Supervised Machine Learning and Detection of Unknown Attacks: An Empirical Evaluation

Miguel S. Rocha[1], Gustavo D. G. Bernardo[1], Luan Mundim[1],
Bruno B. Zarpelão[2], and Rodrigo S. Miani[1](✉)

[1] Federal University of Uberlândia, Uberlândia, Brazil
{miguelsr,gustavo.bernardo,luanmundim,miani}@ufu.br
[2] State University of Londrina, Londrina, Brazil
brunozarpelao@uel.br

Abstract. Intrusion Detection Systems (IDS) have become one of the organizations' most important security controls due to their ability to detect cyberattacks while inspecting network traffic. During the last decade, IDS proposals have increasingly used machine learning techniques (ML-based IDS) to create attack detection models. As this trend gains traction, researchers discuss whether these IDS can detect unknown attacks. Most ML-based IDS are based on supervised learning, which means they are trained with a limited collection of attack examples. Therefore, detecting attacks that were not covered during the training phase could be challenging for these systems. This work evaluates the ability of ML-based IDS to detect unknown attacks. Our general idea is to understand what happens when a detection model trained with a particular attack A receives incoming data from an unknown attack B. Using the CIC-IDS2017 dataset, we found that supervised intrusion detection models, in most cases, cannot detect unknown attacks. The only exception occurs with DoS attacks. For example, an intrusion detection model trained with HTTP Flood DoS (GoldenEye) samples could detect a different HTTP DoS attack type (Slowloris).

1 Introduction

The increasing number of people connected to the Internet and the massive use of this network have turned this environment into a growing target for cybercriminals. For this reason, having a solid cybersecurity structure is necessary to maintain and sustain organizations' computing resources. In this scenario, tools such as Firewalls, Antivirus, and Intrusion Detection Systems (IDS) are some of the main alternatives capable of mitigating attack attempts to keep computing systems safe.

An IDS is a device that monitors computing systems and detects attack attempts [19]. Regarding its deployment, an IDS can be classified as host-based or network-based. Host-based IDS monitors host characteristics such as CPU

L. Barolli (Ed.): AINA 2023, LNNS 654, pp. 379–391, 2023.
https://doi.org/10.1007/978-3-031-28451-9_33

usage, accessed system files, or applications. Network-based IDS focuses on monitoring network traffic and identifying malicious behavior in the network [1]. The focus of this work is network-based IDS.

There are two types of network-based IDS [11]. Signature-based IDS require a database with information on attack signatures, which they compare with the collected network packets. When the characteristics of a network packet match an attack signature, the IDS generates an alert, and the human operator might take the necessary actions. The second type is referred to as anomaly-based IDS. These systems aim to distinguish normal from malicious behavior. An advantage of anomaly-based IDS, compared to signature-based ones, is that it can detect new or unknown attacks due to its characteristic of analyzing the system's behavior.

Anomaly-based IDS are usually created using a labeled dataset in batch mode. This means that the machine learning algorithm is applied once to a static training dataset, and after that, the produced model is used to make predictions for incoming data. Therefore, when the supervised approach is adopted, the "unknown attack detection" assumption of anomaly-based IDS relies heavily on the quality of training data and the similarities of unknown attacks to the already known ones.

Despite the vast literature on this topic, only some studies seek to analyze the behavior of supervised machine learning algorithms in detecting unknown attacks. For example, Zhang et al. [21] proposed a specific model based on deep learning techniques to identify unknown attacks. The proposed approach was evaluated in the datasets "DARPA KDDCUP 99" [18] and CIC-IDS2017 [15]. The results were satisfactory, but the work only investigated a small sample of attacks and did not evaluate simpler algorithms commonly found in the literature, such as Random Forest, Support Vector Machine (SVM), and Decision Tree. Besides, they used a hybrid approach for the learning model (supervised/unsupervised) and also a method to update the classifier. Other studies such as [6,17], and [20] also propose hybrid and specific intrusion models for detecting unknown attacks. Our main contribution is providing a common ground to understand the impact of the most common setup for anomaly-based IDS for detecting unknown attacks: a simple supervised model with no incremental learning techniques.

