International Journal of Information Security (2023) 22:947–959
https://doi.org/10.1007/s10207-023-00676-0

REGULAR CONTRIBUTION

From zero-shot machine learning to zero-day attack detection

Mohanad Sarhan[1] · Siamak Layeghy[1] · Marcus Gallagher[1] · Marius Portmann[1]

Published online: 15 March 2023

Abstract

Machine learning (ML) models have proved efficient in classifying data samples into their respective categories. The standard ML evaluation methodology assumes that test data samples are derived from pre-observed classes used in the training phase. However, in applications such as Network Intrusion Detection Systems (NIDSs), obtaining data samples of all attack classes to be observed is challenging. ML-based NIDSs face new attack traffic known as zero-day attacks that are not used in training due to their non-existence at the time. Therefore, this paper proposes a novel zero-shot learning methodology to evaluate the performance of ML-based NIDSs in recognising zero-day attack scenarios. In the attribute learning stage, the learning models map network data features to semantic attributes that distinguish between known attacks and benign behaviour. In the inference stage, the models construct the relationships between known and zero-day attacks to detect them as malicious. A new evaluation metric is defined as *Zero-day Detection Rate (Z-DR)* to measure the effectiveness of the learning model in detecting unknown attacks. The proposed framework is evaluated using two key ML models and two modern NIDS data sets. The results demonstrate that for certain zero-day attack groups discovered in this paper, ML-based NIDSs are ineffective in detecting them as malicious. Further analysis shows that attacks with a low Z-DR have a significantly distinct feature distribution and a higher Wasserstein Distance range than the other attack classes.

Keywords Machine learning · Network Intrusion Detection System · Wasserstein Distance · Zero-day attacks · Zero-shot learning

1 Introduction

Over the past few years, machine learning (ML) capabilities have been used to enhance the performance and efficiency of various technological applications [1]. ML is a subset of Artificial Intelligence (AI) [2], involving a set of statistical algorithms that can learn from data without being explicitly programmed [3]. ML models are recognised for their superior ability to extract and learn complex data patterns that are not feasible to observe by domain experts [4]. The learnt patterns are used to predict, classify, and regress future events and scenarios. ML has been a disruptive innovation [5] in multiple industries where operational automation and efficiency are required. Therefore, ML models have been widely deployed across multiple domains, proving great success over traditional computing algorithms. The same motivation has led to implementing ML models in the cybersecurity domain [6] to enhance and strengthen organisations' security posture. ML operations are capable of detecting complicated modern attacks that require advanced innovative detection capabilities [7]. The addition of the intelligence element to the organisation's security strategy adds sophisticated layers of defense [8] that can limit the number of internal and external threats if designed efficiently [9].

Network Intrusion Detection Systems (NIDSs) are essential security tools that detect threats as they penetrate the network environment of an organisation [10]. Traditional signature-based NIDSs scan incoming network traffic for any Indicator of Compromise (IOC), also known as attack signatures, such as source IPs, domain names, and hash values, which could indicate malicious traffic [11]. One of the primary and ongoing challenges of securing computer networks with signature-based NIDSs is the detection of zero-day attacks [12]. A zero-day attack is a new kind of threat that has not been seen before [13], designed to infiltrate or disrupt network communications. It is an unknown vulnerability to security administrators that hackers can exploit before its remediation. A recent example is a zero-day vulnerability discovered in Microsoft Windows in June 2019

✉ Mohanad Sarhan
m.sarhan@uq.net.au

[1] University of Queensland, Brisbane, Australia

that targeted local escalation privileges [14]. Generally, when a zero-day attack is discovered, it is added to the publicly shared Common Vulnerabilities and Exposures (CVE) list [15] and defined using a CVE code and a severity level [15]. From a network layer perspective, zero-day attack detection is generally carried out by adding threat-related IOCs to a list of detection databases [16] used by signature-based NIDSs. As such, signature-based NIDSs are deemed unreliable in detecting zero-day attacks simply because the complete set of IOCs has not been discovered or registered for monitoring at the time of exploitation.

Organisations protected by signature-based NIDS are vulnerable to zero-day attacks without the discovery of IOCs associated with the threat. Therefore, the focus has been diverted to the design of ML-based NIDSs [12], an enhanced modern edition of traditional NIDSs to overcome the limitations faced in the detection of zero-day or unseen attacks. ML-based NIDSs are designed and deployed to scan and analyse incoming network traffic for any anomalies or malicious intent [12]. The analysis process is carried out by comparing the behavioural pattern of the incoming network traffic with the learnt behaviour of safe and intrusive traffic [17]. During the design process, the ML model is trained using a set of benign and attack samples, where the hidden complex traffic pattern is learnt. Unlike the signature-based NIDSs, which solely rely on IOC for detection, the ML-based NIDSs utilise the learnt behavioural pattern to detect network attacks [17]. This has a great potential of detecting zero-day attacks as the requirement of obtaining IOC is obsolete [18]. The main difference between signature- and ML-based NIDSs is the detection engine functionality, whereas signature-based detection relies on IOCs. In contrast, ML-based detection focuses on malicious and benign behavioural patterns. Most of the available research work has aimed at designing and evaluating ML-based NIDSs to detect known attack groups. However, limited research has focused on evaluating zero-day attack detection to measure the benefits of ML-based NIDSs over signature-based NIDSs.

A large number of proposed ML-based NIDSs do not consider the most likely re-occurring scenario of zero-day attacks, where a new attack class may appear after the learning stage and deployment of the ML model. Zero-shot learning (ZSL) is an emerging methodology used to evaluate and improve the generalisability of ML models to new or unseen data classes [19]. This technique assumes that the training data set might not include the entire set of classes that the ML model could observe once deployed in the real world. ZSL addresses the ever-growing set of classes that might render it unfeasible to collect training samples for each of them [20]. ZSL involves the recognition of new data samples derived from previously unseen classes. As such, ZSL addresses one of the main challenges in building a reliable NIDS: the evaluation of recognising new attack classes that are not available in the training phase [21]. This includes zero-day attacks that could lead to fatal consequences for the adopting organisation if undetected [13]. A reliable ML-based NIDS must be evaluated across a test set of unknown attacks not available in the training set (unseen classes), simulating the likely scenario of a zero-day threat.

This paper proposes a new ZSL framework to evaluate the performance of ML-based NIDSs in recognising zero-day attack scenarios. The framework measures how well an ML-based NIDS can detect unseen attacks using a set of semantic attributes learnt from seen attacks. There are two main stages of the proposed ZSL setup. In the attribute learning stage, the models extract and map the network data features to the unique attributes of known attacks (seen classes). In the inference phase, the model associates the relationships between seen and zero-day (unseen) attacks to assist in their discovery and classification as malicious. The training and testing sets containing the seen and unseen classes remain disjoint throughout the setup. Unlike traditional evaluation methods, the proposed set-up aims to evaluate ML-based NIDS in detecting zero-day attacks using a new metric, Zero-day Detection Rate (Z-DR). The proposed methodology has been implemented using two widely used ML models in the research field. It has been evaluated on two key NIDS data sets, each consisting of a wide range of modern attacks. Furthermore, the results obtained were analysed using the Wasserstein Distance (WD) technique to investigate and explain the variation in the Z-DR with different attack groups. The key contributions of this paper are a) the proposal of a novel ZSL-based methodology to evaluate NIDSs in recognition of unseen (unseen) attack types, b) the implementation of the framework using two widely-used ML models and two modern NIDS data sets, and c) the analysis and explanation of the detection results using the WD technique. In Sect. 2, key related works are discussed, followed by a detailed explanation of the proposed ZSL-based methodology in Sect. 3. The experimental methodology followed in this paper and the results obtained are discussed and explained in Sects. 4 and 5, respectively.

