2024 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT) | 979-8-3503-6808-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/COMNETSAT63286.2024.10862241

DeepNIDS: A Deep Neural Network-Based Network Intrusion Detection System for IoT

Xuan-Duong Nguyen
University of Information Technology
Vietnam National University
Ho Chi Minh City, Vietnam
duongnx.17@grad.uit.edu.vn

Xuan-Ha Nguyen
University of Information Technology
Vietnam National University
Ho Chi Minh City, Vietnam
hanx.17@grad.uit.edu.vn

Hoang-Hai Huynh
University of Information Technology
Vietnam National University
Ho Chi Minh City, Vietnam
haihh.18@grad.uit.edu.vn

Khanh-Hoi Le-Minh
University of Information Technology
Vietnam National University
Ho Chi Minh City, Vietnam
hoilmk@uit.edu.vn

Kim-Hung Le
University of Information Technology
Vietnam National University
Ho Chi Minh City, Vietnam
hunglk@uit.edu.vn
Corresponding author: hunglk@uit.edu.vn

***Abstract*—Recently, the widespread use of Internet of Things (IoT) has been triggering an exponential increase in the number of smart devices lacking hardware security supports. This gives rise to various challenges in Cyber-threat protection. In this paper, we present DeepNIDS: a lightweight neural network-based network intrusion detection system (NIDS) that effectively detects abnormal traffic. The core algorithm of DeepNIDS is a novel Convolutional Neural Network (CNN) model, specifically designed for classifying network traffic patterns. To enhance the detection performance, we employ 2D reshaped features as the input of our model, which is extracted and reshaped from network traffic over a period using Damped Incremental Statistics algorithm. Our experimental results show that DeepNIDS could identify nine types of attacks, showing superior detection capabilities over existing NIDS, with an average accuracy of approximately 98.48%.**

***Index Terms*—ANOMALY DETECTION, CONVOLUTIONAL NEURAL NETWORK, NETWORK INTRUSION DETECTION.**

I. INTRODUCTION

The rapid advancement in internet technology has significantly benefited various sectors including healthcare, public safety, and industrial Internet of Things (IoT), enhancing the modernization of daily life. This also raises several concerns about the security of computer networks with a notable increase in network attacks over recent years [1]. In addition, the techniques of attacks have been becoming more diverse and sophisticated. The attackers launch different types of attacks on the target systems and exploit the security vulnerabilities. Such attacks might include block network traffic, control system, leak private data which make the network systems vulnerable and dangerous [2]. Therefore, it is necessary to introduce Network Intrusion Detection Systems (NIDSs) to protect network systems from attackers.

Recently, machine learning, particularly its subfield deep learning, has been successfully applied in a variety of application domains, such as image and speech recognition, natural language processing, and autonomous driving, revolutionizing these fields with its capabilities in classification, regression, and clustering. However, most of the deep learning approaches requires massive computational resources for executing inference tasks [3]. This limits the NIDS to being operated on resource-constrained devices with limited memory and processing power like gateways or routers [4], [5]. In addition, the false-positive rate of such methods is significantly high. This leads to massive false alerts sent to network administrators.

To handle these problems, we introduce a lightweight deep learning-based NIDS powered by Convolutional Neural Networks (CNN) architecture. The core model uses 2D reshaped features, which are extracted and reshaped from network traffic over a time span using Damped Incremental Statistics algorithm. It repeatedly modifies the network parameters to approximate the corresponding output to increase the detection quality. As a result, our proposed IDS effectively detects various types of network attack (nine types) with high accuracy. Due to its simplicity, the model could also be integrated into network devices to monitor network traffic and identify abnormal network traffic patterns. Our contributions are summarized below:

- We propose lightweight deep learning-based NIDS powered by convolutional neural networks that effectively detects various networking attack types.
- We introduce a novel approach to analyze network features reshape $1D$ features to $2D$ features, which notably increases detection accuracy.
- We thoroughly evaluate several performance indicators of our proposal on the public networking datasets.

The outline of this paper is structured as follows: Section II presents related works. The details of our system are introduced in Section III. Section IV introduces our implements, the evaluation dataset, the metrics, and reports our experiment results. Finally, section V summarizes the paper.