Therefore, our goal is to evaluate the performance of Machine-Learning (ML) based IDS in detecting unknown attacks. First, we investigated the similarities between attacks using an unsupervised technique called t-Distributed Stochastic Neigh or Embedding (t-SNE). The idea here is to understand whether some attacks share common characteristics. If this is true, we expect the supervised model to learn some of these common patterns in the training phase, facilitating the detection of attacks with similar patterns. Next, we evaluated the performance of a classification algorithm (Random Forest) in detecting attacks that were not covered in the training dataset. These experiments were carried out on the CIC-IDS2017 dataset, and the results suggest that supervised models cannot detect unknown attacks in most cases.

The rest of the paper is organized as follows. Section 2 focuses on related work about ML-based IDS and its ability to detect unknown attacks. Section 3 presents the dataset and discusses the preprocessing methods applied. Section 4 introduces our investigation on the similarities of different attack types. Section 5 describes the performance of supervised intrusion detection models in detecting unknown attacks, and finally, Sect. 6 presents the concluding remarks and future work.

2 Related Work

Louvieris et al. [8] present a technique for detecting unknown attacks by identifying attack resources. This technique relies on k-means clustering, Naive Bayes feature selection, and C4.5 decision tree classification methods. Potnis et al. [12] focus on detecting DDoS-type attacks. The authors propose a hybrid web intrusion detection system (HWIDS) to detect this attack. Five types of DDoS that target the application layer are covered, as well as their unknown variants. The proposed system has an accuracy of 93.48% and a false negative rate of 6.52% in detecting unknown attacks. Alzubi et al. [3] introduce a new ensemble-based system called Unknown Network Attack Detector (UNAD), which proposes a training workflow composed of heterogeneous and unsupervised anomaly detection techniques. According to the study, this approach performs better when detecting unknown attacks and achieves promising results.

Shin et al. [16] point out that although anomaly detection is a good approach for detecting unknown attacks, false positives are highly probable. The work proposes a hybrid form of intrusion detection (signature-based and anomaly-based) using the Fuzzy c-means technique (FCM) alongside other ones, such as Classification and Regression Trees (CART), to avoid this problem.

Al-Zewairi et al. [2] reiterate that the problem of detecting completely unknown attacks on a system is still an open field of research as these attacks represent a complex challenge for any IDS. The work emphasizes that some definitions for unknown attacks are inconsistent and proposes a categorization into two types of attacks (Type-A and Type-B). In this case, the first type represents new attacks, and the second one represents unknown attacks but in already known categories. Experiments were carried out with IDS based on neural networks to detect Type-A and Type-B attacks as a binary classification problem. The results on two datasets (UNSW-NB15 and Bot-IoT) showed that the evaluated models (deep and shallow ANN classifiers) had poor overall generalization error measures - the classification error rate for several types of unknown attacks was around 50%.

Serinelli et al. [14] use a supervised approach to investigate unknown attack detection in the following datasets: KDD99, NSL-KDD, and CIC-IDS2018. To mimic an unknown attack, the authors performed two simple attacks (DoS using the Hping tool and PortScan using the Nmap tool) in a VirtualBox environment. Then, they recorded the packet capture files and inputted them into three intrusion detection models trained with the following algorithms: SVM, Random Forest, and XGBoost. The results show that, in most scenarios, both attacks have misclassification errors. In some cases, when the unknown attack exhibits a network traffic profile similar to the well-known attacks, the model can identify the attack type correctly. In our work, we investigated the similarity of different attack types to understand whether some attacks have similar network traffic profiles.

