Anomaly network intrusion detection system based on NetFlow using machine/deep learning

Touati B. Adli[a], *Salem-Bilal* B. Amokrane[b], *Boban* Z. Pavlović[c], *Mohammad Zouaoui* M. Laidouni[d], *Taki-eddine Ahmed* A. Benyahia[e]

University of Defence in Belgrade, Military Academy, Department of Telecommunications and Informatics, Belgrade, Republic of Serbia

[a] e-mail: adlitouati94@gmail.com, **corresponding author**, ORCID iD: https://orcid.org/0009-0000-2673-6954

[b] e-mail: amokranesalembilal@gmail.com, ORCID iD: https://orcid.org/0009-0009-7588-5708

[c] e-mail: bobanpav@yahoo.com, ORCID iD: https://orcid.org/0000-0002-5476-7894

[d] e-mail: mohammedz.laidouni@gmail.com, ORCID iD: https://orcid.org/0009-0008-6042-0513

[e] e-mail: benyahia.taki@gmail.com, ORCID iD: https://orcid.org/0009-0006-6025-6915

DOI: 10.5937/vojtehg71-46058;https://doi.org/10.5937/vojtehg71-46058

FIELD: computer sciences, telecommunications, cybersecurity

ARTICLE TYPE: original scientific paper

Abstract:

Introduction/purpose: Anomaly detection-based Network Intrusion Detection Systems (NIDSs) have emerged as a valuable tool, particularly in military fields, for protecting networks against cyberattacks, specifically focusing on Netflow data, to identify normal and abnormal patterns. This study investigates the effectiveness of anomaly-based machine learning (ML) and deep learning (DL) models in NIDSs using the publicly available NF-UQ-NIDS dataset, which utilizes Netflow data, with the aim of enhancing network protection.

Methods: The authors Sarhan, M., Layeghy, S., Moustafa, N. and Portmann, M. in the conference paper Big Data Technologies and Applications, in 2021, involve a preprocessing step where 8 features are selected for the training phase out of the 12 available features. Notably, the IP source and destination addresses, as well as their associated ports, are specifically excluded. The novelty of this paper lies in the preprocessing of the excluded features and their inclusion in the training phase, employing various classification ML and DL algorithms such as ExtraTrees, ANN, simple CNN, and VGG16 for binary classification.

Results: The performance of the classification models is evaluated using metrics such as accuracy, recall, etc., which provide a comprehensive analysis of the obtained results. The results show that the ExtraTrees ML model outperforms all other models when using our preprocessing features, achieving a classification accuracy of 99.09%, compared to 97.25% in the reference dataset.

Conclusion: The study demonstrates the effectiveness of anomaly-based ML and DL models in NIDSs using Netflow data.

Key words: Network intrusion detection system (NIDS), Netflow features, Machine/Deep learning, anomaly-based NIDS.

Introduction

As technology progresses, internet networks offer new communication opportunities but also increase vulnerability to intrusions and attacks. This is a significant concern in the military, where technology reliance is growing, and cyber-attacks are becoming more frequent and advanced. To combat these threats, a flexible defense system capable of analyzing large amounts of network traffic is required. Anomaly-based Intrusion Detection System (IDS) offers a valuable methodology for detecting both known and unknown attacks in intrusion detection systems (Van et al., 2017). In the military context, traditional cybersecurity measures such as antivirus software and firewalls are no longer sufficient to protect against advanced threats. To adequately secure military networks against cyber-attacks, an IDS can provide continuous monitoring of the network for potential threats and offer an additional layer of protection (Labonne, 2020).

Network-based Intrusion Detection Systems (NIDSs) are a specific type of IDS that operate at the network layer, analyzing network traffic in real-time for signs of intrusion or malicious activity. In addition to anomaly-based NIDSs, NetFlow is another valuable tool that can be used in the field of NIDSs and attack detection. NetFlow provides network traffic information that can be analyzed to identify patterns and potential threats, allowing for early detection and response to cyber-attacks. By combining the power of anomaly-based NIDSs and NetFlow analysis, military networks can be more effectively protected against a wide range of cyber threats. With the use of advanced technologies such as Deep learning and Machine learning, military networks can become even more resilient against sophisticated attacks.

The paper is structured as follows. Firstly, a comprehensive definition of IDSs, specifically focusing on NIDSs and Anomaly-based NIDSs, is provided. Next, NetFlow is defined, and an overview of the datasets used in the study is presented. The specific ML and DL techniques utilized in the study are presented, along with the results of reproducing the study conducted by (Sarhan et al., 2021) for binary classification. The authors of the original study excluded the IP and port features from the dataset in the training phase, resulting in an 8-features model. The paper introduces a new contribution that involves a preprocessing step applied to the excluded IP and port features, resulting in a 13-features model. This contribution allows us to explore the potential of using these features for improving the performance of the classification in the context of anomaly-based NIDSs with NetFlow data. Then, the NetFlow features for both models were adapted to the input of deep learning techniques by converting the features vector to images.

Finally, the paper presents the results of machine and deep learning for both 8 and 13 feature models and provides recommendations for future research in the field of anomaly-based NIDSs using machine and deep learning techniques with NetFlow data.

Intrusion Detection System (IDS)

Confidentiality, Integrity, and Availability (also known as the CIA triad) are three fundamental concepts of information security. An intrusion or a cyber-attack is defined as all unauthorized activities that compromise one, two, or all of these three components of an information system (Labonne, 2020).

Intrusion detection is the process of monitoring network traffic and computer events to detect unauthorized or malicious activities. An Intrusion Detection System (IDS) is any device or software application that performs this function. An IDS uses its knowledge, including databases, statistics, and artificial intelligence, to transform monitored activities into alerts.

IDSs are sometimes confused with two other security tools: firewalls and Intrusion Prevention Systems (IPSs). Firewalls, IDSs, and IPSs are security tools used to protect network systems but have different methods. Firewalls detect intrusions at the network perimeter and analyze packet headers to filter traffic based on predetermined rules. IDSs monitor network activities and generate alerts, but cannot block suspicious activity on

Adli, T.B. et al, Anomaly network intrusion detection system based on NetFlow using machine/deep learning, pp.941–969

their own. IPSs function like IDSs but can take proactive action to block threats. IPSs automate the process, while firewalls and IDSs require human intervention to process alerts (Labonne, 2020).

Types of IDSs

IDSs can be classified into three categories according to the type of activities that are analyzed: host-based IDSs (HIDS) network-based IDSs (NIDSs), and application-based IDSs (Labonne, 2020; Tufan et al., 2021).