II. RELATED WORKS

The significant rise in technology over the past decade has led to an explosion in network scales and network nodes

handling various applications. Therefore, recently, applying Artificial Intelligence (AI) to Network Intrusion Detection System (NIDS) is newly trending on researching to replace outdated Signature-based NIDS [6]. However, these researchers mostly focus on improving the detection rate and performance in training, executing but not toward deploying NIDS on a specific network context such as IoT network, real-time monitoring [7].

At the dawn of abnormally-based NIDS, researchers proposed a simple algorithm for NIDS. In [8] the authors compute during a training phase a profile byte frequency distribution then use Mahalanobis distance during the executing phase to compute the similarity of the new data to the preconfigured profile. An alert is generated when the distance of the new input exceeds this threshold. Some simple machine-learning algorithms like KNN or SVM also proposed in [9] and [10]. Another popular Machine learning model is Artificial neural network (ANN). This model can perform nonlinear modeling from large data sets, but the main issue is high time consumption because of its complex nature. This drawback slows down the training phase, resulting in an ineffective solution. To deal with this problem, in [11], Ali et al. presented an improved model based on Particle swarm optimization and FLN. The model was evaluated on the KDD Cup'99 dataset and got a better result compared with several efficiency algorithms. In [12], the authors proposed a robust decision tree for intrusion detection system that could detect various attact at high accuracy. In [13], the authors raised a deep hierarchical model based on Convolutional neural networks (CNN) and Bidirectional LSTM (BiLSTM) architecture. This model was tested on two public datasets named NSL-KDD and UNSW-NB15 and performed better than other competitors. In [14], the authors proposed an two-stage scheme that could detect various variant of DDos attracts effectively in both accuracy and processing rate.

All the above researches have inspired us to use a CNN model in building a NIDS. Our research aims to optimize models which can classify various attacker type with low resources consumption. Unlike others, our research uses Damped incremental statistic Feature extraction from [15] to dynamically retain and retrieve implicit contextual characteristics of network traffic. This will reduce our CNN model's complexity and also provide the scalable ability for real-time monitoring.

III. The DeepNIDS

A. Overview

Our proposal is deep learning-based NIDS based on a CNN model designed to detect and classify abnormal samples in network traffic efficiently. The detection process consists of two phases: (1) extracting raw data throughout monitoring the network traffic; (2) detecting and classifying abnormal network traffic. Since we aim to run the model on a constrained network device with limited memory and computational power, our model is optimized to minimize complexity but still archives well-performance on classify network traffic. It will include five following components:

- **Packet Capturer**: This component is used to receive network packets by using the external framework and librar, such as NFQueue [16], afpacket [17], and tshark [18].
- **Packet Parser**: It parses raw binary packet into valuable and unified information (meta-information) required by the Feature Extractor. This component acts as a decapsulation to convert information.
- **Feature Extractor (FE)**: A custom component takes charge of calculating n features from the packet's metadata then these features used to create the vector $\vec{x} \in R^n$. Vector x describes the features of package and its relevant packages.
- **Pre-processing features**: A custom component to normalize and reshape features from 1-D into 2-D. This step performs mapping between the output of FE and the input of the classification model.
- **Classification model (CM)**: CM is the core component of the proposed NIDS using a novel CNN-based model to identify abnormal traffic behaviors. The detection results are then transferred to the action manager to propose appropriate reactions if a network attack is detected.

We describe FE, Pre-processing features, CM components in detail, and ignore Packet Capturer and Packet Parser components that are well-explained at their external libraries and frameworks. To clearly understand how our proposal works, the workflow is illustrated in Fig. 1.

1) First of all, the Packet Capture monitors the raw network packages, captures and sends them to the Packer Parser.
2) The Packer Parser takes raw packets and exploits metadata information from them. The output is prepared in a local file or sent to FE.
3) The FE uses this metadata to compute 100 statistic features that describe the current network states. These features form the vector $\vec{x} \in R^n$.
4) Perform a pre-processing features step to normalize vector $\vec{x}$, then reshape normalized vector into $2 - D$ vector, consider as an image I with the flatten size equal n. This image is pushed into the CM. In training mode, metadata needs to be labeled and prepared in a local file.