In [5], Ferreira and Antunes propose an evaluation of the CIC-IDS2018 dataset and compare the performance of some supervised and bioinspired algorithms, namely: CLONALG Artificial Immune System, Learning Vector Quantization (LVQ), and Back-Propagation Multi-Layer Perceptron (MLP). They also investigated how this approach can deal with some unknown attacks. The authors only work with two scenarios: a) detect different DoS attack types (training with GoldenEye and test with Slowloris), and b) train with data from one attack type and test with another type (DoS traffic × DDoS traffic). According to the authors, the proposed IDS performed better for scenario b) when it comes to identifying unknown attacks.

In comparison to the studies mentioned above, the following differences can be found in our work: i) only some specific attack types were investigated (DoS/DDoS and PortScan), ii) some works proposed specific methods (hybrid techniques - supervised/unsupervised, for instance) tailored to identify unknown attacks, iii) some works still use outdated datasets such as KDD99 and NSL-KDD and iv) absence of a benchmark for detection of unknown attacks using supervised IDS model built with traditional ML classifiers instead of Deep Learning.

3 Dataset and Preprocessing

When selecting the dataset for our experiments, several factors were considered, including the availability of labels, diversity of attack types, and recent attack data [7]. We chose the CIC-IDS2017 [10,15] dataset, as it is one of the most complete and consolidated options among datasets for IDS [13].

The dataset contains 2,830,743 records comprising over 78 network flow features with both normal traffic and attacks observed during a week (Monday to Friday). The dataset encompasses seven attack classes and 16 attack types. Table 1 shows the attack classes and types that were used in this work. The dataset also includes raw network packet data (PCAP format) and network flows labeled as normal (benign) attacks. There is only one day (Monday) with only normal traffic.

For better performance and correct functioning of the classification algorithms, we conducted the following pre-processing procedures using Python language, version 3.9.10: i) Encode categorical variables, ii) Treating infinity and null values, and iii) Normalization.

Concerning the categorical variables, we first converted benign and attack labels to 0 and 1. We then handled the “DestinationPort”. This attribute represents the destination port requested by the attacker, which required proper treatment. As a result, we created a column for the following ports: 80, 53, 21, 22, 123, and 443. These ports are frequently targeted by attackers and associated with HTTP, DNS, FTP, SSH, NTP, and HTTPS protocols. We also created other columns for the following TCP/UDP ports: i) under 1024 but not the previous seven values and ii) greater than 1024. There are no attributes for the source port. We did both treatments for categorical variables using one-hot encoding, so we deleted the original column.

Network flows with infinite and null values were removed from the model since they were very few concerning the original dataset size. Finally, we applied the divide-by-maximum normalization over all features except the "Destination-Port" by dividing the attribute value by the highest value in its category.

4 Similarity Analysis of Different Attacks

4.1 General Idea and Motivation

Supervised models tend to be out of calibration with the increasing number and variety of attacks. Consequently, they require recurrent evaluations to ensure their effectiveness, which can be pretty costly due to the volume of information. However, if different attacks have similar features, we can suppose that the supervised models will continue to perform well. For example, suppose a supervised model is trained with samples from attack A, and a new attack B, similar to attack A, is fed into the classifier. In that case, we expect that the model would be able to detect attack B.

The main idea here is to look for evidence of similarity between different attacks. For this purpose, we performed t-Distributed Stochastic Neigh or Embedding (*t-SNE*) [9]. *t-SNE* is a dimension reduction technique developed to facilitate the visualization of high-dimensional datasets [9]. Our idea is to identify groups of attacks that share similar characteristics.

By applying this technique, we seek to analyze how the network flows associated with attacks are visually distributed in the search space and check if different types of attacks appear within the same groups formed by *t-SNE*, therefore, sharing similarities. If groups are composed of different attack types, then this would indicate that such attack types might have similar network traffic characteristics. In that case, we have more confidence that the rule or the logic used to detect attack types in the same group is also similar.