An HIDS is installed on individual computer systems to analyze files, processes, and system logs for suspicious activity. It can detect attacks through indicators like failed logins or high CPU usage. An NIDS analyzes network traffic using sensors placed at various points. It is more scalable and cross-platform than HIDSs, commonly used to protect IT infrastructure. However, a combination of both NIDSs and HIDSs can be used to achieve a higher level of security. For the purpose of this work, the term ”IDS” specifically refers to NIDSs. Application-based IDS is a type of HIDS that focuses on monitoring a specific application.

IDSs can be categorized based on the type of detection method they use. There are three main categories: signature-based detection, anomaly-based detection, and hybrid detection. Signature-based detection compares monitored data with a database of attack signatures, detecting known attacks. This method can only detect known attacks, even with the latest updates. Anomaly detection identifies unknown attacks by flagging deviations from normal behavior. This approach does not require a pre-existing database and can identify unknown attacks. However, it can generate a significant number of false positives. Hybrid detection combines both methods to detect known and unknown attacks, reducing false positives and improving accuracy.

Anomaly-based NIDSs

Anomaly detection plays a critical role in network security, as anomalies can indicate rare but serious events. The network-based NIDS analyzes network-related events, such as traffic volume, IP addresses, service ports, protocol usage, etc. It must detect all types of anomalies in the network. In network-based NIDSs, intrusions typically are referred to as anomalous through continuous observation and modeling of normal behavior in the

networks. However, some anomalous behavior may be normal, highlighting the need for anomaly-based NIDSs to adapt to dynamic network environments with new protocols and updated behaviors. Various techniques, such as statistical-based, knowledge-based, and machine learning-based, have been used in anomaly-based NIDSs, but there are still research challenges to improve their performance and suitability with current network data characteristics (Van et al., 2017). Anomaly detection techniques are the most commonly used IDS detection type and are the most investigated topic in the literature among researchers (Bahlali, 2019).

Our work primarily focuses on researching and implementing anomaly detection in network-based NIDSs, commonly referred to as anomaly detection-based NIDSs. Various ML and DL techniques will be explored to enhance the performance of Anomaly detection-based NIDSs in detecting network traffic anomalies using NetFlow features.

NIDS dataset and NetFlow

NIDS Dataset

Acquiring real-world network data flows is difficult due to security and privacy concerns, which make it challenging to access such data (Sarhan et al., 2022). Due to the challenges of obtaining real-world network data flows, many researchers have developed network testbeds as a means to generate synthetic datasets. These NIDS datasets contain labeled network flows that are made up of certain features extracted from network traffic. The features in a dataset are pre-determined by the authors based on their expertise in the relevant domain and the tools used during the extraction process (Sarhan et al., 2022). In recent years, the most widely used NIDS datasets (Sarhan et al., 2021) that have been released within the past five years are shown in Table 1.

These datasets are highly relevant as they capture modern behavioral network attacks. It is important to note that these datasets differ significantly in terms of their feature sets, and therefore, the information they contain varies considerably (Sarhan et al., 2021). This difference in these datasets makes the evaluation of proposed ML-based NIDSs often unreliable when tested on multiple datasets using their original feature sets (Sarhan et al., 2022).

Adli, T.B. et al, Anomaly network intrusion detection system based on NetFlow using machine/deep learning, pp.941–969

Table 1 – The most relevant NIDS datasets (Sarhan et al., 2021)
Таблица 1 – Наиболее релевантные наборы данных NIDS (Sarhan et al., 2021)
Табела 1 – Најрелевантнији NIDS скупови података (Sarhan et al., 2021)

Dataset	Release year	Number of features
UNSW-NB15	2015	49
BoT-IoT	2018	42
CSE-CIC-IDS2018	2018	75
ToN-IoT	2020	44

NetFlow

NetFlow is a network protocol used for network traffic monitoring and analysis. Compared to pcap format, NetFlow data contains less data, making it easier to collect and process. Additionally, NetFlow is less intrusive to privacy, further enhancing its appeal as a preferred network log format (Cao et al., 2022). Rather than focusing on individual packets, flow monitoring analyzes the flow of traffic, making it a more scalable approach to traffic analysis. This process involves observing packets, exporting flows using protocols like NetFlow and IPFIX, collecting data, and analyzing that data in its entirety (Hofstede et al., 2014). Every flow in NetFlow contains network statistics representing a connection between two hosts. These statistics can be utilized to compute performance metrics and to identify any unusual or abnormal network behavior (Cao et al., 2022).

NetFlow version 9 (NetFlow v9) is the most used version of NetFlow. It is a protocol that enables the collection and export of flow records, providing detailed information about network traffic patterns such as source and destination IP address, source and destination port, protocol, etc. (Cisco. 2011).

NetFlow v9 fields play a crucial role in IDSs by providing valuable information for monitoring, analyzing, and tracking network traffic in real-time, enabling the identification of potential security threats.

Anomaly Detection using Machine learning and Deep learning

Machine learning

Machine learning (ML) has proven to be a highly effective approach to solving diverse problems. One area where machine learning models can be applied is NIDSs, which involves categorizing input data into specific classes, such as ”benign” or ”attack”, as well as identifying various types of attacks (Fosić et al., 2023).

Various machine learning algorithms such as decision trees, Extra-Trees, SVM, etc., are employed for classification. For this study, a supervised machine learning approach was adopted using a NetFlow dataset with uniquely labeled records. Benign traffic was labeled as 0 (class 0), while anomalies or network attacks were labeled as 1 (class 1).

An ExtraTrees ensemble classifier was utilized as it belongs to the ”trees” family and has demonstrated reliable performance in NIDS datasets, allowing for a valid comparison with (Sarhan et al., 2021).

Artificial Neural Network (ANN)

The ANN is a type of machine learning algorithm consisting of interconnected neurons organized as an input layer, a number of hidden layers, and an output layer. Each layer has a specific number of neurons. The information enters the neural network via the input layer, it is processed in the hidden layers and the result can be retrieved in the output layer (Anitha & Arockiam, 2019; Cahyo et al., 2016).

This study implements an ANN to assess its effectiveness in training NetFlow features, aiming to extract meaningful information and improve the accuracy of NIDSs.

Deep Convolutional Neural Networks (CNNs)

Deep learning (DL) is a sub-field of ML that models the learning process using multiple layers of neurons. DL algorithms offer a more automated solution by allowing models to learn feature representations directly from data. This approach is highly effective as a tool for NIDSs, due to its ability to process and learn the data to discover complex features (Rizvi et al., 2023).

In the context of DL, the convolutional neural networks (CNNs), have shown promise in efficiently selecting features and identifying the latent relationships among them (Liu et al., 2019). Inspired by the success of CNNs in image classification tasks, this work aims to apply CNNs to NIDSs leveraging their ability to extract meaningful NetFlow features and classify data accurately (Liu et al., 2019).