2 Related works

This section discusses key related papers that aim to evaluate NIDSs for the detection of zero-day attacks. Although most of the articles propose sophisticated ML-based NIDSs [22], the evaluation focuses on detecting a range of known attacks, where traditional signature-based NIDSs have achieved satisfactory performance throughout the years. Therefore, it is surprising that only a few papers have attempted to challenge ML-based NIDSs in the detection of unknown or zero-day attacks. In the case of unsupervised anomaly detection systems, where the model only learns the behaviour of benign traffic, NIDSs fundamentally work to detect each attack type as an unknown attack. However, it is noted that such models lead to many false alarms leading to alert fatigue [23], as it does not consider the attack behavioural pattern. Overall, a limited number of papers follow a ZSL methodology to detect zero-day attacks. To the best of our knowledge, none of these works has aimed to utilise modern network data sets that represent current network traffic characteristics to evaluate their approach.

In [24], the author has evaluated the zero-day attack detection performance using a signature-based NIDS. The paper studies the frequent claim that such systems cannot detect zero-day attacks. The experiment studies 356 network attacks, of which 183 are unknown (zero-day) to the ruleset. The paper utilised the Snort tool, a well-known signature-based NIDS. The Metasploit framework is used to simulate attack scenarios. The detection rate is calculated by applying a Snort rule set that does not disclose the vulnerabilities relevant to the attack. The results show that Snort has an unreliable detection rate of 17% against zero-day attacks. The paper argues that the frequent claim that signature-based NIDSs cannot detect zero-day attacks is incorrect, since 17% is significantly larger than zero. The author mentions that more mechanisms should be implemented to complement signature-based NIDS in detecting unregistered attacks. The results of this paper can be seen as a baseline for zero-day attack detection.

Zhang et al. [21] have evaluated ML-based NIDSs detection performance against zero-day attacks. The authors have used ZSL to simulate the occurrence of zero-day attack scenarios. The authors used a sparse autoencoder model that projects the features of known attacks into a semantic space and establishes a feature-to-semantic mapping to detect unknown attacks. ML models learn the distinguishing information between the attack and benign classes by mapping the feature and attribute space. The paper used the attacks present in the NSL-KDD data set, released in 1998, to simulate a zero-day scenario; the data set contains four attack scenarios. The results demonstrate that the average accuracy is 88.3% for all available attacks in the data set.

In [25], Hindy et al. aimed to improve unsupervised outlier-based detection systems that generally suffer from a high false alarm rate (FAR). The paper explored an autoencoder to detect zero-day attacks to maintain a high detection rate while lowering the FAR. The system is evaluated across two key data sets; CICIDS2017 and NSL-KDD. The methodology involved training the classifiers using benign data samples and evaluating the detection of zero-day attacks. The results are compared to a one-class support vector machine, where the autoencoder is superior. The results demonstrate a zero-day detection accuracy of 89–99% for the NSL-KDD data set and 75–98% for the CICIDS2017 data set. However, the proposed models do not consider attack behaviour, and the number of undetected attacks and false alarms is unmeasured.

Li et al. [26] focused on attribute learning methods to detect unknown attack types. The authors followed a ZSL method to design an NIDS to overcome the limitations in anomaly detection faced by current methods. The architecture involves a pipeline using a Random Forest (RF) feature selection and a spatial clustering attribute conversion method. The results demonstrate that the proposed method overcomes the state-of-the-art approaches in anomaly detection. The attribute learning framework converts network data samples into unsupervised cluster attributes. The NSL-KDD data set has been used to evaluate the proposed framework, where it could detect DoS (apache2) and Probe (saint) attacks achieving an overall accuracy of 34.71%. The authors compared its performance with a decision tree classifier with a poor overall accuracy of 13.59%.

In [27], Kumar et al. propose a robust detection model to detect zero-day attacks. The model utilises the concept of high-volume attacks to derive high-traffic volume attacks using heavy-hitter low-volume attacks to derive signatures for low-volume attacks using the graph technique. The proposed framework consists of two stages Signature generation and an evaluation phase. The detection accuracy is evaluated using signatures generated in the training phase. Using a real-time attack data set, 91.33% and 90.35% accuracies were achieved following binary- and multi-classification methods. Using a benchmark CICIDS18 dataset, a performance of 91.62% and 88.98% was achieved.

In general, several studies have evaluated the performance of ML-based NIDSs in detecting unknown attacks. However, only a small number adopted the emerging ZSL-based setup to simulate the occurrence of zero-day attacks. Moreover, minimal experimental work has been done on current zero-day attack scenarios with recent data sets and attack types, which limits the identification of sophisticated attacks that cannot be detected in zero-day scenarios. In addition, it is surprising that some recent work still uses the NSL-KDD data set for evaluation purposes, given that it is more than 20 years old. The attack scenarios in the data set do not represent modern network traffic characteristics and threats, limiting the reliability and evaluation of the proposed methodology [28]. In this paper, a ZSL approach is proposed to evaluate ML models in the recognition of a broader range of modern zero-day attacks. A new metric defined as Z-DR is utilised to measure the detection accuracy of each unseen class. The results presented in this paper are explained and analysed using the WD technique to provide additional insights.

3 Proposed methodology

In a traditional ML evaluation methodology, the learning model is trained and tested on the same set of data classes. The model learns to identify patterns directly from each data class in the training stage. In the testing stage, the model applies the learnt patterns to identify the data samples derived from the same data classes used in the training stage. The data set used in an experimental set-up is split into training and testing partitions. The learning model is trained on the training set that contains the same number and type of classes present in the test set used in the evaluation stage. This evaluation approach follows the assumption that the data set collected for the training of ML models includes the complete set of classes that the model will observe post-deployment in production. In the case of currently proposed ML-based NIDSs, the model is trained and tested using a set of known attack classes. Therefore, the model is evaluated to determine how well it can detect data samples derived from known attack groups as malicious.

The training set D_{tr} and testing set D_{tst} of a NIDS data set can be represented as follows:

$$D_{tr} = \{(x, y)|x \in X_{tr},\ y \in Y_{tr}\} \tag{1}$$

$$D_{tst} = \{(x, y)|x \in X_{tst},\ y \in Y_{tst}\} \tag{2}$$

$$where \ \ X_{tr} \subset X, \ \ X_{tst} \subset X$$

in which x represents a data sample (flow) chosen from the training sets X_{tr} and testing sets X_{tst}. X represents all data samples, y represents the corresponding labels, Y_{tr} represents the set of class labels observed in the training phase, and Y_{tst} represents the set of class labels used in the testing phase. In traditional ML, $Y_{tr} = Y_{tst}$, that is, the set of classes observed during the training phase is identical to the set of classes encountered by the model during testing.