4.2 Results

To understand the similarities across network flows associated with different attacks, we applied a reduction to two dimensions of *t-SNE* using only data points of the following attack classes in the CIC-IDS2017 dataset: Brute force, DoS, and Web attack. We do not use the benign class, as the objective is to compare only attacks. Finally, we used the entire feature set available in the dataset. We run this experiment using R version 4.2.1 with library M3C version 1.20.0 [4]. Figure 1 shows the visual representation of CIC-IDS2017 attacks through a two-dimensional reduction using *t-SNE*.

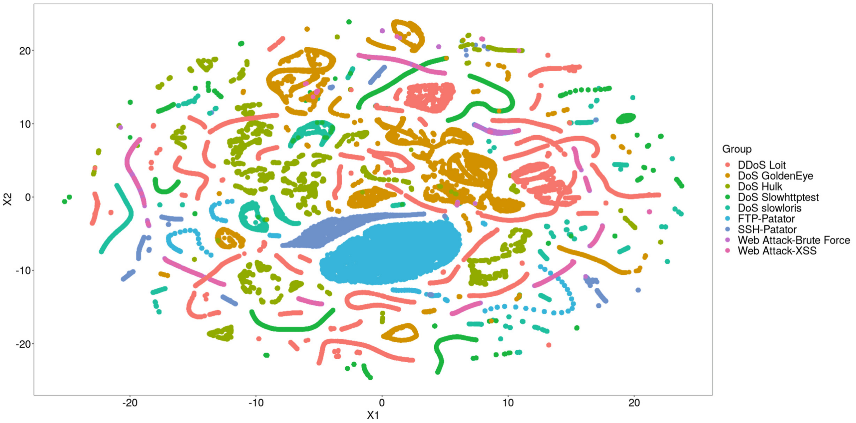

Fig. 1. *t-SNE* visualization in two dimensions

As our goal is to look for similarities between the attacks, what we should observe in the *t-SNE* plot, if it existed, would be groupings of different attacks. Figure 1 shows us that different attacks, represented by colors, are not in the same groupings; we see a clear separation. We can also see that the same type of attack has different patterns, such as DDoS Loit, represented by the orange color, has several groups distributed by the graph. Therefore, this indicates that the different attacks form isolated groups, and even within the same attack type, we observe different behaviors.

Consequently, the analysis shows no conclusive evidence of similarity between different attacks, and the data patterns may differ even within the same type. In the next section, we will conduct some supervised analyzes to see how the intrusion detection model behaves when instances that do not belong to the training set are presented to the classifier.

5 Supervised Models Performance in Detecting Unknown Attacks

The results presented in the previous section indicated that network traffic flows belonging to different types of attacks did not have clear similarities in their features. This may suggest that an intrusion detection model trained with certain attack types might not be able to detect attacks that were not previously seen. The main goal of this section is to empirically evaluate the performance of supervised intrusion detection models when presented to unseen/unknown attack types. Next, we show the experimental methodology and the results obtained using the CIC-IDS2017 dataset.

5.1 Experimental Methodology

Our experimental methodology consists of the following steps:

1. Organization of attack classes and establishment of a baseline dataset;
2. Selection of classification algorithms;
3. Creation of different training/testing sets according to attack classes and types;
4. Performance evaluation of supervised intrusion detection models in different scenarios.

Each data instance in the dataset will have a label B (benign) or A (attack). Attack instances might be grouped in classes $C_1, C_2, ..., C_i$ where i denotes the number of attack classes. Each attack class is composed of a type $t_{i,j}$ where j denotes the number of attack types. In this work, we investigated three classes of attacks: DoS, Brute Force, and Web Attacks (WA). Inside each class, we have several attack types. The DoS attack class is composed of DoS GoldenEye, DoS Hulk, DoS SlowHTTPTest, DoS SlowLoris, and DDoS Loit. The Brute Force class consists of the following attacks: FTP-Patator and SSH-Patator. Finally, the Web Attack class contains the Web Attack Brute Force and Web Attack XSS types. The Baseline dataset consists of samples from both labels B (benign) and A (attack). We selected random samples from CIC-IDS2017 to compose each class. Table 1 details the Baseline dataset composition.