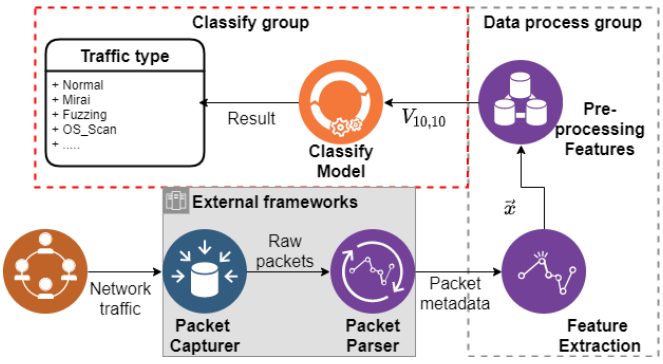

Fig. 1: The overview of DeepNIDS

5) The CM receives the image I and ...
   - In training mode: we group images and corresponding labels into train data and validation data. Then use these data to train and validate the model.
   - In executing mode: we employ the well-trained model to identify attack patterns from the network traffic.

B. Feature Extractor

Feature extraction is a method to extract critical data from raw data. Therefore, in order to create highly efficient and accurate machine learning models, the method used to extract data must selectively extract specific information in each attack. For example, for SYN Flooding attack that packets can be millions of attack packets generated by the SYN Flooding attack. It can also be that packet is generated for client connection requests to the network normally. We need context information to get an accurate judgment from these as well as actual attack cases. For this reason, we chose the Damped Incremental Statistics algorithm, proposed by Realguard [5], as the algorithm to extract the features. However, we remove the SrcIP field to match our model (100fts = 10x10). In each time window, the feature extractor computes the features based on three meta information per packet as srcMAC-IP (source IP and MAC address), Channel (source and destination IP address), and Socket (source and destination port TCP or UDP Socket).

Let $S = x_1, x_2, ...$ ($x_i \in \mathbb{R}$) is an unbounded data stream, and:

$$d_\lambda(t) = 2^{-\lambda t}$$

is the decay function where $\lambda > 0$ is the decay factor and t is the timestamp difference between two observations. The feature extractor maintains an array:

$$IS_{i,\lambda,t_{last}} = (w, LS, SS, SR_{ij})$$

for each stream $S_{i,\lambda}$, where w is the current weight ,t_{last} is the last updated timestamp of IS_i, LS and SS are the linear sum and the squared sum of instance observed so far, and SR_{ij} is the sum of residual products between two attribute streams, which are used for calculating 2-D features. To update the $IS_{i,\lambda,t_{cur}}$ in realtime with x_{cur} at time t_{cur}, the feature extractor has the update steps shown as follows:

Algorithm 1: Update $IS_{i,\lambda,t_{cur}}$

Input : $IS_{i,\lambda,t_{last}}, x_{cur}, t_{cur}, t_{last}, r_j$
Output: $IS_{i,\lambda,t_{cur}}$
$\Delta = d_\lambda(t_{cur} - t_{last}) IS_{i,\lambda,t_{last}}$
$IS_{i,\lambda,t_{cur}} = \Delta + (1, x_{cur}, x_i^2, r_i r_j)$
Return $IS_{i,\lambda,t_{cur}}$

C. Pre-processing features

Because of the variety of instruction types, the features are variance, also fluctuating. Hence, we perform a normalization methodology to normalize the features. Let x denote the set of features extracted of a package on FE module, while $x \in \mathbb{R}^n$, where $n = 100$. On train stage, we calculate the mean (μ) and standard deviation (s) of feature $i^{th}(1 \le i \le n)$. We standardize features by removing the mean then scaling to unit variance. The normalize version of x is calculated as

$$z_i = \frac{x_i - \mu_i}{s_i} (\forall i : 1 \le i \le n) \quad (1)$$

Inspired by image processing success, we propose a novel features viewing: reshape the $1-D$ feature to $2-D$ feature. Since the size of z is 100 as same as size of x, we reshape z to an image with 10 size. We can use CNN layers in a new image instead of using ANN, which has more complex computation and parameters than CNN.