The traditional ML set-up has been commonly used in the ML-based NIDSs evaluation process, proving effective in measuring the detection rate of known attack classes. However, obtaining data samples for each attack class is challenging for different reasons. For instance, zero-day attacks have emerged repeatedly over the past few decades and present a severe risk to the organisation of computer networks. A zero-day attack can be a new kind of modified threat that has not been seen or available earlier [13]. Furthermore, due to the wide variety of tactics and techniques used in executing network attacks, each threat presents a unique behavioural pattern, and the collection of each attack type for ML training is unrealistic. Therefore, the traditional ML evaluation set-up removes the conclusion that ML-based NIDSs are effective in the detection of zero-day or unseen attack scenarios due to their unavailability at the time of training.

ZSL techniques have been adopted to address such shortcomings in the evaluation of ML systems that are required to detect a more extensive set of classes than the one used in training. ZSL was developed principally to overcome the issue of not having training samples available. ZSL is a promising approach to leverage supervised learning for the recognition of unavailable training data samples [19]. Unlike traditional ML methods, the objective of ZSL is to improve the recognition of unseen classes by generalising the learning model to data samples not derived from pre-observed classes. This approach overcomes the limitation of evaluating ML-based NIDSs in the detection of zero-day attacks because the collection of data samples of zero-day attacks remains an impossible task simply due to their absence at the time of the ML-based NIDSs development phase.

In this paper, we propose a ZSL-based methodology, illustrated in Fig. 1, to evaluate ML-based NIDSs in the recognition of zero-day attacks. The proposed methodology will overcome the necessity of collecting training data samples of all the attack classes that the model will observe post-deployments. In the attribute learning stage, the model captures the semantic attributes of the attack behaviour using a set of known attacks and benign data samples. The attributes present the distinguishing vectors between the attack and benign network traffic. In the inference stage, the learnt knowledge is utilised to reconstruct the relationships between known attack classes and the zero-day attack to classify the unseen threat as malicious. Three main data concepts exist as part of the proposed methodology: 1) Known attacks—these are precedent attacks for which labelled data samples are available during training. 2) Zero-day attacks—these unknown attacks will emerge post-deployment for which labelled data samples are unavailable during training. 3) Semantic attributes—the distinguishing information that the ML model will learn from the known attacks to detect the zero-day attacks.

The proposed methodology assumes that the model is evaluated using zero-shot samples derived from an attack class that is unavailable during the training stage at the testing stage. Given an NIDS data set, we can define a ZSL training set D^z_{tr} for attack classes z as follows:

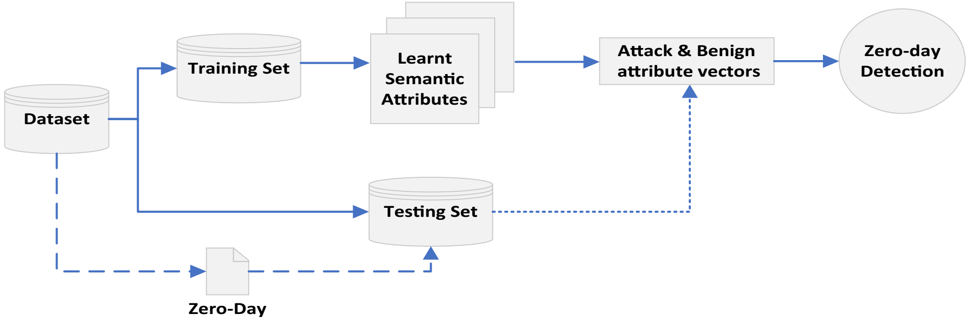
$$D^z_{tr} = \{(x, y)|x \in X_{tr},\ y \in Y^z_{tr} = \{b,\ a_1,\ a_2,\ ..., a_n\} \setminus \{a_z\}\}$$
$$for \ \ z \in \{1, ..., n\} \tag{3}$$

$$D_{tst} = \{(x, y)|x \in X_{tst},\ y \in Y_{tst} = \{b,\ a_1,\ a_2,\ ..., a_n\}\} \tag{4}$$

$$where \ \ X_{tr} \subset X, \ \ X_{tst} \subset X$$

the set of training classes Y^z_{tr} consists of benign traffic b and n attack classes $a_1, ..., a_n$, but importantly, minus the zero-day attack class a_z. In contrast, the test data set D_{tst} always consists of samples of all classes without removing any attack

Fig. 1 Proposed methodology



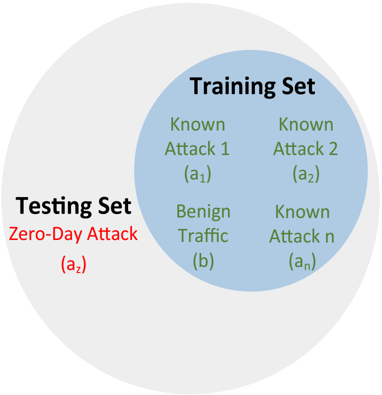


Fig. 2 Venn diagram of the training (D_{tr}^{z}) and test (D_{tst}^{z}) classes

class. By excluding an attack class z from the training phase, we are essentially simulating a zero-day attack scenario, as the ML model has not been trained on the respective attack class. The training set D_{tr}^{z} and the testing set D_{tst} remain disjoint throughout the experiment.

The data set structure of the ZSL evaluation scenario is illustrated in Fig. 2.

During the training stage, the model maps the network data features to the relevant semantic attributes that define the attack behaviour. Assuming that S is the set of samples of known attack classes that can be used to train an ML model, each known attack data sample is denoted by x, their respective labels are denoted by y, and the semantic information about the attack behaviour is denoted by h. Therefore, x and y can be values of one of the known attack samples and classes, respectively. This can be represented using the following notation:

$$S = \{(x, y, h) | x \in X_{tr},\ \ y \in \{a_1,\ a_2,\ ..., a_n\} \setminus \{a_z\}, h \in H\} \\ for \quad z \in \{1, ..., n\} \tag{5}$$

where H is the set of learnt attributes used to predict zero-day attacks.

During the inference phase, the zero-day attack traffic class a_z is added to the test set to measure the zero-day detection accuracy. Therefore, the test set includes known attacks, a zero-day attack, and benign data samples. This follows a generalised ZSL setting where the test set includes seen (known attacks and benign classes) or unseen (zero-day attack class) data samples [29]. This is appropriate for ML-based NIDS evaluation as it represents a real-world environment and a more practical scenario than the conventional ZSL setting. As the test set only includes samples from the unseen class, which is challenging to guarantee from a network security perspective. The goal of the testing phase is to reconstruct the relationships between known attacks and zero-day attacks and associate them as malicious. The nearest neighbour search, also known as the class-class similarity, instance-label assignment technique, is utilised by the models to perform the reconstruction phase, in which the model recognises the test sample that corresponds to the same or the nearest position in the semantic space. We simulate a zero-day attack scenario for each available attack class in the data set by removing this class from the corresponding training set.