Table 1. Number of attack samples organized by class and type

Type	Class	Number of samples
Benign	Benign	174421
FTP-Patator	Brute Force	794
SSH-Patator	Brute Force	590
DDoS Loit	DoS	12803
DoS GoldenEye	DoS	1043
DoS Hulk	DoS	23107
DoS Slowhttptest	DoS	550
DoS slowloris	DoS	580
Web Attack - Brute Force	Web Attack	151
Web Attack - XSS	Web Attack	65

We selected the following classification algorithms as potential candidates for developing an intrusion detection system: Decision Tree Classifier, Stochastic Gradient Descent Classifier, Multi-layer Perceptron, Gaussian Naive Bayes, K-Nearest Neighbors Classifier, Random Forest Classifier, Linear Support Vector Classifier, Logistic Regression, and Extra-trees Classifier. These algorithms represent some of the most used ones in the ML-based IDS literature. Then, we evaluated the prediction performance of each classification algorithm using the Baseline dataset. We adopted a hold-out strategy of 90/10, and the goal was to classify attack and benign samples (binary task) correctly. Our preliminary results showed that the Random Forest algorithm had a good performance in terms of Precision, Recall, AUC, and training time. For this reason, we selected Random Forest for the rest of the experiments. It is noteworthy that we used Python 3.9.10, scikit-learn, and default parameters for all algorithms in every experiment.

The next step involves creating different training/testing sets according to attack classes and types. We first created six datasets to evaluate the performance of intrusion detection models in detecting unknown attack classes. The forming rule for these sets can be summarized as follows:

- Select an attack class C_k;
- The training set will be composed of all samples from the selected attack class C_k plus benign samples;
- The testing set will be composed of a different attack class C_j and benign samples.
- Repeat the process until all attack classes are evaluated in training and testing sets.

For example, if we select the attack class $C_1 =$ DoS, we will have the training set composed of all DoS samples and the testing set consisting of samples from $C_2 =$ Web Attack. In both sets (train/test), we add a number of benign samples proportional to the number of attack samples. Table 1 details the number of samples for each class.

We also created 24 other datasets to observe the results from a more detailed perspective, moving our focus from attack classes to types. The forming rule for these sets is similar to the previous one and can be summarized as follows:

- Select an attack type $t_{k,j}$;
- The training set will be composed of all samples from the selected attack type $t_{k,j}$ and also of benign samples;
- The testing set will be composed of a different attack type from the same attack class $t_{k,l}$ and also of benign samples.
- Repeat the process until all attack types inside every attack class are evaluated in training and testing sets.

For example, if we select the attack type $t_{1,1} =$ DoS GoldenEye, we will have the training set composed of all DoS GoldenEye samples and the testing set composed of samples from $t_{1,2} =$ DoS Hulk. In both sets (train/test), we add a number of benign samples proportional to the number of attack samples. Table 1 details the number of samples for each class.

We evaluated the models using the following metrics: Precision, Recall, and AUC (Area Under the Curve). In our experiments, the positive class represents an attack, and the negative class represents a benign network flow. Recall represents the ratio of correctly classified positive samples (attacks) to the total positive examples in the dataset. Precision quantifies the number of positive class predictions that belong to the positive class. Finally, we plotted each model's receiver operating characteristic (ROC) curves. The ROC curve plots the True Positive Rate (TPR) vs. the False Positive Rate (FPR) of a given model. The area under the ROC curve is defined as the Area Under the Curve (AUC). The AUC is a value between 0 and 1. Efficient models have AUC values closer to 1.