D. Deep learning in DeepNIDS

The classification model of DeepNIDS is basically motivated by the idea of VGG nets architecture [19]. Due to the loss of information in a small input image, we add carefully the convolutional layers follow three rules: (1) Applying convolutional layers except max-pooling layers would not change the size of the feature map size; (2) The layers have an equivalent number of filters if the size of the output feature map are equal; (3) The number of filters doubles if the size of the feature map is halved. The downsampling layers are performed with a stride of 2. The network ends with a fully-connected layer with softmax. Relu activation is applied to almost all hidden layers. The network architecture is illustrated in Fig. 2.

In detail, the input of the network is 10×10 image. Three continuous parts contain layers such as $Conv2D$ with activation functions or max-pooling applied to extract features and reduce the feature map's size. Two fully connected (fc) layers follow by these parts. The soft-max layer is added at the end of the architecture, contains 10 channels (one for normal and nine for nine class which is described in section IV-A). Our proposed model is very lightweight, has only 9,442 trainable parameters.

IV. Evaluation

A. Evaluation Dataset

In order to work towards a lightweight IDS that can run on simple network devices and detect a wide variety of attacks, we base our evaluation on the Kitsune [1]dataset provided by the team Kitsune comes from the Ben-Gurion University of the Negev [15]. The total packet used in our experiments is 21 million packets. Each vector has 115 features and is labeled as a normal connection or an attack. It is built on a nosier network that includes nine attack types with more than 4,800,000 malicious packets.

B. Evaluation Results

Figure 3 presents the accuracy and fallout score of DeepNIDS in detail. As shown in the figure, our accuracy score is exceptionally high in detecting all attack types. The average accuracy is measured at about 98.48%. DeepNIDS could accurately detect five over nine attack types (OS Scan,

[1]The full Kitsune datasets: https://goo.gl/iShM7E.

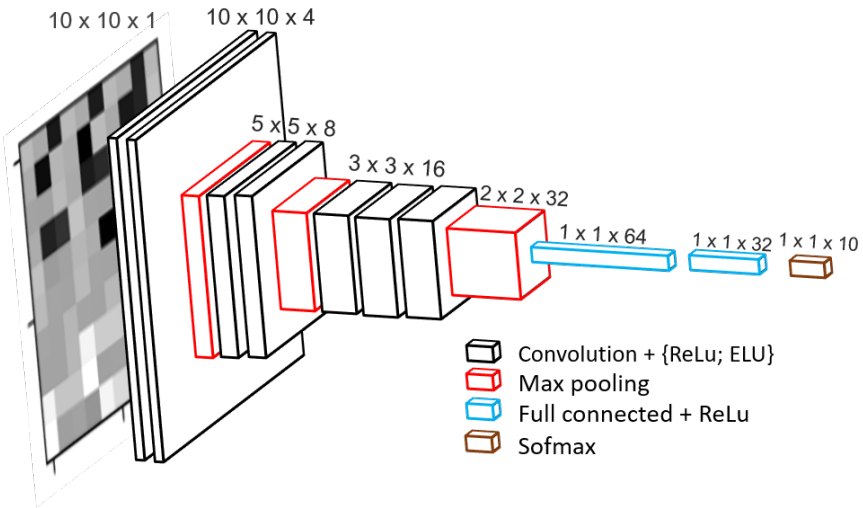

Fig. 2: The architecture of the DeepNIDS's classification model with corresponding kernel size and number of feature map for each convolutional layer and number of unit for each dense layer.

SSDP Flood, SSL Renegotiation, Video Injection, and Active WireTap). The lowest accuracy of our proposal is only 0.92 recorded from the Mira attack. The superior of DeepNIDS is also presented in a very low Fallout (FPR) score that equals 0.004120 on average. This means that we could significantly reduce false alarms of IDS systems that annoy network administrators.

To demonstrate the superior of DeepNIDS comparing with several common NIDSs (such as Gaussian Mixture Models (GMM) [20], Isolation Forest [21],pcStream [22], Suricata [23], Kitsune [15]), we analyze TPR (recall) score and illustrate in Fig. 4. We can see that DeepNIDS outperformance its competitors on several types of attack such as OS Scan, SSL Renegotiation, SYN Dos, Video Injection, Wiretap. Our proposal could correctly detect these attacks (The TPR score equals 1), while this figure of our competitors fails to detect.