The proposed ZSL methodology aims to evaluate ML-based NIDSs in their ability to generalise and detect new and unseen attack classes post-deployment, which is a very realistic and relevant scenario in network security. The insights gained from applying our methodology can be used further to optimise the ML model's hyperparameters and architecture, to enhance its generalisability to new attack types. Furthermore, network data feature selection experiments can be performed to identify critical features required to predict behavioural attack patterns to detect zero-day attacks. Reliable detection of zero-day attacks is the fundamental limitation that existing signature-based NIDS faces. Therefore, the practical motivation for organisations to switch to an ML-based NIDS is the classification of zero-day attacks, which is the focus of our proposed method.

4 Experimental setup

The evaluation of the ML-based NIDS capability to detect zero-day attacks is crucial. In this paper, two commonly used ML models have been used in the design of ML-based NIDSs, Random Forest (RF) [30] and Multi-Layer Perceptron (MLP) [31]. The RF classifier is designed using randomised 50 decision tree classifiers in the forest. The model utilises the Gini impurity loss function [32] to measure the quality of a split with no maximum tree depth defined. The RF model requires 2 data samples to split an internal node and 1 data sample to be at a leaf node. The MLP neural network model is structured with 100 neurons in two hidden layers, each performing the Rectified Linear Unit (ReLU) [33] activation function. The Adam optimiser is used for the model's loss function and parameter optimisation with a 0.001 learning rate. A 0.0001 L2 regularisation parameter is used to avoid over-fitting and the training rounds are set to 50. The semantic representations are learnt by the RF and MLP models in the training phase using their respective loss optimisation function. A fivefold cross-validation method is adopted in the inference stage to calculate the mean results.

In this paper, two NIDS data sets are used to evaluate the ML models following the proposed methodology, i.e. UNSW-NB15 [34] and NF-UNSW-NB15-v2 [35]. The data sets are synthetic and were created via virtual network testbeds representing modern network structures. In designing such data sets, specific attack scenarios are conducted, and the corresponding network traffic is captured and labelled with the respective attack type. In addition, benign network traffic is generated that represents benign traffic and is captured and labelled accordingly. Both malicious traffic and non-malicious traffic are captured in the native packet capture (pcap) format, and network data features are extracted to represent explicit information regarding the data flow. The chosen data sets include a variety of modern network attacks, each of which can be used to simulate the incoming of a zero-day attack. Such data sets have been widely used in the literature, as they do not present the privacy limitations faced by the collection and labelling of real-world production networks.

- **UNSW-NB15** [34]- A well-known and widely used NIDS data set was released in 2015 by the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS). The synthetic data set uses the IXIA Perfect Storm tool to generate benign network activities and pre-meditated attack scenarios. The data set contains 49 features, listed and discussed in [34], extracted by the Argus and Bro-IDS tools, and twelve additional SQL algorithms. The data set consists of 2,218,761 (87.35%) benign and 321,283 (12.65%) attack samples, that is, 2,540,044 network data samples in total.
- **NF-UNSW-NB15-v2** [35]- A NetFlow data set based on the UNSW-NB15 data set has recently been generated and released in 2021. The data set is generated by extracting 43 NetFlow-based features, explained in [35], from the pcap files of the UNSW-NB15 data set. The nprobe feature extraction tool extracts network data flows, and the flows are labelled using the appropriate data labels. The total number of data flows is 2,390,275, of which 95,053 (3.98%) are attack samples and 2,295,222 (96.02%) benign.

This paper uses the complete set of data samples in each data set. This is required as distinct nodes on the testbed have been used to launch attack scenarios targeting specific network ports. Initially, the flow identifiers such as sample id, source/destination IPs, source/destination ports, and timestamps are dropped to avoid learning bias towards the attacking and victim endpoints. Moreover, all categorical-based features are converted to numerical-based values using the label encoding technique, where each label is assigned a unique integer. Once a complete numerical data set is obtained, the Min-Max Scaler technique is applied to normalise all values between 0 and 1 to accommodate efficient experiments.

The standard classification performance metrics of precision, detection rate (DR), false alarm rate (FAR), area under the curve (AUC), and F1 score are used for our evaluation. These metrics are defined based on the numbers of True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN), as shown in Table 1. In addition to these standard metrics, we define a new evaluation metric called Zero-Day Detection Rate ($Z\text{-}DR_z$), also shown in Table 1, which is defined as the specific detection rate of the zero-day attack class a_z, which is excluded from the training data set. The TP_{a_z} and FN_{a_z} are the number of True Positives and False Negatives explicitly calculated for the samples from the zero-day attack class a_z. The new metric, defined in Eq. 6, measures how well the ML model can detect zero-day attacks of class a_z. The $Z\text{-}DR_z$ is used to explicitly measure the performance of the trained model in recognising the zero-day attack samples. The DR provides insights into the detection of the complete set of attack samples.

$$Z\text{-}DR_z = \frac{TP_{a_z}}{TP_{a_z} + FN_{a_z}} \times 100 \tag{6}$$

5 Evaluation

In this section, two ML models, MLP and RF, have been used to detect zero-day attacks using our proposed ZSL evaluation framework. The experiments use two synthetic NIDS data sets (UNSW-NB15 and NF-UNSW-NB15-v2). Each available attack class in the data sets is considered to simulate a zero-day attack incident. The models are evaluated based on the Z-DR and the test set's overall detection accuracy, including known attacks, a zero-day attack, and benign data samples. This represents a generalised ZSL set-up where the

Table 1 Evaluation metrics

Metric	Definition	Equation
Accuracy	The percentage of correctly classified samples in the test set	$\frac{TP+TN}{TP+FP+TN+FN} \times 100$
Detection rate (DR)	The percentage of correctly classified total attack samples in the test set	$\frac{TP}{TP+FN} \times 100$
False alarm rate (FAR)	The percentage of incorrectly classified benign samples in the test set	$\frac{FP}{FP+TN} \times 100$
Area under the curve (AUC)	The area underneath the DR and FAR plot curve in the test set	N/A
F1 score	The harmonic mean of the model's precision and DR	$2 \times \frac{DR \times Precision}{DR + Precision}$
Zero-day detection rate ($Z\text{-}DR_z$)	The percentage of correctly classified zero-day attack samples in the test set.	$\frac{TP_{a_z}}{TP_{a_z}+FN_{a_z}} \times 100$

test set includes known and unknown data samples, which is appropriate for ML-based NIDS evaluation. The baseline used for the Z-DR comparison is the traditional DR metric to highlight the difference in each scenario. Finally, the results are analysed and explained using WD to explain the variance of Z-DR with different attack classes.

5.1 Results

Tables 2, 3, 4, 5 display the complete set of results collected. Each table represents a unique ML model and data set combination. The first column in each table lists the attacks used to simulate a zero-day attack incident. The second column displays the corresponding Z-DR value, and the rest presents the remaining evaluation metrics collected over the complete test set, including the zero-day attack, known attacks and benign data samples.

