Internet of Things 16 (2021) 100462

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Review article

Intrusion detection based on Machine Learning techniques in computer networks

Ayesha S. Dina, D. Manivannan [*]

University of Kentucky, Department of Computer Science, Lexington, Kentucky, 40508, USA

A R T I C L E I N F O

Keywords:
Network security
Computer security
Cybersecurity
Intrusion detection
Intrusion prevention
Machine learning

A B S T R A C T

Intrusions in computer networks have increased significantly in the last decade, due in part to a profitable underground cyber-crime economy and the availability of sophisticated tools for launching such intrusions. Researchers in industry and academia have been proposing methods and building systems for detecting and preventing such security breaches for more than four decades. Solutions proposed for dealing with network intrusions can be broadly classified as signature-based and anomaly-based. Signature-based intrusion detection systems look for patterns that match known attacks. On the other hand, anomaly-based intrusion detection systems develop a model for distinguishing legitimate users' behavior from that of malicious users' and hence are capable of detecting unknown attacks. One of the approaches used to classify legitimate and anomalous behavior is to use Machine Learning (ML) techniques. Several intrusion detection systems based on ML techniques have been proposed in the literature. In this paper, we present a comprehensive critical *survey of ML-based intrusion detection approaches* presented in the literature in the last ten years. This survey would serve as a supplement to other general surveys on intrusion detection as well as a reference to recent work done in the area for researchers working in ML-based intrusion detection systems. We also discuss some open issues that need to be addressed.

1. Introduction

Security breaches have increased significantly in the last decade, due in part to a profitable underground cyber-crime economy. Tools and techniques used in such security breaches have also become very sophisticated. Traditional tools such as anti-virus software and firewalls for preventing cybersecurity breaches cannot help in preventing or detecting such sophisticated attacks. One way to deal with this problem is to install hardware/software to continuously monitor the network for intrusions/attacks. Research in intrusion detection started in 1972 [1] when James Anderson published his report on the need for detecting breaches in computer systems [2]. Since then, several monitoring systems such as Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS) (which are also called Intrusion Detection and Prevention Systems (IDPS)) have been proposed and implemented. These systems can further be classified as host-based, network-based and hybrid [3]. These systems can be centralized, distributed or hybrid, depending on how events related to intrusions/attacks are collected, processed and acted upon. Each of these approaches has its advantages and disadvantages in terms of cost, performance, and other metrics.

* Corresponding author.
E-mail addresses: adi252@uky.edu (A.S. Dina), mani@cs.uky.edu (D. Manivannan).
URL: http://www.cs.uky.edu/~manivann (D. Manivannan).

https://doi.org/10.1016/j.iot.2021.100462
Received 29 August 2021; Received in revised form 7 October 2021; Accepted 8 October 2021
Available online 27 October 2021

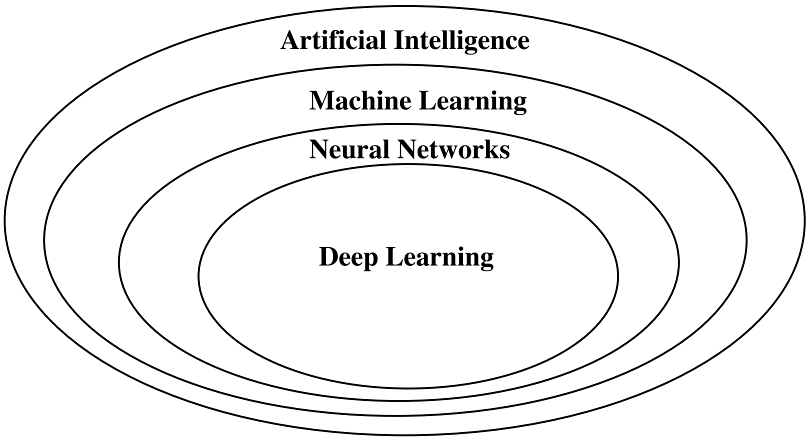

Fig. 1. Relationship between AI, ML, ANN and DL.

IDSes can be further classified as signature-based and anomaly-based. Signature-based IDSes use the signatures of known attacks to determine attacks. Such IDSes are not capable of detecting new types of attacks or zero-day attacks. On the other hand, anomaly-based IDSes collect data relating to the behavior of legitimate users and then current observed behavior is analyzed to determine if those are that of legitimate users or of malicious users; hence these types of IDSes are capable of detecting unknown attacks. However, anomaly-based IDSes generally result in high false-positive rate while signature-based IDSes usually have more false-negatives. Anomaly-based approach requires developing a model of legitimate users' behavior by first collecting and monitoring data related to normal operation of system in the training phase. Then, current observed behavior is compared with the model to classify it as legitimate or anomalous. Next, we discuss some of the models used for anomaly detection.

Artificial Intelligence (AI), Machine Learning (ML), Neural Networks (NN) (also known as Artificial Neural Networks or ANN) and Deep Learning (DL) are related fields; each of these fields, in this order, can be thought of as a sub-field of the previous one as depicted in Fig. 1. In particular, ANN and DL are sub-fields of ML. John McCarthy, winner of Turing award and one of the founders of the discipline AI, defines AI as "It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable". In ML, algorithms are used to parse data related to solving a specific problem (e.g., intrusion detection), learn from it and then this learning is used to discover patterns that are useful for solving the problem. NNs are a network of neurons. Neurons are processing units which model the functioning of a biological neuron, typically by computing weighted average of its inputs using a variety of algorithms. In a NN, data passes through several layers of interconnected neurons. Haykin [4] defines NN as "Massively parallel combination of simple processing units which can acquire knowledge from the environment through a learning process and store the knowledge in its connections".