Our work involves transforming NetFlow features into images and utilizing two different architectures for classification. The first architecture utilizes a simple CNN structure, while the second is based on the VGG16 model. The comparative analysis of these two architectures will provide insight into the optimal approach for utilizing neural networks in NIDSs.

Evaluation Metrics

In this study, the selection of appropriate performance metrics was given careful consideration to assess the effectiveness of the NIDS model:

1. **Accuracy** $= \dfrac{TP + TN}{TP + FP + TN + FN}$
2. **Recall (Detection Rate or TPR)** $= \dfrac{TP}{TP + FN}$
3. **Precision** $= \dfrac{TP}{TP + FP}$
4. **F1-Score** $= 2 * \dfrac{Recall * Precision}{Recall + Precision}$
5. **AUC (Area Under the Curve)** $= \displaystyle\int_0^1 TPR(FPR)\, d_{FPR}$

   where $TPR(FPR)$ is the function that maps each $FPR = \frac{FP}{FP+TN}$ value to the corresponding TPR.
6. **Score time (μs)** : refers to the duration required for predicting a single test sample.

where prediction **TP** = true positive, **TN** = true negative, **FP** = false positive and **FN** = false negative.

Experiments & results

Hardware and library used

The experimentation phase involved a hardware setup consisting of an 11th Gen Intel(R) Core(TM) i7-11800H processor with 16 virtual CPUs running at a frequency of 2.30GHz. The system was also equipped with 16GB of RAM and an NVIDIA RTX 3060 GPU.

The Python programming language (3.9.16) and the Scikit-learn platform (1.2.1) were utilized for machine/deep learning classification tasks. Additionally, TensorFlow (2.10.1) and Keras (2.10.0) were also used in this study.

NF-UQ-NIDS dataset

The first step of the proposed classification model and methodology is to collect data on traffic flow. The dataset selected for this study is the NF-UQ-NIDS, which is a pre-labeled NetFlow packet containing benign and attack data. This dataset, as published by (Sarhan et al., 2021), was created by merging and converting the four datasets, into NetFlow version 9 format. A total of 12 relevant features were chosen to construct this dataset. The Table 2 shows the descriptions of these features.

The advantage of this dataset is that it offers the advantages of shared datasets and it is more recent than other publicly available datasets which will facilitate a reliable evaluation of proposed learning models across various network settings and attack scenarios.

The NF-UQ-NIDS dataset comprises 11994893 flow records labeled, as either benign or attack. The dataset includes twenty (20) types of attacks, out of which 9208048 (76.77%) are benign flows and 2786845 (23.23%) are attacks. Various types of features, including categorical, numeric (integer, decimal, and binary), and temporal features, are used in the dataset.

Data pre-processing

Data pre-processing involves transforming the raw data into a format that can be used for machine/deep learning tasks. Furthermore, the presence of nominal features or categorical features, and Non-similar scale features can pose a challenge during data pre-processing. To address the first

Table 2 – NetFlow features of NF-UQ-NIDS with brief descriptions
Таблица 2 – Характиристики NetFlow NF-UQ-NIDS с кратким описанием
Табела 2 – NetFlow обележја NF-UQ-NIDS-а са кратким описима

Feature	Description
IPV4_SRC_ADDR	IPv4 source address
IPV4_DST_ADDR	IPv4 destination address
L4_SRC_PORT	IPv4 source port number
L4_DST_PORT	IPv4 destination port number
PROTOCOL	IP protocol identifier byte
TCP_FLAGS	Cumulative of all TCP flags
L7_PROTO	Layer 7 protocol (numeric)
IN_BYTES	Incoming number of bytes
OUT_BYTES	Outgoing number of bytes
IN_PKTS	Incoming number of packets
OUT_PKTS	Outgoing number of packets
FLOW_DURATION_MILLISECONDS	Flow duration in milliseconds

challenge, encoding techniques might be required to transform these features into a suitable format. As for the second issue, normalization may be necessary to ensure that all features take the same range of values.
In the case of the NF-UQ-NIDS dataset, the main issues were identified as nominal features and differences in feature value ranges. To address these issues, One-Hot Encoding and Feature Normalization were used.

The authors of (Sarhan et al., 2021) utilized only eight (8) NetFlow features out of the total twelve (12) features present in the NF-UQ-NIDS dataset. In particular, they excluded the source and destination IP addresses as well as their associated ports during the model training.

However, taking inspiration from (Figueiredo et al., 2023), our main contribution involves the incorporation of the dropped features (IP source/destination and ports) in our study. This inclusion aims to improve the detection of malicious IP addresses and assess the impact compared to the approach adopted by (Sarhan et al., 2021).

Source and destination ports pre-processing

In order to make the dataset suitable for ML and DL, the source and destination ports were merged into a unified feature, preserving the net-

work application's corresponding port for each flow. However, having both ports in the dataset would not be useful for an ML model, since one of the ports is typically a dynamic port that is assigned during the network routing process. These dynamic ports are usually found in the higher range of ports (49152 to 65535), whereas the lower port numbers are reserved for specific network applications. A single feature called "port" was created for each flow, consolidating port numbers between 0 and 4096. The process for converting port numbers is outlined in Algorithm 1. Flows with port numbers above 4096 were mapped to the category 4096, which might limit the NIDS's ability to distinguish between different ports beyond this threshold. Nevertheless, this approach still covers the most frequently used ports in both benign and malicious network traffic (Figueiredo et al., 2023).

Algorithm 1 Port number conversion

for Row in Dataset **do**
  $sp \leftarrow SourcePort$
  $dp \leftarrow DestinationPort$
  if $sp \leq dp$ **then**
    $Port \leftarrow sp$
  else
    $Port \leftarrow dp$
  end if
  if $Port \geq 4096$ **then**
    $Port \leftarrow 4096$
  end if
  $Row \leftarrow Row + Port$
end for

Source and destination IP pre-processing

The inclusion of source and destination IP addresses and ports in the training phase is a key aspect of this study. An IP address served as an identifier for each system in the network; it is hard to translate into a feature for ML. Two of the most common approaches to solve this problem are (Figueiredo et al., 2023): (a) removing these features altogether as in (Sarhan et al., 2021) which results in the loss of valuable contexts, such as the general network location, or (b) using a dictionary to translate the

Adli, T.B. et al, Anomaly network intrusion detection system based on NetFlow using machine/deep learning, pp.941–969

IP addresses to a number, which can be reversed in the end to identify a malicious IP address. Although a dictionary can effectively map individual systems and detect patterns such as traffic originating from the same IP address, this method may not work well in a different network context due to high misclassification rates and the increasing dataset size.