In Tables 2 and 3, the performance of the MLP and RF classifiers, when evaluated using the UNSW-NB15 data set, is presented. During the simulation of zero-day attacks, the Exploits, Reconnaissance, and DoS attacks are detected at around 90% using the MLP classifier. The RF classifier is more effective in detecting Exploits and DoS attacks. The MLP and RF models detect 20% and 15% of the Fuzzer attack data samples, respectively. The MLP model is superior to RF in detecting Generic and Shellcode attack types, achieving a high Z-DR of 96% and 97% compared to 59% and 91%, respectively. The Analysis attack type is deemed complex in its detection as a zero-day attack where the MLP model achieved an 84%, and the RF model detected 81%. Other attack types, such as Backdoor and Worms, were almost entirely detected by both ML models when observed as zero-day attacks.

The performance of both ML models depends on the complexity of the incoming zero-day attacks. The models successfully detected 95% or more samples of attacks such as Generic, DoS, Backdoor, Shellcode, and Worms. However, Exploits, Reconnaissance, and Analysis are harder to detect, with both models achieving around 90% Z-DR. However, in the likely scenario of the models observing attacks related to the Fuzzers attack group as a zero-day attack, ML-based NIDSs would be highly vulnerable as more than 80% of their data samples were undetected and classified as benign samples. The extremely low Z-DRs of both models present severe risks to organisations protected by ML-based NIDSs in a scenario of a new zero-day attack group similar to Fuzzers. The MLP classifier generally achieved an average of 85.5% detection rates in zero-day attacks. The RF classifier was slightly inferior, with an average detection rate of 80.67%.

In Fig. 3, the detection rate of each attack group in the UNSW-NB15 data set is measured in known attack and zero-day attack scenarios. Figure 3a and 3b represents the performance using the MLP and RF models, respectively. The drop in detection rate is highly notable in certain attack types such as Fuzzers and Reconnaissance. The DR value dropped by around 70% and 10%, respectively, for the two ML models. Furthermore, there are distinct differences in the performance of the two models. The MLP model was more successful in detecting Generic attacks as a zero-day at a Z-DR of 95.90% compared to 59.06% achieved by RF.

Table 2 Performance evaluation of MLP on UNSW-NB15

Zero-day attack	Z-DR	Accuracy	F1 score	FAR	DR	AUC
Exploits	**90.31**	98.73	0.92	0.47	89.09	0.94
Fuzzers	**20.10**	96.94	0.74	0.15	59.16	0.80
Generic	**95.90**	98.93	0.93	0.36	90.09	0.95
Reconnaissance	**91.82**	98.93	0.91	0.48	90.17	0.95
DoS	**92.80**	99.0	0.91	0.35	87.70	0.94
Analysis	**84.35**	99.06	0.91	0.53	91.36	0.95
Backdoor	**99.04**	99.03	0.91	0.60	92.10	0.96
Shellcode	**97.15**	99.08	0.91	0.48	90.67	0.95
Worms	**98.25**	99.06	0.90	0.51	90.60	0.95

Table 3 Performance evaluation of RF on UNSW-NB15

Zero-day attack	Z-DR	Accuracy	F1 score	FAR	DR	AUC
Exploits	**94.43**	99.07	0.94	0.33	91.95	0.96
Fuzzers	**14.77**	96.92	0.73	0.06	57.58	0.79
Generic	**59.06**	97.64	0.82	0.38	73.19	0.86
Reconnaissance	**89.08**	99.05	0.93	0.36	90.26	0.95
DoS	**96.89**	99.25	0.93	0.36	92.52	0.96
Analysis	**81.37**	99.22	0.92	0.36	91.37	0.96
Backdoor	**99.60**	99.28	0.93	0.37	92.58	0.96
Shellcode	**90.80**	99.25	0.92	0.35	91.59	0.96
Worms	**100.00**	99.28	0.93	0.37	92.24	0.96

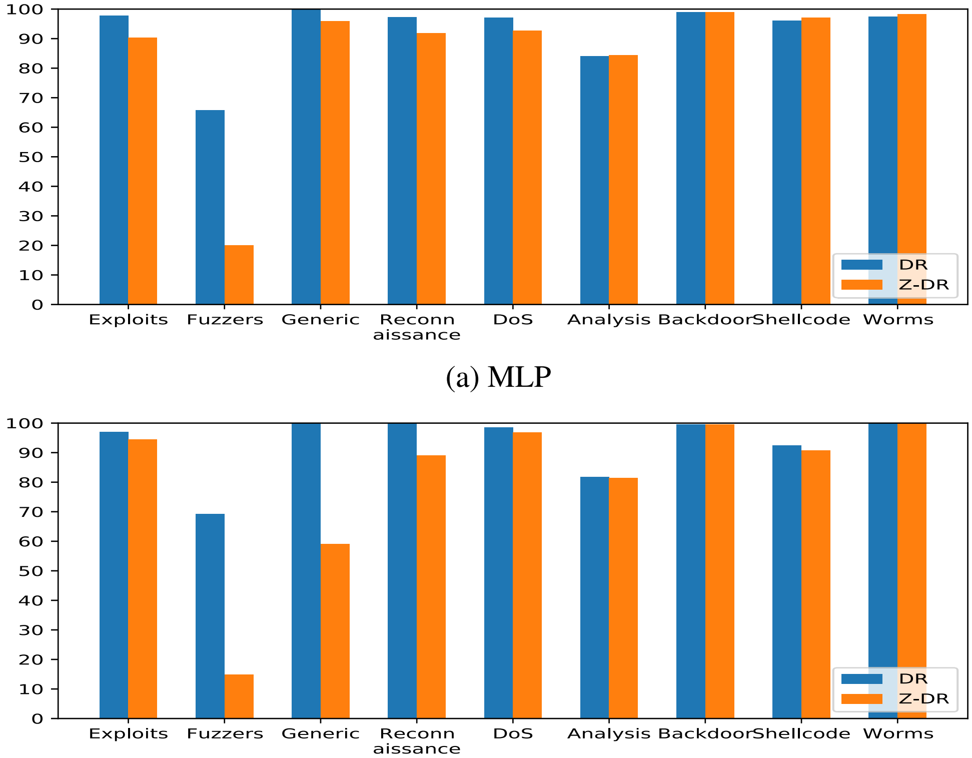
Table 4 Performance evaluation of MLP on NF-UNSW-NB15-v2

Zero-day attack	Z-DR	Accuracy	F1 score	FAR	DR	AUC
Exploits	**81.47**	98.89	0.92	0.29	87.74	0.94
Fuzzers	**76.19**	98.96	0.91	0.19	85.72	0.93
Generic	**99.57**	99.62	0.97	0.33	98.85	0.99
Reconnaissance	**99.75**	99.60	0.96	0.30	97.89	0.99
DoS	**90.68**	99.55	0.95	0.31	96.53	0.98
Analysis	**88.47**	99.60	0.95	0.34	98.23	0.99
Backdoor	**97.28**	99.59	0.95	0.29	96.90	0.98
Shellcode	**98.60**	99.63	0.96	0.30	97.94	0.99
Worms	**100.00**	99.63	0.95	0.32	98.46	0.99