NNs, in general, can be classified into feed-forward, recurrent, and convolutional Networks. In feed-forward NNs, connections between the neurons do not form a cycle. In recurrent NNs (RNNs), connections between neurons form a directed graph. In RNNs, a neuron's output can serve as input to the same neuron. Convolutional NNs (CNNs), widely used in image processing, can take images as input and be able to differentiate images by assigning weights and biases to various aspects of the image. Some researchers consider all these NN models as sub-fields of DL while others define Deep learning as an NN with three or more layers.

Following are some ML based approaches tried by various researchers for intrusion detection: Artificial Neural Networks, Association Rules and Fuzzy Association Rules, Bayesian Network, Clustering, Decision Trees, Ensemble Learning, Evolutionary Computation, Hidden Markov Models, Inductive Learning, Naïve Bayes, Sequential Pattern Mining, and Support Vector Machine [5]. We discuss some of these approaches in Section 2. In this paper, our focus is to present a critical survey various intrusion detection approaches based on ML (which includes NN and hence DL), presented in literature. There are many surveys on intrusion detection techniques which we discuss in Section 4. Unlike other surveys, this survey focuses *only* on ML based intrusion detection techniques presented over the last decade.

The rest of the paper is organized as follows: In Section 2, we present some background required. In Section 3, we present a critical survey of the ML based intrusion detection techniques presented in the literature over the last decade. In Section 4, we discuss the related work. In Section 5, we discuss our observations about the surveyed works and also bring out some open issues that remain to be addressed. Section 6 concludes the paper.

2. Background

ML algorithms can be broadly classified as (i) supervised and (ii) unsupervised. Supervised learning algorithms use labeled data in the training phase for determining abnormalities in new data samples. In unsupervised learning, the training data are not labeled; the learning algorithm used tries to group/classify the training data based on some grouping techniques. Signature-based IDSes generally

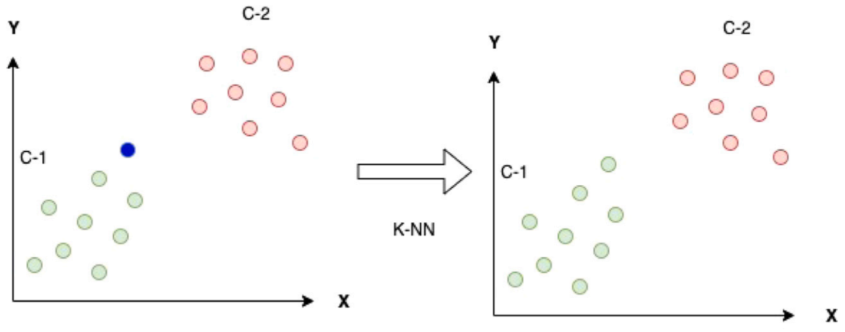

Fig. 2. K-NN algorithm based classification.

use supervised learning algorithms whereas anomaly-based IDSes use unsupervised learning algorithms. Binary classification is the basic classification type where the number of classification labels is two. For intrusion detection, binary classification uses the labels normal and attack (or anomalous, abnormal, etc.). Under Multi-class classification, the number of labels used for classifying data can be more than two. For example, for intrusion detection, data can be labeled as normal, Denial of Service Attack (DoS), User to Root Attack (U2R), Remote to Local Attack (R2L), Probing Attack, etc. Next, we discuss some types of algorithms used for classifying data.

2.1. Some classification techniques

In this subsection, we describe some of the classification techniques used by ML algorithms for classifying and labeling training datasets.

2.1.1. Clustering based classification

Clustering is an unsupervised learning method for discovering patterns in unlabeled data and grouping similar data [6]. There are several algorithms for clustering such as K-nearest neighbors algorithm (K-NN) [7], K-means clustering (which originated from signal processing), Mean-Shift Clustering [8], Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [9], Gaussian Mixture Models [10], Hierarchical Clustering, and others.

For example, basic idea behind K-NN algorithm developed by Fix et al. [7] and later generalized by Cover et al. [11] is illustrated in Fig. 2. This example illustrates how continuous data can be classified using Euclidean distance. In this example, training data are classified into two types (C-1 and C-2). Data of type C-1 are colored green and data of type C-2 are colored red. When a new data point (colored blue) needs to be classified, K-NN selects K neighboring points (K is a user defined constant) and calculates the Euclidean distances between the new data point and the chosen K neighbors. Then, using some algorithm such as majority voting algorithm, it decides whether to classify the new data point as type C-1 or type C-2 data point. In this example, after applying K-NN, the blue data point is classified to be of type C-1 because it has been determined to be closer to majority of the data points in C-1. For discrete variables, such as in natural language processing, Hamming Distance can be used for text classification.

2.1.2. Decision tree based classification

A decision tree is tree structure, in which each leaf node represents a class label and each internal node is a decision node or a chance node. A decision node performs a test on a feature and each branch node of a decision node represents an outcome of the test. Some of the well-known algorithms for building decision trees are ID3 [12], and C4.5 [13]. Both these algorithms use the concept of information entropy for building decision trees from training datasets for labeling data. Generally, using decision trees result in high classification accuracy. Studies have shown that clustering methods along with decision trees can help in reducing processing time [5].

Fig. 3, is an example of a decision tree for labeling TCP connections. In this figure, rectangular shapes represent features and branches represent rules. These are some of the features defined for TCP connections in the NSL-KDD dataset. A TCP `logged_in` feature has a flag associated with it. The flag is set to either zero or one. Zero means unsuccessful login and one means successful login. The feature `serror_rate` represents the percentage of TCP connections from the same host that have "SYN" errors.

2.1.3. Artificial Neural Networks based Classification

Artificial Neural Networks (ANN) are based on the working of human brain and are composed of interconnected artificial neurons capable of certain computations on their inputs [14]. An ANN consists of several layers of neurons. The input data activate the neurons in the first layer of the network. The output of the first layer serves as input to the second layer; the output of the second layer serves as input to the third layer and so on. The input and output layers are called hidden layers. The final output layer of an ANN generates the final classification of training data.