To strike a balance between the two prevalent options, a particular approach was applied, involving the conversion of each IP address to a binary feature denoting either Internal or External. (Figueiredo et al., 2023). The assignment of the ”Internal” label was based on IP addresses belonging to a private address space (starting with ”192.168.”, ”172.16.”, or ”10.”), while IP addresses outside this range were labeled as ”External” (Algorithm 2). Since Internal and External are fundamental characteristics of every network flow, this method yields more contextual information about the network compared to simply removing the source and destination IPs. Moreover, this feature is context-independent, making it easy to apply the model to different networks.

Algorithm 2 IP address conversion

```
for Row in Dataset do
    if IP starts with ”192.168.” or ”172.16.” or ”10.” then
        IP ← Internal
    else
        IP ← External
    end if
end for
```

After mapping the IP source and destination addresses into the categories ”Internal” and ”External,” it is necessary to employ data encoding techniques to convert these categorical features into numerical representations.

Encoding data is the process of transforming some input to numbers, usually in a way that is reversible and allows the translation between the resulting output and the original input (Figueiredo et al., 2023).

Assigning a unique number to each category when encoding categorical features can result in an ordinal encoding which may mislead ML models. As such, a binarization technique called One-Hot Encoding was used. This technique converts each category of a specific feature into a new binary

feature with the value one (1) meaning that it belongs to this category and zero (0) otherwise.

After applying this technique to the mapped IP addresses, two new features are obtained for each IP address, as illustrated in Figure 1.

IP Address
External
Internal
Internal
External

One-Hot Encoding →

External IP Address	Internal IP Address
1	0
0	1
0	1
1	0

Figure 1 – One-Hot Encoding of an IP address
Рис. 1 − Горячее кодирование IP-адреса
Слика 1 – One-Hot кодирање ИП адресе

Data normalisation

The normalization step is important for the training process since the difference in the feature scales can cause problems during the training. With the normalization, each feature would have an equal impact on the model prediction results.

The Min-Max normalization technique was utilized to scale all values in the dataset between 0 and 1. This technique performs a linear transformation on the original data. The advantage of Min-Max normalization is that it preserves the relationships among the original data values (Labonne, 2020; Bahlali, 2019). The normalized feature is given by:

$$\hat{x_i} = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}. \quad (1)$$

where x_i and $\hat{x_i}$ denote the original and the normalized feature value, respectively.

1D NetFlow data to 2D NetFlow images

In this work, two different approaches were explored. The first approach involved constructing an image directly from the features. The second approach involved constructing the image by building a square surrounding correlation matrix (SC matrix), as utilized in (Liu et al., 2019)

First approach: reshaping features image

For the 8 features, constructing an image with a size of 3x3 was insufficient. To address this issue, zeros were added to the missing pixels, as shown in Figure 2.

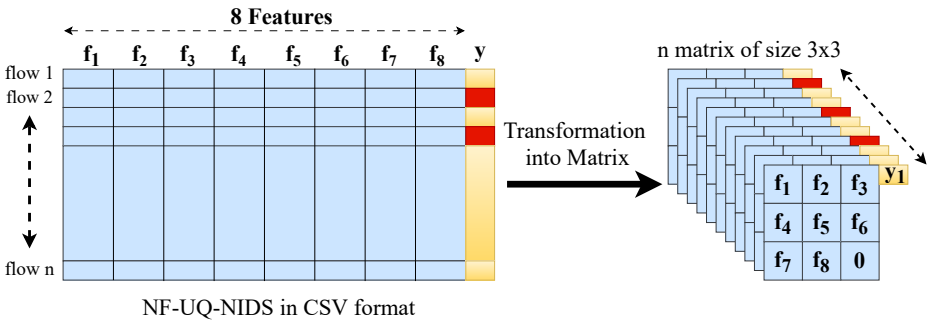

Figure 2 – NetFlow to matrix transformation by reshaping
Рис. 2 − Преобразование NetFlow в матрицу путем изменения формы
Слика 2 – Преобликовање NetFlow у матричну трансформацију

For the 13-feature scenario, the Recursive Feature Elimination (RFE) technique was employed to select the nine most significant features for the analysis. Following this, a simple reshaping technique was applied to transform the data into images of size 3x3.

Second approach: using SC matrix

Based on the approach in (Liu et al., 2019) which proposes a localization technique that utilizes the correlation matrix to process NetFlow data, the correlation matrix is used to evaluate the correlations among the features in NetFlow data. In this study, the top-k (k=8) highly correlated features $X_{i0}, X_{i1}, , X_{i7}$, were selected from the Netflow features $X_0, X_1, ..., X_n$, for each numeric feature X_i to construct a square SC matrix. For each row of NetFlow data, an image is generated by concatenating the SC matrices of all features. An example of NetFlow images generated using this approach is shown in Figure 3.

This technique provides a powerful approach for extracting meaningful information from NetFlow data and improving the accuracy of NIDSs. In the case of 8 features, the issue of a missing value for the ninth pixel was resolved by substituting it with zero, maintaining the size of the image at

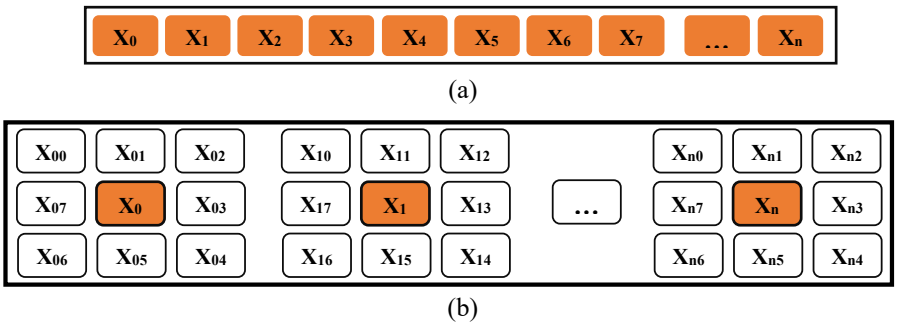

Figure 3 – Transformation of 1D NetFlow features to 2D NetFlow
(a) NetFlow features (b) 2D image
Рис. 3 – Преобразование 1D NetFlow в 2D NetFlow
(a) Атрибут NetFlow (b) 2D-изображение
Слика 3 – Трансформација 1D NetFlow обележје у 2D NetFlow
(a) NetFlow обележје (b) слика 2D

3x27 pixels. Similarly, for 13 features, images with dimensions of 3x39 pixels were generated.

