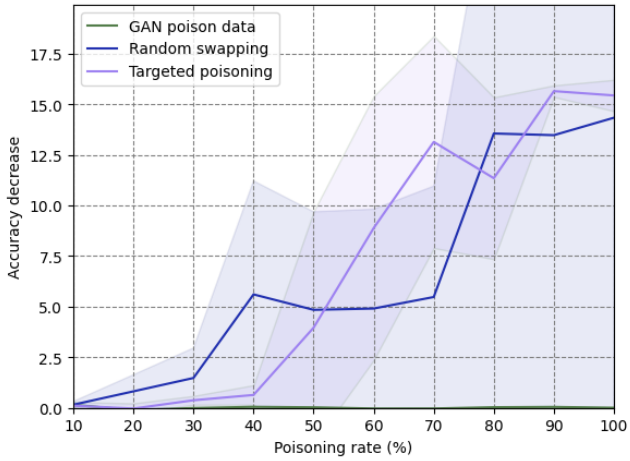

Figure 5: Robustness of our DL model for infiltration detection against three adversarial attacks.

the CTGAN successfully produces synthetic data that match the statistical properties of the real data.

> *GAN-generated samples closely resemble the real data.*

As depicted in Figure 5, increasing the poisoning rate often degrades the model accuracy of our model. Among the three attacks, GAN-based poisoning attack has a lesser impact on our model performance, while two other attacks have more effect on the accuracy of our model, especially after increasing the poisoning rate to 50%. Interestingly, our model still achieves very good accuracy even under the targeted poisoning attack with the poisoning rate up to 40%.

> *The model still achieves pretty good accuracy, especially 95% for infiltration detection, even under a high volume 40% of poisoned data.*

RQ3. SHAP produces explanations of predictions of a model by identifying the most important features based on a feature attribution framework and Shapley values. Due to our resource limitation, we randomly select 50 samples from each dataset for malicious class to generate explanations in the form of SHAP value. We have shown the top 20 class-wise important features by sorting the sum of magnitudes of SHAP value over these samples for the botnet dataset, as depicted in Figure 6. Here the length of the bar means how much influence the feature has on the prediction. Some important observations using SHAP framework are as follows:

- The most important feature in detecting all 3 attacks is *flow duration*. It is also a common characteristic used in machine learning algorithms [3, 13] because malicious communications often exhibit specific flow duration patterns. For instance, some botnets establishing brief connections while others are more chatty, resulting in longer duration.
- A simple strategy of a botnet to prevent detection is by randomly reconnecting as a normal established connection. Thus, some features related to the number of TCP packets with some flags, such as $tcp.fin$ (flag Finished) and $tcp.rst$ (flag Reset), for closing a connection, are indeed important for botnet detection.

By comparing the SHAP signatures with (in Figure 7) and without attack (in Figure 6), we clearly see the differences between the list of important features produced by SHAP. For instance, the feature $duration$ is no longer the most important feature for botnet detection. Furthermore, none of those features related to number of TCP packets with specific flags appear in the top 20 feature list under the target label flipping attack. As a result, SHAP signatures can be useful to detect the existence of adversarial attacks by identifying the features whose SHAP values have been significantly altered due to the attack.

> *XAI explanations show that our model's predictions have parity with the domain knowledge. Moreover, SHAP signatures are useful to detect attacks.*

RQ4. As shown in Figure 5, the accuracy decrease metric after adding more synthetic data produced by CTGAN is almost zero, which implies that the original model and the retrained model have the same performance against the testing dataset. It could suggest that the model after adversarial training is now more robust against adversarial attacks.

> *The adversarial training is effective to improve the robustness of our model.*

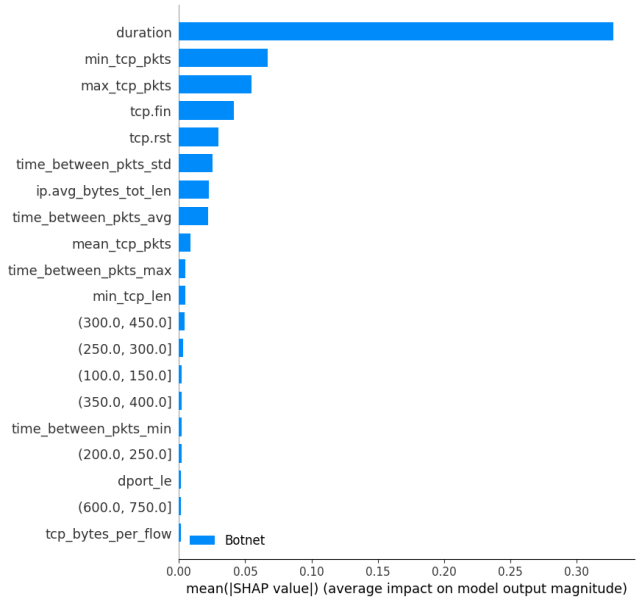

Figure 6: SHAP summary plot for botnet detection.