5.2 Results and Discussion

Table 2 shows the performance of supervised models when detecting unknown attack classes. Precision, Recall, and AUC rates for all models differ significantly from the baseline. For example, an intrusion detection model trained with only DoS samples cannot detect unknown attacks related to Web Attack and Brute force. The same holds when analyzing intrusion detection models trained with Web Attack and Brute Force. The low Recall values for all experiments indicate that supervised models trained with a particular attack class cannot identify a new (unseen) attack class.

Table 2. Performance of supervised models according to attack class

Training	Test	Precision	Recall	AUC
Baseline	Baseline	0.9257	0.9454	0.9791
DoS	Web Attack	0.2303	0.0973	0.5663
DoS	Brute Force	0	0	0.4742
Web Attack	DoS	0.7442	0.0001	0.6361
Web Attack	Brute Force	0.0185	0.0001	0.4966
Brute Force	DoS	0	0	0.4836
Brute Force	Web Attack	0	0	0.8366

The subsequent analysis refers to investigating the behavior of the intrusion detection model when it is trained with specific attacks inside a class and what happens when a new variant of such attack type is presented to the classifier. The results presented in Sect. 4.1 showed no conclusive evidence of similarity between different attack types. We intend to check whether the same conclusion holds when we use supervised models. Table 3 summarizes the performance of each intrusion detection model regarding the attack type.

The scenario here is somewhat different compared to intrusion models trying to detect a new attack class. When analyzing the 20 scenarios of DoS attack types, we have mean values of 0.83, 0.41, and 0.91 for Precision, Recall, and AUC, respectively. This result indicates that a model trained with a DoS type might be able to identify a new/unseen DoS attack type. In some cases, the intrusion

Table 3. Performance of supervised models according to attack types

Training	Test	AC	Prec.	Recall	AUC
Baseline	Baseline	–	0.925	0.945	0.979
DoS GoldenEye	DoS Hulk	DoS	0.988	0.336	0.947
DoS GoldenEye	DoS Slowhttptest	DoS	0.994	0.830	0.983
DoS GoldenEye	DoS Slowloris	DoS	0.997	0.830	0.994
DoS GoldenEye	DDoS LOIT	DoS	0	0	0.778
DoS Hulk	DoS GoldenEye	DoS	0.743	0.999	0.971
DoS Hulk	DoS Slowhttptest	DoS	1	0.022	0.981
DoS Hulk	DoS Slowloris	DoS	0.996	0.587	0.985
DoS Hulk	DDoS LOIT	DoS	0.999	0.168	0.999
DoS Slowhttptest	DoS GoldenEye	DoS	0.999	0.747	0.957
DoS Slowhttptest	DoS Hulk	DoS	0.979	0.209	0.979
DoS Slowhttptest	DoS Slowloris	DoS	0.997	0.995	0.997
DoS Slowhttptest	DDoS LOIT	DoS	0.972	0.138	0.972
DoS Slowloris	DoS GoldenEye	DoS	0.994	0.742	0.992
DoS Slowloris	DoS Hulk	DoS	0.976	0.352	0.968
DoS Slowloris	DoS Slowhttptest	DoS	0.988	0.952	0.993
DoS Slowloris	DDoS LOIT	DoS	0	0	0.946
DDoS LOIT	DoS GoldenEye	DoS	0	0	0.628
DDoS LOIT	DoS Hulk	DoS	0.978	0.034	0.742
DDoS LOIT	DoS Slowhttptest	DoS	0.991	0.044	0.838
DDoS LOIT	DoS Slowloris	DoS	0.996	0.214	0.653
FTP-Patator	SSH-Patator	BF	0	0	0.637
SSH-Patator	FTP-Patator	BF	0	0	0.746
WA Brute Force	WA XSS	WA	0.934	0.998	0.983
WA XSS	WA Brute Force	WA	0.852	0.983	0.975

detection model's performance is even better than the baseline, e.g., in a scenario where we have DoS Slowhttptest in the training set and DoS Slowloris in the testing set. Both attacks are related to the Application Layer (HTTP protocol) in this case. However, in other cases, the intrusion detection model could not detect an unknown DoS attack type, for example, DoS Slowloris (training) and DoS LOIT (test).