Fig. 4 illustrates the generic structure of an FNN network. An FNN has an input layer and an output layer. It can have any number of hidden layers. The FNN depicted in Fig. 4 has four neurons in its input layer, three neurons in the output layer and has several hidden layers.

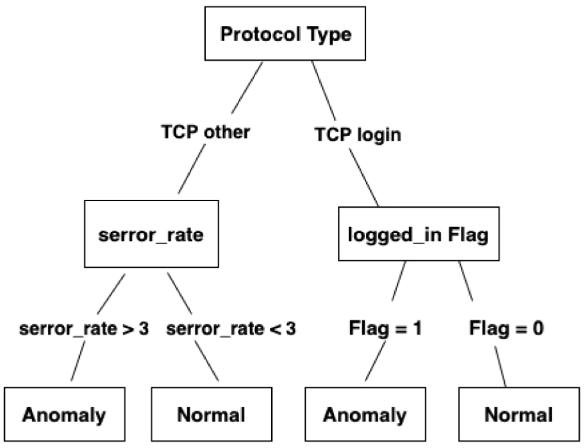

Fig. 3. A Decision Tree (DT).

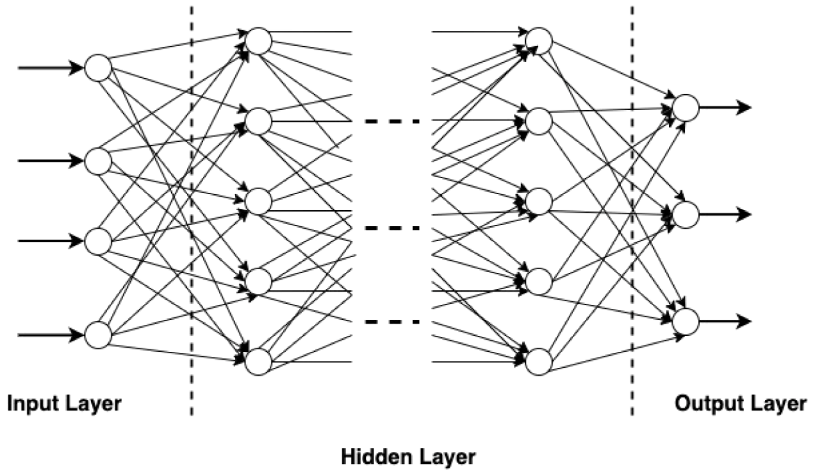

Fig. 4. Feed Forward Neural Network (FNN).

2.1.4. *Other classifiers*

There are also many other classifiers such as the ones based on Naïve Bayes, Association Rules and Fuzzy Association Rules, Ensemble Learning, Evolutionary Computation, Hidden Markov Models, Inductive Learning, Sequential Pattern Mining, Support Vector Machine, etc. [15]. Next, we discuss some of the well known datasets used for training ML based IDSes.

2.2. *Data imbalance problem in datasets*

In many of the datasets used for evaluating ML based IDSes, number of instances for various classes are not equally distributed. i.e., some classes have more instances than others [16]. The classes that have large number of instances are called majority classes and those classes that have small number of instances are called minority classes. The ratio between the number of instances in minority class and the number of instances in a majority class could be as high as 1:100, 1:1000 or even 1:1000000 [17]. Many of the ML-based algorithms used for classification of data are based on the assumption that there are equal number of instances for each class. So, if data are imbalanced in the dataset, the interpretation of performance metrics of ML-based IDSes could be affected, as we point out in Section 2.4.

2.2.1. *Solution for data imbalance problem*

Several techniques [17] are used for solving the data imbalance problem in datasets. At the data level, most commonly used technique is sampling with replacement. It could be random sampling with replacement where each sample has equal chance to be selected or **Stratified Sampling (SS)** [18] wherein prior knowledge of data is needed to create homogeneous subgroups (called strata), based on their similarities and then selecting random sample from each group. Stratified random sampling helps in obtaining a sample that best represents the entire data. At algorithm level [17], weight of classes can be adjusted to solve this problem also.

Table 1
Class distribution of the KDD99 dataset.

Class	Training	(%)	Testing	(%)
Normal	972781	19.85	60593	19.48
DoS	3883390	79.27	231455	74.41
Probe	41102	0.83	4166	1.33
U2R	52	0.001	245	0.07
R2L	1106	0.02	14570	4.68

Table 2
Class distribution of the NSL-KDD dataset.

Class	Training_{20}	(%)	Training_{+}	(%)	Testing	(%)
Normal	13449	53.3	67343	53.5	9711	43.1
DoS	9234	36.7	45927	36.4	7458	33.1
Probe	2289	9.1	11656	9.3	2421	10.7
U2R	11	0.04	52	0.041	67	0.3
R2L	209	0.83	995	0.78	2887	12.8

Under-fitting and over-fitting: Under-fitting occurs when a ML model cannot capture the underlying structure of the data adequately. For example, if we try to fit a linear model to non-linear data, under-fitting would occur. Such a model would fail to capture the trend of the data. This happens when there is insufficient data to train the ML model. On the other hand, over-fitting occurs when an ML model learns every detail of the training dataset including noise; this may fail to fit unseen data. To overcome this, re-sampling techniques like K-fold cross validation can be used.

2.3. Datasets used for training machine learning models

In this section, we discuss some datasets that are commonly used to evaluate IDSes based on machine learning techniques.