Table 5 Performance evaluation of RF on NF-UNSW-NB15-v2

Zero-day attack	Z-DR	Accuracy	F1 score	FAR	DR	AUC
Exploits	**59.28**	98.07	0.84	0.11	73.33	0.87
Fuzzers	**51.32**	98.38	0.85	0.10	74.61	0.87
Generic	**99.11**	99.75	0.98	0.15	98.05	0.99
Reconnaissance	**99.57**	99.77	0.98	0.15	98.15	0.99
DoS	**93.68**	99.71	0.97	0.15	96.85	0.98
Analysis	**87.95**	99.75	0.97	0.14	97.15	0.99
Backdoor	**99.49**	99.76	0.97	0.16	97.84	0.99
Shellcode	**95.94**	99.75	0.97	0.16	97.70	0.99
Worms	**100.00**	99.77	0.97	0.14	97.59	0.99

Fig. 3 Comparison between DR vs Z-DR of attacks in UNSW-NB15



(a) MLP

(b) RF

Both models achieved a 100% detection rate when the attack class was observed in the training set. The RF classifier has been slightly more efficient in detecting the Exploits and DoS attack groups as a zero-day.

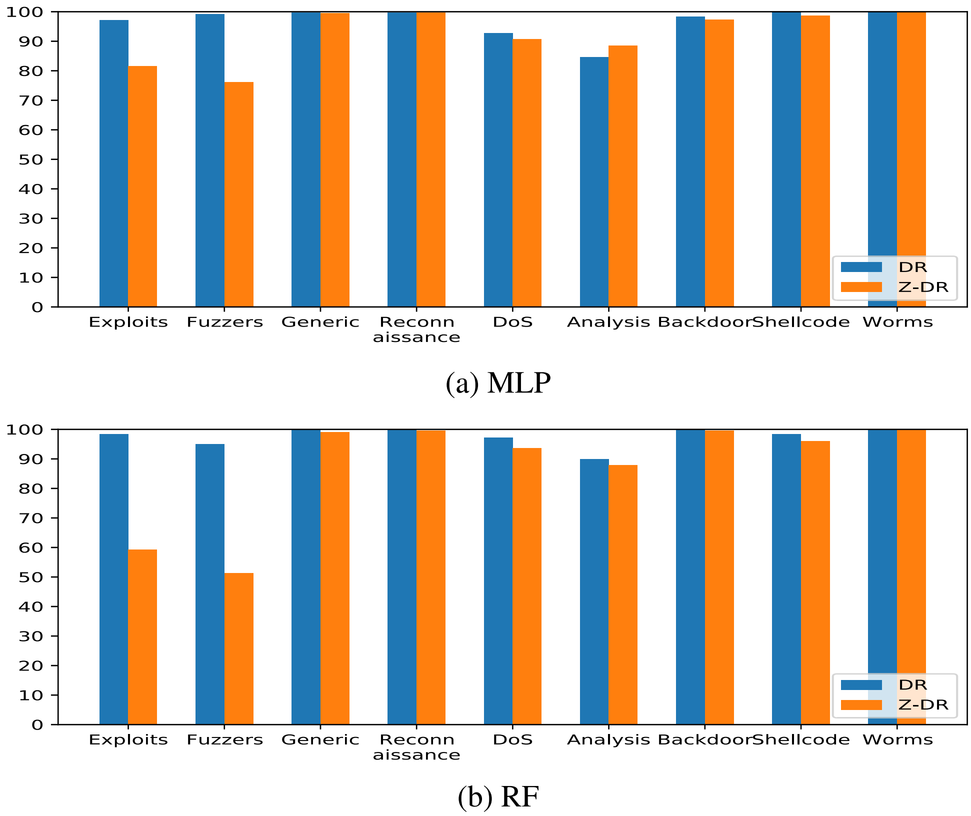
In Tables 4 and 5, the zero-day attack detection performance of the ML models is evaluated using NF-UNSW-NB15-v2, the NetFlow-based edition of the UNSW-NB15 data set. The MLP model is superior to the RF model in detecting zero-day Exploits and Fuzzers attack groups with a detection rate of 82% and 76% compared to 59% and 51%, respectively. The ML models did not successfully apply the learnt semantic attributes of the attack behaviour to relate the Exploits and Fuzzers zero-day attacks as malicious traffic. Attacks such as Generic, Reconnaissance, Backdoor, and Shellcode present a significantly lower cybersecurity risk to organisations protected by ML-based NIDS when observed for the first time as zero-day attacks. The utilised models correctly detected close to 100% of their data samples as intrusive traffic. Moreover, the DoS and Analysis attack groups were slightly harder to detect, as both ML models detected around 90% of their data samples.

Most of the attacks in the NF-UNSW-NB15-v2 data set were reliably detected using the two ML models in a zero-day attack scenario. The learning models successfully utilised the learnt information from known attacks to detect zero-day attack types. However, the Exploits and Fuzzers attack scenarios seem harder to detect if the ML models encounter them as zero-day attacks. The MLP and RF models achieve average Z-DR values of 92.45% and 87.37%, respectively. The UNSW-NB15 and NF-UNSW-NB15-v2 data sets contain the same attack groups and differ only in their respective feature sets. The NetFlow-based feature set of NF-UNSW-NB15-v2 results in an increased Z-DR of around 7% for each of the two ML models. This demonstrates an advantage of using NetFlow-based features in the detection of zero-day attack scenarios.

In Fig. 4, the detection rate of each attack group in the NF-UNSW-NB15-v2 data set is shown for known and zero-day attack scenarios. Figure 4a and 4b shows the performance of the MLP and RF models, respectively. In this data set, a significant drop in detection rates is observed for the Exploits and Fuzzers attack groups, with an average decrease of 28% and 35%, respectively, for the two ML models in a zero-day attack scenario. The ML models could detect the attacks for the rest of the attack groups; however, the DoS and Analysis were slightly sophisticated in their detection, even in a known attack scenario.

Overall, effective Z-DRs have been achieved by both ML models on most of the zero-day attack data samples. This demonstrates the efficiency of the proposed technique and increases the motivation to adopt ML-based NIDSs in securing organisational parameters. However, the Fuzzers attack group is challenging for such systems to detect in a zero-day scenario. The Fuzzers group contains attack scenarios

Fig. 4 Comparison between DR vs. Z-DR of attacks in NF-UNSW-NB15-v2



(a) MLP

(b) RF

in which the attacker sends a large amount of random data, causing a system to crash while aiming to discover security vulnerabilities. While from a security perspective, network scanning traffic often appears similar to benign traffic with an increased volume [36], during the next Section, we analyse the statistical distribution of the Fuzzers attack data samples.

5.2 Analysis

In order to investigate the results provided in the previous subsection, particularly the low Z-DRs of some attack classes, this section examines the distribution differences of features between the training and test sets, i.e. where all attacks are seen (training set) and where there is an unseen attack (testing set). Since the main objective of this analysis was to find any possible differences between the ZSL training and testing sets, statistical measures that could identify differences between (feature) distributions were explored.