We combined 9 types of attacks into a large dataset which have 10 classes (9 types of attacks and 1 class normal) and evaluate our model on that one. Figure 5 shows the performance accuracy score of our model. The average accuracy score is 98.48% in all types of attack. Note that the normal approximate to 99,55% although it has more than 16000000 samples. The detailed detection performance is illustrated in Figure 6.

V. Conclusion

DeepNIDS is a neural network-based NIDS that combines incremental statistics feature extraction framework with a CNN model to efficiently detect several attack types (nine types) with minimal false alarms. The feature extraction component is designed for high-speed extracting statistics feature over a dynamic number of data streams. In addition, a pre-processing feature is added to standardize and reshaped original features to increase the detection performance. The experimental results show that the superior of DeepNIDS with high detection accuracy (reported about 98.48%) while Fallout (FPR) is extremely low. Our future research will focus on deploying our research in a specific context such as a software-defined network (SDN) and hope to apply it to practical contexts.

Acknowledgment

This research is funded by University of Information Technology-Vietnam National University HoChiMinh City under grant number D1-2024-09.

References

[1] Bao-Sam Tran, Thi-Huyen Ho, Thanh-Xuan Do, and Kim-Hung Le. Empirical performance evaluation of machine learning based ddos attack detections. In *Recent Advances in Internet of Things and Machine Learning: real-World Applications*, pages 283–299. Springer, 2022.

[2] Oluwadamilare Harazeem Abdulganiyu, Taha Ait Tchakoucht, and Yakub Kayode Saheed. A systematic literature review for network intrusion detection system (ids). *International Journal of Information Security*, 22(5):1125–1162, 2023.

[3] Kim-Hung Le, Minh-Huy Nguyen, Trong-Dat Tran, and Ngoc-Duan Tran. Imids: An intelligent intrusion detection system against cyber threats in iot. *Electronics*, 11(4):524, 2022.

[4] Nguyen Dat-Thinh, Ho Xuan-Ninh, Le Kim-Hung, et al. Midsiot: A multistage intrusion detection system for internet of things. *Wireless Communications and Mobile Computing*, 2022, 2022.

[5] Xuan-Ha Nguyen, Xuan-Duong Nguyen, Hoang-Hai Huynh, and Kim-Hung Le. Realguard: A lightweight network intrusion detection system for iot gateways. *Sensors*, 22(2):432, 2022.

[6] Zeeshan Ahmad, Adnan Shahid Khan, Cheah Wai Shiang, Johari Abdullah, and Farhan Ahmad. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, page e4150, 2020.

[7] Thanh-Nhan Nguyen, Khanh-Mai Dang, Anh-Duy Tran, and Kim-Hung Le. Towards an attention-based threat detection system for iot networks. In *International Conference on Future Data and Security Engineering*, pages 301–315. Springer, 2022.

[8] Ke Wang and Salvatore J Stolfo. Anomalous payload-based network intrusion detection. In *International Workshop on Recent Advances in Intrusion Detection*, pages 203–222. Springer, 2004.

[9] Xianwei Gao, Chun Shan, Changzhen Hu, Zequn Niu, and Zhen Liu. An adaptive ensemble machine learning model for intrusion detection. *IEEE Access*, 7:82512–82521, 2019.