Evaluation

The attack detection performance of the NetFlow datasets NF-UQ-NIDS was evaluated, reproducing the results of the authors (Sarhan et al., 2021) for binary classification. The evaluation was conducted using 8 features, and the obtained results were compared with our results using 13 features. The additional features were obtained through data pre-processing, including the IP source and destination and their corresponding ports.

To evaluate the performance of the proposed ML and DL models on the NF-UQ-NIDS dataset, an ExtraTrees ensemble classifier was selected based on its demonstrated success in achieving reliable performance on NIDS datasets (Sarhan et al., 2021). Additionally, a simple ANN model was implemented as an ML classifier. The DL model utilized in this study employed a simple CNN architecture and incorporated transfer learning from the VGG16 model.

ExtraTrees classifier

An ExtraTrees ensemble classifier consisting of 50 randomized decision tree estimators was applied using the sklearn library in Python *Extra-*

TreesClassifier(n_estimators=50, class_weight="balanced"). The option "balanced" is set due to the imbalanced dataset. To ensure the datasets are reliably evaluated, five-fold cross-validation is conducted, and average metrics such as accuracy, Area Under the Curve (AUC), precision, recall, F1-score, and the time required to predict a single test sample in microseconds (μs) are calculted using the *sklearn* library. The results are shown in the Table 3.

Table 3 – Binary classification results using ExtraTrees ML
Таблица 3 - Результаты бинарной классификации с использованием ExtraTrees ML
Табела 3 - Резултати бинарне класификације коришћењем ExtraTrees ML

Metrics	8 features	13 features
Accuracy	0.9744	**0.9909**
AUC	0.9917	**0.9940**
Recall	0.9672	**0.9861**
Precision	0.9632	**0.9884**
F1-score	0.9459	**0.9804**
Score time (μs)	5.87	**5.03**

The results show that the 13-feature model performs better than the 8-feature model across all evaluated metrics.

The 13-feature model has an accuracy of 0.9909, which is higher than the 8-feature model's accuracy of 0.9744. Additionally, the 13-feature model has a higher AUC (0.9940) than the 8-feature model (0.9917), indicating better overall performance in distinguishing between the two classes. The 13-feature model also shows better recall (0.9861) and precision (0.9884) than the 8-feature model (0.9672, 0.9632, respectively), which means it is able to correctly identify more positive samples (higher recall) and make fewer false positive predictions (higher precision) than the 8-feature model. The F1-score is higher for the 13-feature model (0.9804) than for the 8-feature model (0.9459), indicating that it has a more optimal trade-off between precision and recall.

The 13-feature model has a slightly lower time to predict a single test sample than the 8-feature model, with 5.03 μs for the 13-feature model and 5.87 μs for the 8-feature model.

Our results indicate that the additional features provide valuable information that improves the model's ability to distinguish between benign and attack traffic, and ultimately improve the model's attack detection performance.

ANN model

The summary in Figure 4 provides a detailed description of the proposed ANN model architecture. The ANN is based on an input layer with 8 or 13 inputs for both 8 and 13 feature models.

Model: "sequential"

Layer (type)	Output Shape	Param #
flatten (Flatten)	(None, 8)	0
dense (Dense)	(None, 256)	2304
dense_1 (Dense)	(None, 256)	65792
dense_2 (Dense)	(None, 256)	65792
dense_3 (Dense)	(None, 256)	65792
dense_4 (Dense)	(None, 256)	65792
dense_5 (Dense)	(None, 20)	514
softmax (Softmax)	(None, 20)	0

Total params: 265,986
Trainable params: 265,986
Non-trainable params: 0

Figure 4 – ANN model summary for the 8 features input
Рис. 4 − Краткое описание модели ANN для ввода 8 функций
Слика 4 – Резиме модела ANN за унос од 8 обележја

The evaluation of the ANN model was conducted using a specific configuration, which included the following parameters: *Adamax* optimizer, the learning rate of 0.001, *categorical cross-entropy* loss function, and 30 epochs of training.

The results shown in Table 4 indicate that the addition of four features has significantly enhanced the model's performance. Both models show promising results, with the 8-feature model achieving an accuracy of 0.9285, and the 13-feature model achieving an accuracy of 0.9673. Furthermore, the AUC increased from 0.9806 to 0.9939, indicating the model's improved ability to distinguish between attack and benign samples. The recall increased from 0.8103 to 0.8810, and the precision improved from

Table 4 – Binary classification results using ANN machine learning
Таблица 4 – Результаты бинарной классификации с использованием ANN ML
Табела 4 – Резултати бинарне класификације коришћењем ANN ML

Metrics	8 features	13 features
Accuracy	0.9285	**0.9673**
AUC	0.9806	**0.9939**
Recall	0.8103	**0.8810**
Precision	0.8729	**0.9757**
F1-score	0.8404	**0.9259**
Score time (μs)	115.32	**100.38**

0.8729 to 0.9757. The F1-score also increased from 0.8404 to 0.9259, indicating an overall improvement in performance. Additionally, the prediction time slightly decreased, which is a positive outcome.

Discussion: ExtraTrees Vs ANN

The ExtraTrees with 13 features outperformed the 8-feature model from (Sarhan et al., 2021) with an accuracy of 0.9909 compared to 0.9744.

The ExtraTrees for 8 and 13 features, outperformed the ANN in all the evaluation metrics. However, it is noteworthy that the ANN still achieved a high level of accuracy and showed significant improvement after incorporating the four additional features. The ExtraTrees show better accuracy than the ANN for both the 8 and 13 features. Moreover, both models performed well in terms of the AUC, indicating their ability to distinguish between attack and benign flow. In terms of recall, the ExtraTrees outperformed the ANN for both the 8 and 13-feature models, with consistently better performance observed for the 13-feature model. When it comes to precision, the ExtraTrees using 13 features exhibited better precision metrics. The ExtraTrees using 13 features achieved the highest F1-score, surpassing all other models in performance.

The ExtraTrees model demonstrated a slightly faster score time compared to the ANN model for both the 8 and 13 features.

The results obtained from both the ExtraTrees and ANN models indicate that incorporating the excluded features was more effective in detecting attacks compared to utilizing only 8 features.

CNN model based-NIDS

The process for training a CNN model on NetFlow data involves two key steps: 1) converting 1D NetFlow features into 2D NetFlow images, and 2) inputting the NetFlow image data into the CNN model using both direct training and transfer learning techniques. Transforming 1D NetFlow features into 2D images enables the utilization of the powerful image classification capabilities of CNNs, leading to improved accuracy in NIDSs (Liu et al., 2019).