2.3.1. KDD99

Knowledge Discovery in Databases (KDD) [19], published in 1999, is considered as one of baseline datasets to evaluate different IDSes including ML-based IDSes. This dataset was prepared by Stolfo et al. [20,21]. The KDD99 dataset was prepared based on DARPA 98 [22]. Each record in KDD99 dataset has 41 features and records are either tagged as normal or an attack type. There are basically four type of attacks in the dataset [21]. They are **Denial of Service Attack (DoS)**, **User to Root Attack (U2R)**, **Remote to Local Attack (R2L)**, and **Probing Attack**. The dataset has 78% and 75% duplicate records in training and testing datasets respectively. The U2R (52 in training dataset) and R2L (1106 in training dataset) attacks are insufficient to train a ML model and that makes KDD99 dataset imbalanced. Table 1 contains information about the data distribution for each class the dataset KDD99 [23].

2.3.2. NSL-KDD

Network Socket Layer-Knowledge Discovery in Databases (NSL-KDD) [24] is another commonly used dataset. This dataset was prepared by Tavallaee et al. [21] based on KDD99 dataset. Unlike KDD99 dataset, NSL-KDD dataset does not contain any duplicate records [21] which makes it more suitable for evaluating ML-based IDSes. However, like KDD99, data distribution among various classes is imbalanced. Information about data distribution in various classes in NSL-KDD training set is given in Table 2 [25]. As we can see, the training sample size of U2R and R2L are significantly low.

2.3.3. UNSW-NB15

Moustafa et al. [26] proposed a new dataset in 2015 to evaluate IDSes. It tries to capture network traffic data as realistic as possible. It used tcpdump tool [27] for capturing raw network traffic. Argus and Bro-IDS [28,29] were used to extract key features from traffic data [30]. The dataset has 49 features and $2,540,044$ records. The dataset contains data related to the following nine attack classes: **Fuzzers, Analysis, Backdoor, DoS, Exploit, Generic, Reconnaissance, Shellcode, and Worm.** However, the dataset is not balanced as attack classes Analysis, Backdoor, Shellcode, and Worm have low training samples. Data distribution for various attack classes in UNSW-NB15 is given in Table 3 [31]. The dataset is available at [32]

2.4. Performance metrics used for evaluating intrusion detection systems

In this subsection, we present various metrics used for evaluating the performance of IDSes. Let TP be the number of true positive events detected by an IDS (i.e., the number of events/actions detected/declared by the IDS as intrusions which are actually intrusions). Let TN be the number of true negative events detected by an IDS (i.e., the number of events/actions detected/declared by the IDS as normal which are actually normal). Let FP be the number of false positive events detected by an IDS (i.e., the number of events/actions detected/declared by the IDS as intrusions which are actually not intrusions). Let FN be the number of false negative events detected by an IDS (i.e., the number of events/actions detected/declared by the IDS as normal which are actually intrusions). Performance of an IDS is generally evaluated based on the following metrics:

Table 3
Data distribution for various attack classes in the UNSW-NB15 dataset.

Class	Training	(%)	Testing	(%)
Normal	11200	32	7400	45
Generic	8000	23	3774	23
Exploits	6679	19.05	2226	13.5
Fuzzers	3637	10.37	1212	7.36
DoS	2453	6.99	818	4.97
Reconnaissance	2098	5.98	699	4.25
Analysis	400	1.14	135	0.82
Backdoor	349	1	117	0.71
Shellcode	227	0.65	76	0.46
Worms	26	0.07	9	0.05

Table 4
Confusion matrix or table of confusion.

Predicted class	Actual class	
	Positive	Negative
Positive	**TP**	FP
Negative	FN	**TN**

- **True Positive Rate (TPR):** This is calculated as **TPR** = $\frac{TP}{TP+FN}$. This is the probability that an actual intrusion will be declared as an intrusion by the IDS. This metric is also called the **sensitivity or detection rate** of the IDS.
- **True Negative Rate (TNR):** This is calculated as **TNR** = $\frac{TN}{TN+FP}$. This is the probability that a non-intrusive action will be declared as a non-intrusive action by the IDS. This metric is also called the **specificity** of the IDS.
- **False Positive Rate (FPR):** This is calculated as **FPR** = $\frac{FP}{FP+TN}$. In other words, FPR is the probability that a false alarm will be raised (i.e., the probability an IDS will detect an action/event as intrusion while it is actually not an intrusion).
- **False Negative Rate (FNR):** This is calculated as **FNR** = $\frac{FN}{FN+TP}$. This is the probability that a non-intrusive action/event will be detected as an intrusive action/event by the IDS. This metric is also called **miss rate** of the IDS.

Following are some more metrics derived from TP, TN, FP and FN:
Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$
Precision = $\frac{TP}{TP+FP}$
Recall = $\frac{TP}{TP+FN}$
F_1-Score = $2 * \frac{\textbf{Precision}*\textbf{Recall}}{\textbf{Precision}+\textbf{Recall}}$
Accuracy is the proportion of events that are correctly predicted by the algorithm (as normal or intrusion in binary classification). **Precision** is the proportion of events that are predicted by the algorithm as intrusion are actually intrusions. **Recall** is the proportion of actual intrusions that were predicted as intrusions by the algorithm. **F_1-Score** is the harmonic mean (reciprocal of the arithmetic mean of the reciprocals) of **Precision** and **Recall**.

2.4.1. Confusion matrix

The above metrics may not be very useful in evaluating an algorithm if the data set is imbalanced. **Accuracy** will yield misleading results if the data set is imbalanced. For example, suppose there were 90 events that are intrusions and 10 events are normal in the dataset; if the algorithm classifies all the events as intrusions, the overall accuracy of the algorithm would be 90%, sensitivity would be 100% for events that are not intrusions but sensitivity would be 0% for events that are intrusions.

So, confusion matrix, which is a table layout of observed experimental results of an algorithm, can help in visualizing the performance of the algorithm in such situations. The confusion matrix looks like Table 4. This table contains the number of true positives, true negatives, false positives and false negatives. This allows more detailed analysis than merely calculating metrics like **Accuracy**. Some of the acronyms used in this paper are given in Table 5.