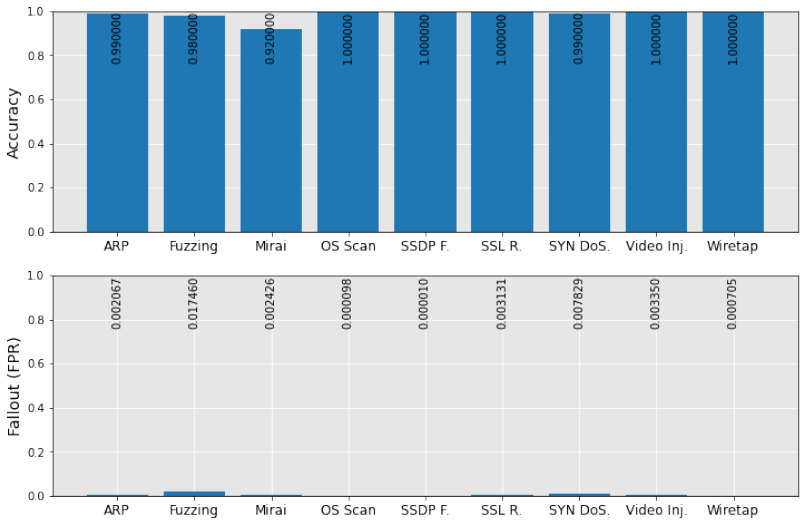

Fig. 3: The accuracy and fallout score of binary classification

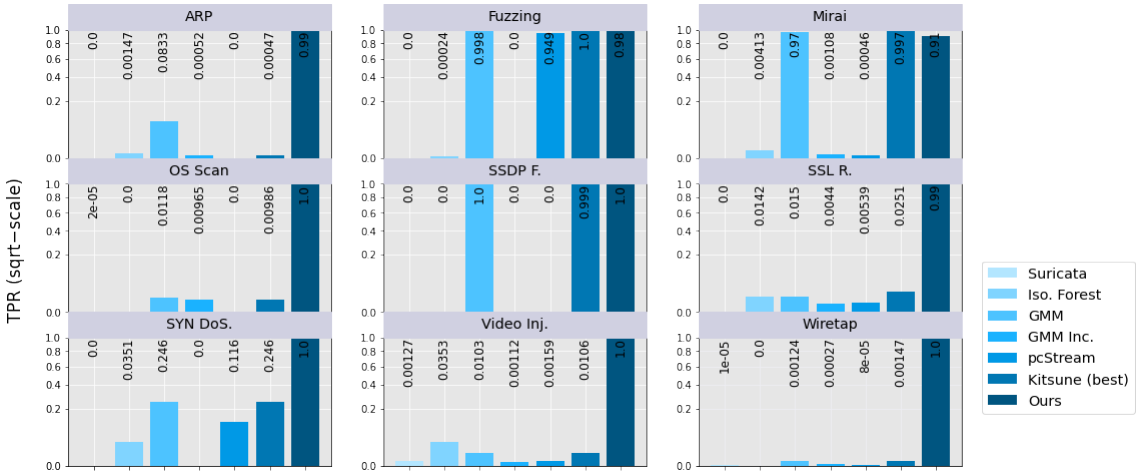

Fig. 4: Compare the results of our model and its competitors.

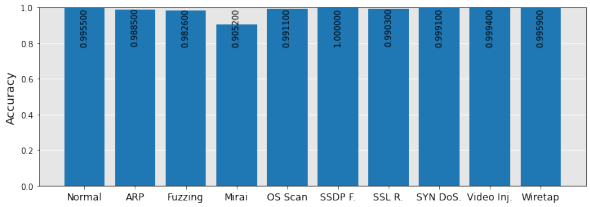

Fig. 5: The accuracy of our model in detecting several attacks.

[10] Xuan-Ha Nguyen, Dat-Thinh Nguyen, and Kim-Hung Le. Benchmarking svm variants for unsupervised intrusion detection system. In *2023 RIVF International Conference on Computing and Communication Technologies (RIVF)*, pages 521–526. IEEE, 2023.

[11] Mohammed Hasan Ali, Bahaa Abbas Dawood Al Mohammed, Alyani Ismail, and Mohamad Fadli Zolkipli. A new intrusion detection system based on fast learning network and particle swarm optimization. *IEEE Access*, 6:20255–20261, 2018.

[12] Dat-Thinh Nguyen and Kim-Hung Le. The robust scheme for intrusion detection system in internet of things. *Internet of Things*, 24:100999, 2023.

[13] Chuanlong Yin, Yuefei Zhu, Jinlong Fei, and Xinzheng He. A deep learning approach for intrusion detection using recurrent neural networks. *Ieee Access*, 5:21954–21961, 2017.

[14] Xuan-Ha Nguyen and Kim-Hung Le. Robust detection of unknown dos/ddos attacks in iot networks using a hybrid learning model. *Internet of Things*, 23:100851, 2023.

[15] Yisroel Mirsky, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai. Kitsune: an ensemble of autoencoders for online network intrusion detection. *arXiv preprint arXiv:1802.09089*, 2018.