In this study, two different CNN models were employed. The first model is a simple CNN composed of three convolutional layers. The second model utilized is the widely recognized VGG16, known for its significant contributions to CNN models.

Simple CNN

The summary in Figure 5 provides a detailed description of our simple CNN model architecture, including the arrangement and specifications of each layer. The proposed CNN model is based on an input layer with an input size of (32,32,1).

For the evaluation of the simple CNN model, a specific configuration was employed, incorporating the following parameters: the *Adamax* optimizer, a learning rate of 0.001, the use of *categorical cross-entropy* as the loss function, and training for a total of 30 epochs. The performance results of the proposed simple CNN model for both cases with 8 and 13 features are presented in Table 5.

Based on the obtained results, for the Simple CNN model, using the 13 features with an image size of 3x3 provides the best overall performance, as it achieved the highest accuracy (0.9884), AUC (0.9970), recall (0.9648), Precision (0.9850) and F1-score (0.9747) compared to the other models, suggesting that the additional features contribute valuable information for the classification of the attacks. Accuracy increased from 0.9508 to 0.9884 going from 8 to 13 features with a 3x3 image. It increased further to 0.9686 with a 3x39 image compared to 0.9657 for 3x27 image.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 32, 32, 16)	160
max_pooling2d (MaxPooling2D)	(None, 16, 16, 16)	0
conv2d_1 (Conv2D)	(None, 16, 16, 32)	4640
max_pooling2d_1 (MaxPooling 2D)	(None, 8, 8, 32)	0
conv2d_2 (Conv2D)	(None, 8, 8, 64)	18496
max_pooling2d_2 (MaxPooling 2D)	(None, 4, 4, 64)	0
flatten (Flatten)	(None, 1024)	0
dense (Dense)	(None, 128)	131200
dense_1 (Dense)	(None, 2)	258

Total params: 154,754
Trainable params: 154,754
Non-trainable params: 0

Figure 5 – A Simple CNN model summary
Рис. 5 − Краткое описание модели Simple CNN
Слика 5 – Једноставан резиме CNN модела

The score time tends to increase as the image size grows larger. Among the different image sizes evaluated, the 3x39 image size exhibited the highest score time (178.34μs). On the other hand, using 8 features with 3x3 images demonstrated the fastest score time, with a minimal difference compared to the 13 features using the same 3x3 image size.

Among the tested configurations, it seems that utilizing 13 features with a 3x3 image size offers the optimal balance of accuracy, AUC, recall, precision, and F1-score. The inclusion of the four features, combined with the reshaping approach, leads to enhanced performance for NIDSs.

VGG16 model

VGG16 has a relatively straightforward architecture compared to other deep learning models. VGG16 has a hierarchical structure that gradually increases the complexity of feature extraction, allowing it to capture both low-level and high-level features in images. (Van et al., 2017).

In the study, a 32x32x3 input layer is utilized. Two strategies are employed: transfer learning with a pre-trained model on the ImageNet dataset and training the VGG16 model from-scratch. The structure of the adapted

Table 5 – Binary classification results using a simple CNN model
Таблица 5 – Результаты бинарной классификации с использованием простой модели CNN
Табела 5 – Резултати бинарне класификације коришћењем једноставног CNN модела

Metrics	8 features		13 features	
	Image 3x3	Image 3x27	Image 3x3	Image 3x39
Accuracy	0.9508	**0.9657**	**0.9884**	0.9686
AUC	0.9889	**0.9889**	**0.9970**	0.9932
Recall	0.8933	**0.8989**	**0.9648**	0.8882
Precision	0.8947	**0.9508**	**0.9850**	0.9743
F1-score	0.8940	**0.9241**	**0.9747**	0.9293
Score time (μs)	**80.89**	108.6 3	81.13	178.34

VGG16 model, specifically designed for the binary classification between benign and attack instances, is shown in Figure 6.

Model: "Adapted VGG16"

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 1, 1, 512)	14714688
flatten (Flatten)	(None, 512)	0
dense (Dense)	(None, 256)	131328
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 2)	514

Total params: 14,846,530
Trainable params: 131,842
Non-trainable params: 14,714,866

Figure 6 – Adapted VGG16 model summary
Рис. 6 − Краткое описание адаптированной модели VGG16
Слика 6 – Прилагођени резиме VGG16 модела

The evaluation of the adapted VGG16 model involved a specific configuration with the following parameters: *Adamax* optimizer, a learning rate of 0.001, *categorical cross-entropy* as the loss function, and training for 30

epochs. The performances of the VGG16 model trained on both 8 and 13 features are presented in Table 6.

Table 6 – Binary classification results using the VGG16 model
Таблица 6 – Результаты бинарной классификации с использованием модели VGG16
Табела 6 – Резултати бинарне класификације коришћењем VGG16 модела

Metrics	8 features				13 features			
	Image 3x27		Image 3x3		Image 3x39		Image 3x3	
	Pre Trained	From scratch	Pre Trained	From scratch	Pre Trained	From scratch	Pre Trained	From scratch
Accuracy	0.9026	**0.9505**	0.9012	**0.9750**	0.9536	**0.9670**	0.9532	**0.9665**
AUC	0.9609	**0.9875**	0.9577	**0.9924**	0.9885	**0.9925**	0.9896	**0.9930**
Recall	0.7739	**0.8911**	0.7729	**0.9162**	0.8668	**0.9199**	0.8693	**0.8855**
Precision	0.8002	**0.8953**	0.7958	**0.9749**	0.9285	**0.9369**	0.9248	**0.9674**
F1-score	0.7868	**0.8932**	0.7842	**0.9446**	0.8966	**0.9283**	0.8962	**0.9246**
Score time (μs)	**346.32**	1062.95	**363.30**	568.34	**431.23**	876.50	**342.89**	888.771

Comparing pre-trained and from-scratch models, the results suggest that the from-scratch models tend to achieve superior performance in terms of accuracy, AUC, recall, and F1-score. However, the pre-trained models have lower score time compared to the from-scratch models.

For the 8 feature, the VGG16 model trained from-scratch with a 3x3 image size achieves the highest accuracy (0.9750), AUC (0.9924), Recall (0.9162), Precision (0.9749) and F1-score (0.9446). In the case of the 13 features trained from-scratch, the results show that both image sizes produce comparable outcomes, particularly in terms of accuracy and AUC.

In conclusion, the from-scratch VGG16 models display superior performance in terms of evaluation metrics, while the pre-trained models excel in computational efficiency. This can be attributed to the fact that pre-trained models are not optimized for the specific task of network intrusion detection, as the VGG16 model was originally pre-trained on the ImageNet dataset, which has a different set of features.