3. Intrusion detection techniques based on machine learning

In this section, we present a critical survey of the papers on intrusion detection that use different ML techniques, published in the last ten years. Most of these papers used supervised learning with binary and/or multi-class classification. They used many publicly available benchmark datasets to evaluate their techniques. We discussed some of the widely used datasets in Section 2.3. We classify the papers based on the classification type used in the paper for labeling datasets (binary or multi-class) and the datasets used for evaluating the approach proposed in the paper.

Table 5
Acronyms used in this paper.

Acronym	Expansion	Detailed Description
PCA	Principle Component Analysis	PCA is commonly used to reduce the dimension of a large dataset [33]. It is a descriptive method and does not require any assumption about data distribution.
SVM	Support Vector Machine	SVM can learn from sample examples and assign labels to unknown objects [34]. It can help in solving classification and regression problems [35]. It supports standard kernel functions and lets the user choose their own function.
ANN	Artificial Neural Network	First developed in the 1950s, inspired by the structure and functioning of brain.
RNN	Recurrent Neural network	A class of ANNs.
DT	Decision Tree	A flowchart-like structure (explained earlier) used for classification of data.
RF	Random Forest	Also called Random Decision Forest, an ensemble learning method for classification.
LR	Linear Regression	A linear approach for modeling the relationship between a scalar response and one or more dependent variables.
DNN	Deep Neural Network	A type of ANN.
DBN	Deep Brief Network	A class of DNNs, composed of multiple layers of hidden units, with connections between the layers.
AE	Auto Encoder	It is a type of ANN used to learn efficient codings of unlabeled data.
VAE	Variational Auto Encoder	A type of ANN.
CNN	Convolutional Neural Network	A type of ANN.
FNN	Feed-forward Neural Network	A type of ANN in which connections between the nodes do not form a cycle.
MLP	Multilayer Perceptron	A class of feed-forward ANNs.
ELM	Extreme Learning Machine	A type of feed-forward ANN.
GA	Genetic algorithm	It is a method for solving optimization problems that mimics biological evolution.
IoT	Internet of Things	Network of objects connecting and exchanging data with other devices and systems over the Internet.
NB Classifiers	Naïve Bayes Classifiers	A family of classifiers based on Bayes' Theorem.
K-NN	K - Nearest Neighbors	A classification method.

3.1. Intrusion detection systems that used binary classification

Binary classification is the basic classification type where the number of classification labels is two. For intrusion detection, binary classification uses the labels normal and attack (or anomalous, abnormal, etc.). In this section, we discuss the papers using binary classification for intrusion detection. We group the papers based on whether they used single data set or multiple datasets for evaluation.

3.1.1. Binary classification based intrusion detection systems that used the dataset KDD99 for evaluation

In this subsection, we discuss the papers that used the dataset KDD99 for evaluation.

The hybrid approach introduced by Elbasiony et al. [36] has the following two parts: misuse (signature-based) detection part and anomaly detection part. It has two phases: online and offline. Online phase is responsible for misuse detection by comparing network traffic with known intrusion patterns (signatures). If there is no match with known intrusion patterns, it will be tagged as a new type of attack and sent to the anomaly detection part. Anomaly detection is accomplished by using weighted K-means algorithm wherein feature importance values are calculated using RF model. Their hybrid framework achieved 98.3% accuracy with 1.6% false positive rate on the KDD99 dataset.

Collaborative Intrusion Detection Systems (CIDSes) which consist of multiple components at the network level and host level for monitoring/analyzing and exchanging events to determine abnormal events have been proposed in the literature. CIDSes have some limitations such as delays due to computational time or complex architecture of the network. To overcome such limitations, Abusitta et al. [37] proposed a CIDS using ML techniques. They introduced stacked Denoising Auto Encoder (SDAE) as an unsupervised feature learning technique to extract robust features. The goal of extracting robust features is to detect intrusion even when the IDS provides partial information. They used logistic regression to perform binary classification. They also tested their model with the data that had incomplete or partial information.

3.1.2. Binary classification based intrusion detection systems that used NSL-KDD dataset for evaluation

Ever et al. [38] employed three ML models, ANN, SVM, and DT. In this study, the goal was to find out the optimal ML technique among ANN, SVM, and DT. They performed two experiments by using 60%, and 70% of the dataset NSL-KDD for training and the rest of the dataset for testing. Their experiment shows that DT achieved 98.84% training accuracy. However, the authors have not shown the accuracy level achieved for the other two ML models.

Begli et al. [39] discussed the challenges in securing remote healthcare monitoring over Internet. They proposed a layered multi-agents based framework for securing such healthcare monitoring. First, they define agents based on patients, nurses, doctors, etc. Then, the agents are grouped into three groups based on their energy capacity, and the amount of data they can hold from different sensors. For each agent group, an IDS based on SVM is designed. They used three IDSes for each group/layer. The intrusion detection success rates of their approach are 95.01%, 50.04%, and 97.2% for first, second, and third IDS respectively.

3.1.3. Binary classification based intrusion detection systems that used UNSW-NB15 dataset for evaluation

Chowdhury et al. [40] proposed a method to effectively detect intrusion based on the network traffic data. They selected three features each time randomly from a features pool, and applied SVM to distinguish between attack and normal traffic. This process was repeated until all possible combinations of the features have been covered. Their method achieved 98.76% accuracy on the dataset UNSW-NB15.