[16] Yusuke Sugiyama and Kunio Goto. Design and implementation of a network emulator using virtual network stack. In *7th International Symposium on Operations Research and Its Applications (ISORA'08)*, pages 351–358. Citeseer, 2008.

[17] Eric Leblond and Giuseppe Longo. Suricata idps and its interaction with linux kernel.

[18] Borja Merino. *Instant traffic analysis with Tshark how-to*. Packt Publishing Ltd, 2013.

[19] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[20] M Bahrololum and M Khaleghi. Anomaly intrusion detection system using gaussian mixture model. In *2008 Third International Conference on Convergence and Hybrid Information Technology*, volume 1, pages 1162–1167. IEEE, 2008.

[21] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages

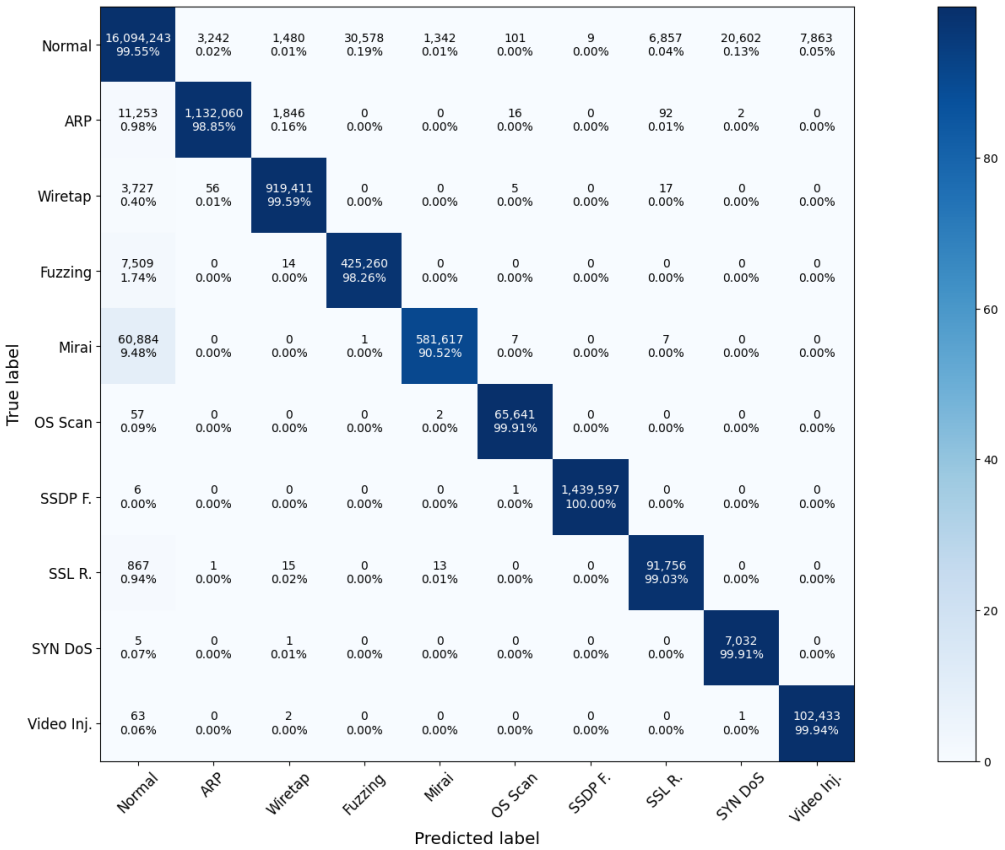

Fig. 6: The confusion matrix of our model in detecting several attacks.

413–422. IEEE, 2008.

[22] Yisroel Mirsky, Tal Halpern, Rishabh Upadhyay, Sivan Toledo, and Yuval Elovici. Enhanced situation space mining for data streams. In *Proceedings of the Symposium on Applied Computing*, pages 842–849, 2017.

[23] Eugene Albin and Neil C Rowe. A realistic experimental comparison of the suricata and snort intrusion-detection systems. In *2012 26th International Conference on Advanced Information Networking and Applications Workshops*, pages 122–127. IEEE, 2012.