Overall, the VGG16 model gives very good results for this network intrusion detection task, with accuracy and AUC over 0.95. This shows that the model has learned the patterns in the NetFlow data very well for detecting network intrusions.

Result summary and discussion

This part presents an overview of the results obtained from various tests of anomaly-based NIDSs. The results show that the ExtraTrees model outperformed all other models for 13 feature inputs. It also showed relatively high recall and precision, which indicates a good balance between identifying true positives and avoiding false positives. Moreover, it had the lowest prediction time (5.03 µs) among all models, which makes it a good choice for real-time applications. Additionally, using ExtraTrees with 13 features has shown better results than the one of (Sarhan et al., 2021) with the highest accuracy of 0.9909.

The ANN model also demonstrates a good performance with 13 features, but its score time is significantly higher than the ExtraTrees model, at 100.38 µs.

As for the deep learning models, the VGG16 from-scratch outperformed the pre-trained model in most cases, especially in terms of precision and recall. However, it had a significantly higher prediction time, which could be a disadvantage in some real-time applications. Regarding the proposed simple CNN model, it showed relatively good performance, especially for image 3x3 in both 8 and 13 features input. However, its performance was not as good as the ExtraTrees but is better than VGG16 models, and its prediction time was higher than ExtraTrees but lower than VGG16.

In conclusion, among the tested models, the ExtraTrees model utilizing 13 features demonstrates superior performance in terms of accuracy, AUC, F1-score, and score time. However, for the DL models, the simple CNN model provides better performance compared to the VGG16 models.

Conclusion

This study presents ML and DL models based-NIDSs using Netflow features. The ML models utilized are ExtraTrees and ANN, while the DL models employed include VGG16 and a simple CNN model proposed in this study. The models were trained on the NF-UQ-NIDS dataset.

Adli, T.B. et al, Anomaly network intrusion detection system based on NetFlow using machine/deep learning, pp.941–969

Our main contribution is the inclusion of the excluded features in the binary classification process, based on the work by (Sarhan et al., 2021). This enhancement aims to improve the performance of the binary classification model in NIDSs to classify the flow data as either ”attack” or ”benign”, resulting in two training datasets: one with the original 8 features and another with the enriched 13 features by using the technique proposed in (Figueiredo et al., 2023). Additionally, both the proposed ML and DL models were evaluated using appropriate performance metrics such as accuracy, recall, precision, and F1-score.

The results demonstrate that the ExtraTrees model outperformed other methods in binary classification using the 13 features and shows better results compared to the one presented in (Sarhan et al., 2021).

These findings suggest that the inclusion of the four excluded features in (Sarhan et al., 2021) contributed to the improved performance of the classifier. The results of this study have practical implications for the development of more efficient and accurate NIDS systems for detecting network attacks.

In future work, the second version of the NF-UQ-NIDS dataset, known as NF-UQ-NIDS-v2, proposed in (Sarhan et al., 2022), will be considered for further investigation. This dataset is advantageous as it contains a larger number of records, totaling 75987976, and includes 43 features. Training machine learning and deep learning models on this dataset can improve their accuracy and robustness due to a larger number of features. This dataset has the potential to enhance the performance of NIDS systems in detecting network attacks.

References

Anitha, A.A. & Arockiam, L. 2019. ANNIDS: Artificial Neural Network based Intrusion Detection System for Internet of Things. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 8(11), pp. 2583–2588. Available at: https://doi.org/10.35940/ijitee.K1875.0981119.

Bahlali, A.R. 2019. *Anomaly-Based Network Intrusion Detection System: A Machine Learning Approach*. Ma thesis, Biskra, Algeria: University of Mohamed Khider, Faculty of Exact, Natural and Life Sciences, Computer Science Departement. Available at: https://doi.org/10.13140/RG.2.2.29553.84325.

Cahyo, A.N., Hidayat, R. & Adhipta, D. 2016. Performance comparison of intrusion detection system based anomaly detection using artificial neural

network and support vector machine. *AIP Conference Proceedings*, 1755(1, art.number:070011), pp. 1–7. Available at: https://doi.org/10.3969/j.issn.1002-6819.2015.01.028.

Cao, C., Panichella, A., Verwer, S., Blaise, A. & Rebecchi, F. 2022. ENCODE: Encoding NetFlows for State-Machine Learning. *arXiv:2207.03890*. Available at: https://doi.org/10.48550/arXiv.2207.03890.

Cisco. 2011. *NetFlow Version 9 Flow-Record Format* [online]. Available at: https://www.cisco.com/en/US/technologies/tk648/tk362/technologies_white_paper09186a00800a3db9.html [Accessed: 10 August 2023].

Figueiredo, J., Serrão, C. & de Almeida, A.M. 2023. Deep Learning Model Transposition for Network Intrusion Detection Systems. *Electronics*, 12(2, art.number:293). Available at: https://doi.org/10.3390/electronics12020293.

Fosić, I., Žagar, D., Grgić, K. & Križanović, V. 2023. Anomaly detection in NetFlow network traffic using supervised machine learning algorithms. *Journal of Industrial Information Integration*, 33, art.number:100466. Available at: https://doi.org/10.1016/j.jii.2023.100466.

Hofstede, R., Čeleda, P., Trammell, B., Drago, I., Sadre, R., Sperotto, A. & Pras, A. 2014. Flow Monitoring Explained: From Packet Capture to Data Analysis With NetFlow and IPFIX. *IEEE Communications Surveys and Tutorials*, 16(4), pp. 2037–2064. Available at: https://doi.org/10.1109/COMST.2014.2321898.

Labonne, M. 2020. *Anomaly-based network intrusion detection using machine learning*. Ph.D. thesis, Institut polytechnique de Paris. [online]. Available at: https://theses.hal.science/tel-02988296 [Accessed: 10 August 2023].

Liu, X., Tang, Z. & Yang, B. 2019. Predicting Network Attacks with CNN by Constructing Images from NetFlow Data. In: *2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*. Washington, DC, USA, pp.61–66, May 27-29. Available at: https://doi.org/10.1109/BigDataSecurity-HPSC-IDS.2019.00022.

Rizvi, S., Scanlon, M., McGibney, J. & Sheppard, J. 2023. Deep Learning Based Network Intrusion Detection System for Resource-Constrained Environments. In: *Goel, S., Gladyshev, P., Nikolay, A., Markowsky, G. & Johnson, D. (Eds.) Digital Forensics and Cyber Crime. ICDF2C 2022. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*. Boston, MA, 508, pp.355–367, November 16-18. Cham: Springer. Available at: https://doi.org/10.1007/978-3-031-36574-4_21.