We found similar results for Web attacks. In both cases, Web Attack Brute Force and Web Attack XSS, the supervised model could detect a new/unseen attack of such type. The only attack class where the trained models were not able to identify unknown attacks was Brute Force. Precision and Recall values were null in both cases (FTP-Patator and SSH-Patator). Since both attacks target different Application Protocols (FTP and SSH), the network traffic similarity between these two attacks is supposedly low.

To better understand the results, we conducted a new round of experiments. We created newer datasets according to the following rationale: 1) what happens if the training set has several attack classes/types and the test set has only one attack class/type that was not part of the training set? and 2) what happens if the training set has only one attack class/type and the training set has several attack classes/types that were not part of the training set? The idea here is to understand whether some unseen attacks can be detected using previous knowledge from other attacks. In case i), for example, if the training set is composed of C_1 = DoS and C_2 = Web Attack, the test set will have only attack samples from C_3 = Brute Force (i.e., the only attack class that was not part of the training set). In case ii), for example, if the training set is composed of C_1 = DoS, the test set will have attack samples from C_2 = Web Attack and C_3 = Brute Force (i.e., the attack classes that were not part of the training set). The same logic applies to the tests considering attack types. As discussed in Sect. 5.1, benign samples are also part of the training and testing sets.

The results of these new experiments showed that even when we trained the intrusion detection model with multiple attack classes/types, it was still challenging for the model to detect unseen attacks. Only two scenarios were close to the baseline. In the first one, we have a training set composed of all attack types except DoS Slowloris, and the testing set is composed of only DoS Slowloris attacks. In this scenario, Random Forest achieved values greater than 90% for Precision, Recall, and AUC. In the second one, we have a training set composed of all attack types except Web Attack - Brute Force and the testing set composed of only Web Attack - Brute Force. In this scenario, we had values higher than 90% for Precision and AUC and 83% for Recall.

Our results indicate the following implications for practice: i) the performance of supervised IDS models is directly related to the attacks presented in the training set, ii) supervised IDS models may be successful in detecting unknown/unseen variants of the same attack types, for example, DoS and some Web Attacks and iii) due to the potential similarity between some attack types, it might be interesting to develop intrusion detection models tailored for them.

6 Conclusion

We proposed a study of ML-based IDS and their ability to detect unknown/ unseen attacks. Our findings show that a supervised model trained with a specific attack type can identify unseen samples of the same attack type in some situations. For example, supervised models trained with DoS GoldenEye could identify attack samples from DoS Slowloris. We also conclude that supervised models did not efficiently detect unseen attack classes. Two different experiments indicate this: i) when a training set is composed of one attack class and a test set has one different attack class, and ii) when a training set is composed of several attack classes, and a testing set has one unseen attack class.

Future work includes conducting similar experiments on datasets such as CSE-CIC-IDS2018, UNSW-NB15, and UGR'16. We also want to apply transfer learning methods to evaluate the capability of ML-based IDS to detect unknown attacks collected from different datasets.