The *Wasserstein Distance (WD)* metric is a commonly used tool in the ML community, which has been successfully used in [37] for quantifying the feature distribution distances. The WD, also known as *Earth Mover distance*, is mathematically defined as a distance function of two probability distributions u and v in Eq. 7 [38]:

$$W(u, v) = \inf_{\gamma \in \Gamma(u,v)} \int_{\mathbb{R}\times\mathbb{R}} |x - y| d\gamma(x, y) \tag{7}$$

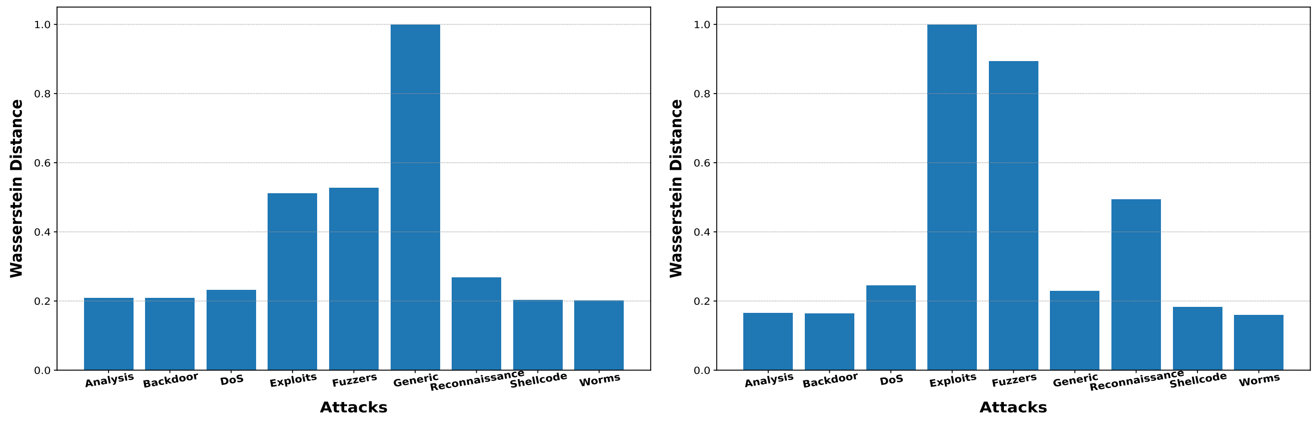
where $\Gamma(u, v)$ is the set of (probability) distributions on $\mathbb{R} \times \mathbb{R}$ where u and v are its first and second factor marginals. $\gamma(x, y)$ can be interpreted as a transport plan/function that gives the amount of mass to move from each x to y to transport u to v, subject to the following constraints:

$$\begin{cases} \int \gamma(x, y)dy = u(x) \\ \int \gamma(x, y)dx = v(y) \end{cases} \tag{8}$$

this indicates that for an infinitesimal region around x, the total mass moved out must be equal to $u(x)dx$. Similarly, for an infinitesimal region around y, the total mass moved in must be equal to $v(y)dy$.

Accordingly, we use WD as the comparison metric and conducted a series of experiments to investigate the differences in the feature distributions of the training and test sets in 9 different zero-day scenarios (one per attack class in each data set). In each scenario, after selecting the training (D^z_{tr}) and testing (D^z_{tst}) sets, the distribution of each feature (except the flow identifier features that were removed in the preprocessing stage) was compared across the two sets using the WD metric (i.e. $W(D^z_{tr}, D^z_{tst})$ in the form of Eq. 7 notation).

The method is performed by measuring the WD between the set of known attacks and the set, including the zero-day



(a) UNSW-NB15

(b) NF-UNSW-NB15-v2

Fig. 5 Average WD of distributions of features (averaged over 45 features) in the train vs. test sets, $W(D_{tr}^{z}, D_{tst}^{z})$, of each zero-day attack, a) UNSW-NB15 data set, and b) NF-UNSW-NB15-v2 data set

attack. Hence, a WD value corresponding to each feature of the data set was obtained for each zero-day attack scenario (45 WD values per each zero-day attack scenario). A higher WD value for a feature indicates a more distinctive distribution between the training sets (D_{tr}^{z}) and testing (D_{tst}^{z}) sets of the corresponding zero-day attack.

Figure 5a and 5b shows the average WD value of the distribution of features (averaged over 45 features) for each zero-day attack scenario, for the UNSW-NB15 and NF-UNSW-NB15-v2 data sets, respectively. In both figures, each column corresponds to an unseen/zero-day attack scenario. Accordingly, the value of WD in each column is the average of 45 WD of the distribution of features in the training and test sets in that zero-day attack scenario. As can be seen, most of the unseen/zero-day attacks have a low WD value of around 0.2, which indicates the overall feature distributions have been similar between the training and testing sets in these zero-day attack scenarios. This shows that these unseen/zero-day attacks have had similar statistical feature distributions to the seen attacks, i.e. attacks present in the training set.

Due to the similarity in the attack types, it is expected to see a higher zero-day detection performance. This mainly includes the Analysis, Backdoor, DoS, Reconnaissance, Shellcode, and Worms attacks. Taking into account Tables 2, 3, 4, and 5, the Z-DR values of these attacks indicate that these attacks are detected with a high detection rate in a zero-day attack scenario. Our results also show a minor degradation in their Z-DR values compared to their (non-zero-day) DR, using the same ML model.

Our analysis using the WD between feature distributions of different attack classes provides a solid explanation of the results presented and is consistent with the main findings of this paper. Overall, the WD function has identified several attack groups with a unique malicious pattern compared to the remainder of the attacks. This matches the results in this paper, as there is a significant difference between their Z-DR and DR values. Therefore, their detection as zero-day attacks using an ML-based NIDS will be challenging from an ML perspective. More studies are required to improve ML-based NIDSs in detecting unique attack behaviour related to sophisticated attacks.

6 Conclusion

A novel ZSL-based framework has been proposed to evaluate the performance of ML-based NIDSs in the recognition of unseen attacks, also known as zero-day attacks. In the attribute learning stage, the model learns the distinguishing attributes of the attack traffic using a set of known attacks. This is accomplished by mapping relationships between the network data features and semantic attributes. In the inference stage, the model is required to associate the relationship of the known attack behaviour to detect a zero-day attack. Using our proposed methodology, two well-known ML models have been designed to evaluate their ability to detect each attack present in the UNSW-NB15 and NF-UNSW-NB15-v2 data sets as a zero-day attack. The results demonstrate that while most attack classes have high Z-DR values, certain attack groups identified in this paper were unreliably detected as zero-day threats. The results presented in this paper were further analysed and confirmed using the WD technique, in which the statistical differences in feature distributions have been directly correlated with the WD and Z-DR metrics. The ability of zero-day attack detection is an essential feature of ML-based NIDSs and is critical for increased practical deployment in production networks. However, this vital issue has attracted only relatively limited attention in the research literature. We hope that the work presented in this paper provides a basis and motivation for further research.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions No funding was received to assist with the preparation of this manuscript.