3.1.4. Intrusion detection systems that used CIDDS-001 dataset for evaluation

Abdulhammed et al. [41] addressed the issue of imbalanced distribution of classes in the datasets. They proposed the following four sampling techniques to balance different classes of data in the dataset. The techniques are [42,43]: (i) Up-sampling the minority class, (ii) Down-sampling the majority class, (iii) Spread sub-sample, and (iv) Class balancer. They used coburg intrusion detection dataset-001 (CIDDS-001) [44] for evaluation, and for classification they used DNN, RF, voting [45], VAE [46], and stacking machine learning [47]. They compared their approach with some of the other approaches. Their approach achieved 99.99% accuracy which is higher than the other compared models. They also show how data imbalance problem impacts the final results.

3.1.5. Binary classification based intrusion detection systems that used multiple datasets for evaluation

Bhamare et al. [48] point out the need for further research in the field of supervised learning for its application to cloud security and presented a binary classification technique. They used two different datasets to determine the robustness of the existing ML models. They trained the ML models with one dataset and tested it with another dataset. They labeled the data in the datasets as normal and attack/anomaly. They used three ML models — SVM, LR, and DT. For training they used UNSW-NB15 [26,30], created by Australian Center for Cyber Security (ACCS) and for testing the robustness of the models, they used ISOT [49] from French chapter of the honeynet project. However, the training dataset is somewhat imbalanced. Only 32% of the total training dataset belong to normal class.

Injadat et al. [50] proposed an ML based intrusion detection framework with multi-stage optimization. To study different optimization algorithms, they designed a framework with multiple stages. In the data preprocessing step, they used Z-score for data normalization and SMOTE algorithm [51] to solve the data imbalance issue in the dataset. SMOTE algorithm was designed to synthetically increase the number of data samples in the minority classes (classes of data that have significantly low sample size) [51]. For feature selection, they used information gain-based feature selection [52,53] and correlation-based feature selection [54,55]. In the final step, they used K-NN and RF as classifiers. Random search (RS), meta-heuristic algorithm such as particle swarm optimization (PSO), GA, and Bayesian optimization (BO) were used as hyper-parameter optimization algorithms. To evaluate their framework, they used Canadian Institute of Cybersecurity's IDS 2017 (CICIDS 2017) [56], and UNSW-NB15 datasets. According to their evaluation, combination of BO with RF outperformed the other classifier combinations. This combination achieved 99.99% detection accuracy. The authors compared their approach with some of the other approaches. Accuracy achieved by their approach has been shown to be 1%–2% higher than that of other compared approaches.

Andresini et al. [57] proposed the idea of multi-channel feature representation of network traffic. In the first step, they trained two AEs separately for normal and attack classes. The goal was to get two different representations of each sample. In the next step, they augmented the original sample with these two reconstructed samples and fed into a one dimensional CNN. Using softmax classifier, they labeled the data as normal and attack. They used KDD99, UNSW-NB15, and CICIDS 2017 datasets to assess the performance of their method. Their method performed better on two out of the three datasets. The accuracy of this technique are 92.49%, 93.49%, and 97.90% on KDD99, UNSW-NB15, and CICIDS 2017 respectively. Table 6 summarizes the merits and demerits of the papers that used binary classification as data classification technique, discussed in this section.

3.2. Intrusion detection systems that used multi-class classification

Under multi-class classification, data in the training dataset are classified into several disjoint classes and data belonging to each class is assigned the same label. Data in different classes are assigned different labels. In intrusion detection systems using multi-class classification, besides data being labeled as normal, there are several other types of labels such as Denial of Service Attack (DoS), User to Root Attack (U2R), Remote to Local Attack (R2L), etc. one for each type of attack. As in the previous section, first we discuss the papers that used only one dataset for evaluation and then we discuss papers that used multiple datasets for evaluation.

Table 6
Summary of the papers that used binary classification.

Paper	Year	Method(s) used	Dataset(s) used	Type of classification used	Strengths and weaknesses
Elbasiony et al. [36]	2013	Weighted K-Means and RF	KDD99	Binary	The authors calculated feature importance of each feature in the dataset.
Bhamare et al. [48]	2016	LR, SVM and DT	UNSW-NB15, ISOT	Binary	Tests the robustness of the models. The training dataset used is imbalanced.
Chowdhury et al. [40]	2016	Simulated Annealing and SVM	UNSW-NB15	Binary	The model achieves 98.76% accuracy with a low false positive rate of 0.09%. Their model is compared only with the Naive SVM model with all the features of the dataset. Time complexity of this model is high due to its iterative nature of selecting random features from the pool.
Abusitta et al. [37]	2019	SDAE and Logistic Regression	KDD99	Binary	The model can predict anomaly even when the IDS provides partial or incomplete information.
Ever et al. [38]	2019	SVM, ANN, and DT	NSL-KDD	Binary	The goal is to select the optimal method from the chosen set of three methods. Their experimental results show only training accuracy but not testing accuracy.
Abdulhammed et al. [41]	2019	DNN, RF, voting [45], and VAE [46], stacking machine learning [47]	CIDDS-001	Binary	They addressed the data imbalance problem in the dataset by providing some solutions.
Begli et al. [39]	2019	SVM	NSL-KDD	Binary	IDS for remote healthcare monitoring. Their approach is not compared with existing ML based IDSes and has not been evaluated with real data.
Injadat et al. [50]	2020	K-NN, and RF	CICIDS 2017, and UNSW-NB15	Binary	Different optimization algorithms have been explored for reducing time and space complexity. This will help in determining the performance of an IDS in data imbalance scenarios.
Andresini et al. [57]	2020	AE, and CNN	KDD99, CICIDS 2017, and UNSW-NB15	Binary	They used the idea of multi-channel representation of samples using two AEs. CNN is used to fetch possible hidden patterns in the multi-channel representation. The structure and characteristics of the attacks were not well described.