References

1. Molina-Coronado, B., Mori, U., Mendiburu, A., Miguel-Alonso, J.: Survey of network intrusion detection methods from the perspective of the knowledge discovery in databases process. IEEE Trans. Netw. Serv. Manage. **17**, 2451–2479 (2020)
2. Al-Zewairi, M., Almajali, S., Ayyash, M.: Unknown security attack detection using shallow and deep ANN classifiers. Electronics **9**(12), 2006 (2020)
3. Alzubi, S., Stahl, F., Gaber, M.M.: Towards intrusion detection of previously unknown network attacks. Commun. ECMS **35**(1), 35–41 (2021)
4. Christopher John [Aut, Cre]: M3c (2017). https://doi.org/10.18129/B9.BIOC.M3C, https://bioconductor.org/packages/M3C
5. Ferreira, P., Antunes, M.: Benchmarking behavior-based intrusion detection systems with bio-inspired algorithms. In: Thampi, S.M., Wang, G., Rawat, D.B., Ko, R., Fan, C.-I. (eds.) SSCC 2020. CCIS, vol. 1364, pp. 152–164. Springer, Singapore (2021). https://doi.org/10.1007/978-981-16-0422-5_11
6. Jongsuebsuk, P., Wattanapongsakorn, N., Charnsripinyo, C.: Network intrusion detection with fuzzy genetic algorithm for unknown attacks. In: The International Conference on Information Networking 2013 (ICOIN), pp. 1–5. IEEE (2013)
7. Kenyon, A., Deka, L., Elizondo, D.: Are public intrusion datasets fit for purpose characterising the state of the art in intrusion event datasets. Comput. Secur. 102022 (2020)
8. Louvieris, P., Clewley, N., Liu, X.: Effects-based feature identification for network intrusion detection. Neurocomputing **121**, 265–273 (2013)
9. van der Maaten, L., Hinton, G.: Viualizing data using t-SNE. J. Mach. Learn. Res. **9**, 2579–2605 (2008)
10. University of New Brunswick, U.o.N.B.: Intrusion detection evaluation dataset (cic-ids2017) (2017). https://www.unb.ca/cic/datasets/ids-2017.html
11. Otoum, Y., Nayak, A.: AS-IDS: anomaly and signature based ids for the internet of things. J. Netw. Syst. Manage. **29**(3), 1–26 (2021)
12. Potnis, M.S., Sathe, S.K., Tugaonkar, P.G., Kulkarni, G.L., Deshpande, S.S.: Hybrid intrusion detection system for detecting DDoS attacks on web applications using machine learning. In: Fong, S., Dey, N., Joshi, A. (eds.) ICT Analysis and Applications. LNNS, vol. 314, pp. 797–805. Springer, Singapore (2022). https://doi.org/10.1007/978-981-16-5655-2_77
13. Ring, M., Wunderlich, S., Scheuring, D., Landes, D., Hotho, A.: A survey of network-based intrusion detection data sets. Comput. Secur. **86**, 147–167 (2019)
14. Serinelli, B.M., Collen, A., Nijdam, N.A.: On the analysis of open source datasets: validating ids implementation for well-known and zero day attack detection. Proc. Comput. Sci. **191**, 192–199 (2021)
15. Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: ICISSp, pp. 108–116 (2018)
16. Shin, G.Y., Kim, D.W., Kim, S.S., Han, M.M.: Unknown attack detection: combining relabeling and hybrid intrusion detection. CMC-Comput. Mater. Continua **68**(3), 3289–3303 (2021)

17. Song, J., Ohba, H., Takakura, H., Okabe, Y., Ohira, K., Kwon, Y.: A comprehensive approach to detect unknown attacks via intrusion detection alerts. In: Cervesato, I. (ed.) ASIAN 2007. LNCS, vol. 4846, pp. 247–253. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76929-3_23
18. Stolfo, S.J., Fan, W., Lee, W., Prodromidis, A., Chan, P.K.: Cost-based modeling for fraud and intrusion detection: results from the jam project. In: Proceedings DARPA Information Survivability Conference and Exposition, DISCEX 2000, vol. 2, pp. 130–144. IEEE (2000)
19. Tsai, J.J., Yu, Z.: Intrusion Detection: A Machine Learning Approach, vol. 3. World Scientific (2011)
20. Xu, M.F., Li, X.H., Miao, M.X., Zhong, C., Ma, J.F.: An unknown attack detection scheme based on semi-supervised learning and information gain ratio. J. Internet Technol. **20**(2), 629–636 (2019)
21. Zhang, Z., Zhang, Y., Guo, D., Song, M.: A scalable network intrusion detection system towards detecting, discovering, and learning unknown attacks. Int. J. Mach. Learn. Cybern. **12**, 1649–1665 (2021)