Research data policy and data availability statements All data generated or analysed during this study are included in this published article Sarhan, M., Layeghy, S. & Portmann, M. Towards a Standard Feature Set for Network Intrusion Detection System Datasets. Mobile Netw Appl 27, 357–370 (2022). https://doi.org/10.1007/s11036-021-01843-0

Declarations

Conflict of interest The authors have no competing interests to declare relevant to this article's content.

Human and animal participants This article does not contain any studies with human participants or animals performed by any authors.

References

1. Ghahramani, Z.: Probabilistic machine learning and artificial intelligence. Nature **521**(7553), 452–459 (2015)
2. Panch, T., Szolovits, P., Atun, R.: Artificial intelligence, machine learning and health systems. J. Glob. Health **8**(2) (2018)
3. Koza, J. R., Bennett, F. H., Andre, D., Keane, M. A.: Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming, pp. 151–170. Springer Netherlands, Dordrecht (1996)
4. Najafabadi, M.M., Villanustre, F., Khoshgoftaar, T.M., Seliya, N., Wald, R., Muharemagic, E.: Deep learning applications and challenges in big data analytics. J. Big Data **2**(1), 1–21 (2015)
5. Bloomfield, R., Khlaaf, H., Conmy, P.R., Fletcher, G.: Disruptive innovations and disruptive assurance: assuring machine learning and autonomy. Computer **52**(9), 82–89 (2019)
6. Buczak, A.L., Guven, E.: A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Commun. Surv. Tutorials **18**(2), 1153–1176 (2015)
7. Alrashdi, I., Alqazzaz, A., Aloufi, E., Alharthi, R., Zohdy, M., Ming, H.: Ad-iot: Anomaly detection of iot cyberattacks in smart city using machine learning. In: 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), pp. 0305–0310, IEEE (2019)
8. Dua, S., Du, X.: Data mining and machine learning in cybersecurity. CRC press (2016)
9. Apruzzese, G. Colajanni, M., Ferretti, L., Guido, A., Marchetti, M.: On the effectiveness of machine and deep learning for cyber security. In: 2018 10th International Conference on Cyber Conflict (CyCon), pp. 371–390, IEEE (2018)
10. Mukherjee, B., Heberlein, L.T., Levitt, K.N.: Network intrusion detection. IEEE Netw. **8**(3), 26–41 (1994)
11. Kumar, V., Sangwan, O.P.: Signature based intrusion detection system using snort. Int. J. Comput. Appl. Inf. Technol. **1**(3), 35–41 (2012)
12. Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., Vázquez, E.: Anomaly-based network intrusion detection: techniques, systems and challenges. comput. Security **28**(1–2), pp. 18–28 (2009)
13. Bilge, L., Dumitraş, T.: Before we knew it: an empirical study of zero-day attacks in the real world. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security, pp. 833–844 (2012)
14. Stellios, I., Kotzanikolaou, P., Psarakis, M.: Advanced persistent threats and zero-day exploits in industrial internet of things. In: Security and Privacy Trends in the Industrial Internet of Things, pp. 47–68, Springer (2019)
15. Mell, P., Grance, T.: Use of the common vulnerabilities and exposures (cve) vulnerability naming scheme, tech. rep., National Inst of Standards and Technology Gaithersburg MD Computer Security Div (2002)
16. Ganame, K., Allaire, M. A., Zagdene, G., Boudar, O.: Network behavioral analysis for zero-day malware detection–a case study. In: International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments, pp. 169–181, Springer (2017)
17. Sinclair, C., Pierce, L., Matzner, S.: An application of machine learning to network intrusion detection. In: Proceedings 15th Annual Computer Security Applications Conference (ACSAC'99), pp. 371–377, IEEE (1999)
18. S. Sahu and B. M. Mehtre, Network intrusion detection system using j48 decision tree. In: 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 2023–2026, IEEE (2015)
19. Xian, Y., Schiele, B., Akata, Z.: Zero-shot learning-the good, the bad and the ugly. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4582–4591 (2017)
20. Wang, W., Zheng, V.W., Yu, H., Miao, C.: A survey of zero-shot learning: settings, methods, and applications. ACM Trans. Intell. Syst. Technol. (TIST) **10**(2), 1–37 (2019)
21. Zhang, Z., Liu, Q., Qiu, S., Zhou, S., Zhang, C.: Unknown attack detection based on zero-shot learning. IEEE Access **8**, 193981–193991 (2020)
22. Sommer, R., Paxson, V.: Outside the closed world: on using machine learning for network intrusion detection. In: 2010 IEEE Symposium on Security and Privacy, pp. 305–316, IEEE (2010)
23. Casas, P., Mazel, J., Owezarski, P.: Unsupervised network intrusion detection systems: detecting the unknown without knowledge. Comput. Commun. **35**(7), 772–783 (2012)
24. Holm, H.: Signature based intrusion detection for zero-day attacks:(not) a closed chapter?. In: 2014 47th Hawaii International Conference on System Sciences, pp. 4895–4904, IEEE (2014)
25. Hindy, H., Atkinson, R., Tachtatzis, C., Colin, J.-N., Bayne, E., Bellekens, X.: Utilising deep learning techniques for effective zero-day attack detection. Electronics **9**(10), 1684 (2020)
26. Li, Z., Qin, Z., Shen, P., Jiang, L.: Zero-shot learning for intrusion detection via attribute representation. In: International Conference on Neural Information Processing, pp. 352–364, Springer (2019)
27. Kumar, V., Sinha, D.: A robust intelligent zero-day cyber-attack detection technique. Complex Intell. Syst. **7**(5), 2211–2234 (2021)

28. Siddique, K., Akhtar, Z., Aslam Khan, F., Kim, Y.: Kdd cup 99 data sets: A perspective on the role of data sets in network intrusion detection research. Computer **52**(2), 41–51 (2019)
29. Felix, R., Harwood, B., Sasdelli, M., Carneiro, G.: Generalised zero-shot learning with domain classification in a joint semantic and visual space. In: 2019 Digital Image Computing: Techniques and Applications (DICTA), pp. 1–8, IEEE (2019)
30. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
31. Hinton, G. E.: Connectionist learning procedures. Mach. learn., pp. 555–610, Elsevier (1990)
32. Breiman, L.: Some properties of splitting criteria. Mach. Learn. **24**(1), 41–47 (1996)
33. Agarap, A. F.: Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375 (2018)
34. Moustafa, N., Slay, J.: Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In: 2015 Military Communications and Information Systems Conference (MilCIS), pp. 1–6, IEEE (2015)
35. Sarhan, M., Layeghy, S., Moustafa, N., Portmann, M.: Towards a Standard Feature Set of NIDS Datasets. arXiv preprint arXiv:2101.11315 (2021)
36. Corchado, E., Herrero, Á.: Neural visualization of network traffic data for intrusion detection. Appl. Soft Comput. **11**(2), 2042–2056 (2011)
37. Layeghy, S., Gallagher, M., Portmann, M.: Benchmarking the Benchmark - Analysis of Synthetic NIDS Datasets. arXiv preprint arXiv:2104.09029 (2021)
38. Ramdas, A., Trillos, N.G., Cuturi, M.: On wasserstein two-sample testing and related families of nonparametric tests. Entropy **19**(2), 47 (2017)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.