3.2.1. Multi-class classification based intrusion detection systems that used KDD99 dataset for evaluation

Horng et al. [58] used SVM based technique to detect intrusions in the network traffic. They used hierarchical clustering for preprocessing data. Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) [59] was used as hierarchical clustering algorithm. This clustering algorithm helped in reducing the size of training data as SVM cannot take large datasets in its training phase. After significant reduction in the size of training dataset, it was fed into the SVM classifiers to train the SVM model. According to their experimental results, their technique achieved 95.72% accuracy with false positive rate of 0.7%.

Li et al. [60], proposed an IDS for IoT using AI, taking advantage of software defined networks (SDN) [61]. Their approach consists of several stages. In the initial stage, they used Bat Algorithm (BA) [62] with swarm division and binary differential evaluation [63] to acquire optimal features from the original network traffic data. In the final stage, they used RF algorithm to classify network traffic data into different classes. Their evaluation showed that their method had 96.42% accuracy and 0.98% false positive rate.

In order to reduce the number of false positives and false negatives in intrusion detection, Ali et al. [64] introduced an ANN-based model named as PSO-FLN. This model is developed using fast learning network (FLN) [65] with the optimization algorithm called particle swarm optimization (PSO) [66]. They evaluated their model with ELM [67] along with different optimization approaches such as GA, Harmony Search Optimization (HSO) [68], and Ameliorated Teaching Learning Based Optimization (ATLBO) [69]. They repeated these approaches with FLN and came up with PSO-FLN model which outperformed other compared models. PSO-FLN achieved 99.69% accuracy.

3.2.2. Intrusion detection systems that used NSL-KDD dataset for evaluation

Zhang et al. [70], introduced a technique based on both GA and DBN to address security in IoT. GA was used to optimize the number of hidden layers and the number of neurons in a layer for each type of attack. This optimization can help in reducing the complexity of the neural network as well as obtaining an optimal neural network architecture for detecting intrusion based on network traffic. They used NSL-KDD dataset to evaluate their technique. Before applying their technique, they normalized the dataset using min–max normalization as it is difficult to use high dimensional dataset to detect intrusion in real time. This method has higher accuracy on minority class type such as U2R compared to other compared methods. Gao et al. [71] introduced a novel approach to detect intrusion in computer networks. NSL-KDD dataset is not balanced. To resolve it, they proposed MultiTree algorithm using DT in four levels by adjusting the ratio of the class types. They introduced a model where they ensemble some base classifiers using DT, RF, K-NN, and DNN, and make classification decision using their adaptive voting algorithm.

3.2.3. Multi-class classification based intrusion detection systems that used UNSW-NB15 dataset for evaluation

Moualla et al. [72], proposed an intrusion detection system using existing ML techniques. In the initial step, they balanced the dataset using synthetic minority oversampling (SMOTE) [73] algorithm which oversamples the minority classes by creating synthetic samples. In the preprocessing stage, they performed simple data cleaning and one-hot-encoding (transforming categorical features into continuous features); then they normalized the dataset based on z-score. For feature selection, they used extremely randomized trees classifier (Extra Tree classifiers) [74] for each class type to capture optimal features. Then, they used ELM classifier for each class type. Finally, to aggregate the results of all the ELM classifiers, they used fully connected layer followed by a logistic regression layer for deciding each class. Their approach achieved an overall accuracy of 98.43%.

3.2.4. Multi-class classification based intrusion detection systems that used MQTT-IoT dataset for evaluation

Hindy et al. [75] presented a case study to demonstrate the effectiveness of ML techniques for intrusion detection in IoT. They used six ML models LR, NB, K-NN, SVM, DT, and RF to detect intrusions in the system. They used the simulated message queuing telemetry transport protocol (MQTT) [76] based IoT dataset to evaluate the ML models. They also addressed uni-directional and bi-directional flow-based and packet-based features while detecting intrusions in the system. They compared these ML models in terms of their accuracy, precision, recall, and F_1 score. They came to the conclusion that flow-based features are important to detect MQTT-based attacks and packet-based features are good for detecting attacks in traditional networks.

3.2.5. Multi-class classification based intrusion detection systems that used multiple datasets for evaluation

Shone et al. [77] studied the feasibility and sustainability of current approaches in network intrusion detection. As the volume of network data is increasing, analyzing network data in depth is required to make the IDSes more effective. Because of the different types of protocols used in the Internet, we have diverse network data. This also makes the task of distinguishing normal network traffic from attack traffic difficult. They combined deep and shallow learning in their model. They used two layers of Non-symmetric Deep Auto Encoder (NDAE) for unsupervised feature learning. Unlike typical AE, the NDAE does not contain a decoder. RF was used to perform the final classification of the network traffic into normal and attack. The authors evaluated their model using five and thirteen layers of classification using the datasets NSL-KDD and KDD99. They performed 10-fold cross validation to overcome over-fitting and under-fitting problem. Since the datasets contained imbalanced data, false alarm rate was very high in some attack classes. Table 7, summarizes the merits and demerits of the papers that used multi-class classification, discussed in this section.

3.3. Intrusion detection systems that used both binary and multi-class classification

In this subsection, we survey the papers that used both binary and multi-class classification for classifying dataset. We classify these papers into different classes based on whether they used a single dataset or multiple datasets for evaluating their approach. First, we discuss the papers that used single dataset for evaluation.

3.3.1. IDSes that used NSL-KDD dataset for evaluation

Salama et al. [78] proposed a three step approach for detecting intrusion in computer networks. In the first step they preprocess the data by changing symbolic features into numeric values, for normalizing and labeling the data. In the second step, to reduce the dimension of features, they used DBN with two layers of Restricted Boltzmann Machine (RBM) [79]. They used back propagation algorithm for DBN. The first layer had 41 inputs(NSL-KDD has 41 features in total). It was reduced to 13 and was then taken as input for second layer. The output of 2nd layer consisted of 5 features. In the third step, they used SVM to do the classification. They compared their hybrid model with DBN as classifier and SVM without feature reduction. Accuracy of their model is shown to be better than DBN, and SVM. They calculated the accuracy of their approach. But other metrics such as precision, recall, and F1 score were not calculated.