Sarhan, M., Layeghy, S., Moustafa, N. & Portmann, M. 2021. NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems. In: *Deze, Z., Huang, H., Hou, R., Rho, S. & Chilamkurti, N. (Eds.) Big Data Technologies and Applications. BDTA WiCON 2020 2020. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*. Virtual Event, 371, pp.117–135, December 11. Cham: Springer. Available at: https://doi.org/10.1007/978-3-030-72802-1_9.

Sarhan, M., Layeghy, S. & Portmann, M. 2022. Towards a Standard Feature Set for Network Intrusion Detection System Datasets. *Mobile Networks and Applications*, 27, pp. 357–370. Available at: https://doi.org/10.1007/s11036-021-01843-0.

Tufan, E., Tezcan, C. & Acartürk, C. 2021. Anomaly-Based Intrusion Detection by Machine Learning: A Case Study on Probing Attacks to an Institutional Network. *IEEE Access*, 9, pp. 50078–50092. Available at: https://doi.org/10.1109/ACCESS.2021.3068961.

Van, N.T., Thinh, T.N. & Sach, L.T. 2017. An anomaly-based network intrusion detection system using Deep learning. In: *2017 International Conference on System Science and Engineering (ICSSE)*. Ho Chi Minh City, Vietnam, pp.210-214, September 11. Available at: https://doi.org/10.1109/ICSSE.2017.8030867.

Аномальная система обнаружения вторжений в сеть на основе NetFlow с использованием машинного/глубокого обучения

Туати Б. Адли, **корреспондент**, *Салем-Билал* Б. Амокране, *Бобан* З. Павлович, *Мохамед Зуауи* М. Лаидуни, *Таки-эддине Ахмед* А. Беняхия

Университет обороны в г. Белград, Военная академия, Департамент телекоммуникаций и информатики, г. Белград, Республика Сербия

РУБРИКА ГРНТИ: 20.23.25 Информационные системы с базами знаний,
49.33.29 Сети связи

ВИД СТАТЬИ: оригинальная научная статья

Резюме:

Введение/цель: Системы обнаружения аномалий на основе сетевого обнаружения вторжений (NIDS) стали ценным инструментом, особенно в области военного применения, для защиты сетей от кибератак, с фокусом на данных Netflow для идентификации нормальных и аномальных паттернов. В данной статье исследуется эффективность моделей машинного обучения (ML) и глубокого обучения (DL) на основе аномалий в NIDS с использованием общедоступного набора данных NF-UQ-NIDS, использующего данные Netflow, с целью повышения защиты сети.

Методы: Авторы Sarhan, M., Layeghy, S., Moustafa, N. и Portmann, M. в своем докладе на конференции «Big Data Technologies and Applications», проведенной в 2021 году использовали этап предобработки, на котором выбираются 8 признаков для фазы обучения из доступных 12 признаков. Были исключены IP-адреса исходных и целевых узлов, а также связанные с ними порты. Новизна данной статьи заключается во включении всех доступных функций на этапе обучения с использованием различных алгоритмов классификации ML и DL, таких как ExtraTrees, ANN, простая модель CNN и VGG16 при бинарной классификации.

Результаты: Производительность моделей классификации оценивается с использованием метрик, таких как точность, полнота и т. д., что обеспечивает комплексный анализ полученных результатов. Результаты показывают, что модель ML ExtraTrees превосходит все остальные модели при использовании признаков на этапе предобработки и достигает 99,09% точности классификации, по сравнению с 97,25% в эталонном наборе данных.

Выводы: Исследование показало высокую эффективность различных алгоритмов классификации моделей ML и DL в NIDS с использованием базы данных Netflow.

Ключевые слова: сетевые системы обнаружения вторжений (NIDS), характеристики Netflow, машинное/глубокое обучение, аномальный NIDS.

Систем откривања аномалија у мрежи на бази NetFLow протокола применом машинског/дубоког учења

Туати Б. Адли, **аутор за преписку**, *Салем-Билал* Б. Амокране, *Бобан* З. Павловић, *Мохамед Зуауи* М. Лаидуни, *Таки-еддине Ахмед* А. Бенјахијаг

Универзитет одбране у Београду, Војна академија, Катедра телекомуникација и информатике, Београд, Република Србија

ОБЛАСТ: рачунарске науке, телекомуникације, сајбер безбедност

КАТЕГОРИЈА (ТИП) ЧЛАНКА: оригинални научни рад

Сажетак:

Увод/циљ: Проналажење мрежних аномалија, базирано на примени система за детекцију злонамерних упада у мрежу (NIDS), представља изузетно вредан алат, посебно у војним применама, за заштиту мрежа од сајбер напада, са посебним фокусом на Netflow податке ради идентификације нормалних и инцидентних ситуација. У овом раду је спроведено истраживање које анализира ефикасност у борби против аномалија применом модела машинског учења (ML) и дубоког учења (DL) у NIDS-у коришћењем јавно доступне базе података NF-UQ-NIDS која садржи Netflow податке, ради побољшања заштите мреже.

Методе: Аутори Sarhan, M., Layeghy, S., Moustafa, N. и Portmann, M. у раду са конференције Big Data Technologies and Applications, из 2021. године, користили су предобраду у којој се 8 обележја издваја за фазу тренинга од укупно 12 доступних обележја. Посебно су изузете изворне и одредишне IP адресе, као и њихови припадајући портови. Главни допринос овог рада односи се на укључивање свих доступних обележја у фазу тренинга, коришћењем различитих алгоритама класификације ML и DL, као што су ExtraTrees, ANN, једноставни CNN и VGG16 за бинарну класификацију.

Резултати: Перформансе анализираних класификационих модела евалуиране су помоћу неколико метрика (тачност, одзив, прецизност и друго), чиме је омогућена свеобухватна компарација добијених резултата. У завршној анализи резултати показују да ML модел ExtraTrees надмашује све остале моделе користећи предложену предобраду свих доступних обележја, постигавши тачност класификације од 99,09%, у поређењу са 97,25% у референтном скупу података.

Закључак: Спроведено истраживање анализира ефикасност различитих алгоритама класификације ML и DL модела у NIDS-у коришћењем базе Netflow.

Кључне речи: систем откривања упада у мрежу (NIDS), Netflow обележја, машинско учење (ML), дубоко учење (DL).

Paper received on / Дата получения работы / Датум пријема чланка: 18.08.2023.
Manuscript corrections submitted on / Дата получения исправленной версии работы / Датум достављања исправки рукописа: 01.12.2023.
Paper accepted for publishing on / Дата окончательного согласования работы / Датум коначног прихватања чланка за објављивање: 02.12.2023.