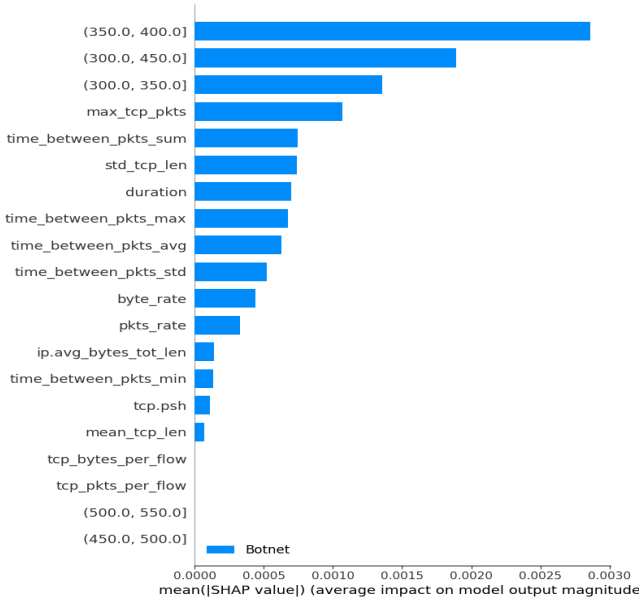

Figure 7: SHAP summary plot under the target label flipping attack with poisoning rate 50% for botnet detection.

6 CONCLUSION

In this paper, we emphasized the role of explainability as a countermeasure technique, illustrating its potential to improve robustness and enhance transparency and user trust in AI-based applications. Specifically, we showcased its application within Montimage AI Platform (MAIP), a new AI-based framework for detecting malicious encrypted traffic, demonstrating how it enables effective explanations and robustness against adversarial attacks.

For future work, two key directions should be pursued. Firstly, it is essential to explore and implement additional defense mechanisms to fortify our model against potential attacks from adversarial ML and XAI models. By augmenting our system with stronger defenses, we can ensure its resilience and reliability in real-world scenarios. Secondly, extending the framework to encompass different use cases, such as user network classification in the context of 5G networks [10], would significantly broaden its practical utility. By adapting the framework to diverse domains, we can evaluate its performance, scalability, and adaptability, thereby contributing to the advancement of AI technologies in specific contexts.

ACKNOWLEDGMENTS

This research is supported by the H2020 projects PUZZLE N° 883540, SPATIAL N° 101021808 and AI4CYBER N° 101070450.

REFERENCES

[1] Alejandro Barredo Arrieta et al. 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information fusion* (2020).
[2] David Brumley, Cody Hartwig, Zhenkai Liang, James Newsome, Dawn Song, and Heng Yin. 2008. Automatically identifying trigger-based behavior in malware. *Botnet Detection: Countering the Largest Security Threat* (2008).
[3] Livadas Carl, R Walsh, D Lapsley, and WT Strayer. 2006. Using machine learning technliques to identify botnet traffic. In *Local Computer Networks, Proceedings 2006 31st IEEE Conference on. IEEE.*
[4] Weidong Cui, Randy H Katz, and Wai-tian Tan. 2005. BINDER: An extrusion-based break-in detector for personal computers. In *USENIX Annual Technical Conference, General Track.*
[5] Dilara Gümüşbaş, Tulay Yıldırım, Angelo Genovese, and Fabio Scotti. 2020. A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems. *IEEE Systems Journal* (2020).
[6] Donghwoon Kwon, Hyunjoo Kim, Jinoh Kim, Sang C Suh, Ikkyun Kim, and Kuinam J Kim. 2019. A survey of deep learning-based network anomaly detection. *Cluster Computing* (2019).
[7] Hemank Lamba, Thomas J Glazier, Javier Cámara, Bradley Schmerl, David Garlan, and Jürgen Pfeffer. 2017. Model-based cluster analysis for identifying suspicious activity sequences in software. In *Proceedings of the 3rd ACM on International Workshop on Security and Privacy Analytics.*
[8] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems* 30 (2017).
[9] Azqa Nadeem, Daniël Vos, Clinton Cao, Luca Pajola, Simon Dieck, Robert Baumgartner, and Sicco Verwer. 2022. Sok: Explainable machine learning for computer security applications. *arXiv preprint arXiv:2208.10605* (2022).
[10] Manh-Dung Nguyen, Vinh Hoa La, R. Cavalli, and Edgardo Montes de Oca. 2022. Towards improving explainability, resilience and performance of cybersecurity analysis of 5G/IoT networks (work-in-progress paper). In *2022 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW).*
[11] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " 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.* 1135–1144.
[12] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
[13] W Timothy Strayer, David E Lapsley, Robert Walsh, and Carl Livadas. 2008. Botnet detection based on network behavior. *Botnet detection* 36, August (2008), 1–24.
[14] Petr Velan, Milan Čermák, Pavel Čeleda, and Martin Drašar. 2015. A survey of methods for encrypted traffic classification and analysis. *International Journal of Network Management* (2015).
[15] Xianmin Wang, Jing Li, Xiaohui Kuang, Yu-an Tan, and Jin Li. 2019. The security of machine learning in an adversarial setting: A survey. *J. Parallel Distributed Comput.* (2019).
[16] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2019. Modeling tabular data using conditional gan. *Advances in Neural Information Processing Systems* 32 (2019).
[17] Zscaler. 2022. State of Encrypted Attacks.