Javaid et al. [80] introduced a deep learning technique based on AE for feature representation and feature learning. They used soft-max regression for classification. Additionally, in the preprocessing stage, they transformed categorical features into continuous features and normalized the dataset using min–max method. They performed two types of evaluations. In order to do cross validation, they used training data for both training and testing. In the second approach, they used different datasets for testing and training. Perez et al. [81] presented a study on different types of hybrid ML models for detecting intrusion in computer networks. They combined supervised and unsupervised ML techniques to design hybrid models so that known attacks can be detected by supervised learning and unknown attacks can be detected by unsupervised learning. They used either ANN or SVM in the first level (for

Table 7
Summary of the merits and demerits of papers that used multi-class classification.

Paper	Year	Method(s) used	Dataset(s) used	Type of classification used	Strengths and weaknesses
Horng et al. [58]	2011	SVM and BRICH	KDD99	Multi-class	The model used hierarchical clustering for filtering out quality data for training the SVM model. Performance of the model on the minority attack classes is not good.
Shone et al. [77]	2018	NDAE and RF	NSL-KDD and KDD99	Multi-class	Extensive experiments were done to compare with other models. The data imbalance problem in the datasets used is not addressed.
Ali et al. [64]	2018	PSO-FLN	KDD99	Multi-class	Performance of the model is better than all the other compared models. The dataset used lacks training data in some attack classes.
Li et al. [60]	2018	Bat Algorithm and RF	KDD99	Multi-class	The proposed method achieved low false positive rate. They used improved Bat Algorithm with differential evolution to identify optimal features. It has room to improve the detection rate of the low frequency attack classes (e.g., User to Root (U2R), Remote to local (R2L) attacks).
Gao et al. [71]	2019	MultiTree and Ensemble adaptive voting algorithm using some base classifiers.	NSL-KDD	Multi-class	Proposed an ensemble approach to combine some base classifiers to develop a model so that it can use the advantages of each classifier. The model's performance against data imbalance problem of NSL-KDD dataset is not satisfactory.
Zhang et al. [70]	2019	GA and DBN	NSL-KDD	Multi-class	The method has higher accuracy on the minority class types such as U2R.
Hindy et al. [75]	2020	LR, K-NN, DT, RF, SVM, and NB	MQTT-IoT	Multi-class	The authors used a novel and simulated dataset to evaluate six ML models. Some attack types have data imbalance issue in the dataset.
Moualla et al. [72]	2021	Extra Tree Classifier and ELM	UNSW-NB15	Multi-class	Overall performance of the model is good. The model has good accuracy on minority classes.

supervised learning). In the next level, after applying ANN or SVM, those entries detected as normal activities were clustered using K-means method (using unsupervised learning). In some hybrid models, they integrated either PCA or Gradual Feature Reduction (GFR) as feature selection method. Afterwards, they compared the performance of these models using the validation dataset.

Yin et al. [82] proposed a two step deep learning approach for intrusion detection. In the preprocessing step, they used one hot encoding to transform categorical data into numerical values. After that, the dataset was normalized using min–max method due to large variations in the data. In the classification step, RNN with forward propagation and backward propagation was used. Forward propagation was responsible for calculating output values and backward propagation was used to calculate the error and update the weights. Cross-entropy was used to calculate the difference between output values, produced by the forward propagation and true value. Both binary and multi-class classification were performed in this approach. Performance of this approach using KDDTest-21 was around 64%.

Lee et al. [83] proposed a constrained optimization-based extreme learning machine (C-ELM) [84] approach for network intrusion detection. Their approach added hidden neurons in an adaptively incremental way which helped in deriving output without re-computation from scratch. They developed a method to adaptively increase hidden neurons and the optimization criteria. As a result, time and effort could be saved during the construction process of an optimal C-ELM. Experimental results show that this approach is effective in building models with good attack detection rates and fast learning speed. Although this was not a new approach, it was more efficient in terms of recomputing the optimum C-ELM, according to the authors.

3.3.2. Intrusion detection systems that used UNSW-NB15 dataset for evaluation

Almogren [85], proposed a method to detect intrusion in the Edge-of-things networks. This approach has the following three steps: network data collection, feature extraction, and classification. In the feature extraction step, they transformed categorical data into numerical data. After that they used min–max normalization to scale the feature values. In the classification step, they used DBN which is stack of Restricted Boltzman Machines (RBMs) [79]. It is a two layer DBN. To find the optimal architecture of the DBN, they varied the number of hidden neurons in the layers. Finally, 64 hidden neurons in layer-1 and 60 hidden neurons in layer-2 were determined to give good performance. The author addressed the issue of imbalanced data by oversampling the dataset. No cross validation was done to measure the confidence of the model's accuracy.

3.3.3. Intrusion detection systems that used BoT-IoT dataset for evaluation

Ge et al. [86], proposed a deep learning based approach to detect intrusions in IoT. They used FNN to detect intrusions. They extract features based on the information from header fields in IP packets. The reason to use header information in IP packets was to capture generic features instead of generating attack oriented features. After extracting these features, they fed all the training data to the FNN to do the classification. They performed both binary and multi-class classification. They used BoT-IoT dataset [87], created in the cyber range lab of the UNSW Institute for Cyber Security. They compared their method with support vector classifier (SVC). Their method achieved higher accuracy than SVC.

3.3.4. Intrusion detection systems that used multiple datasets for evaluation

In this subsection, we discuss the papers that used more than one dataset for evaluation.

Sangkatsanee et al. [88] used two datasets for their experiment. The first one is the RLD09 (Reliable Lab Data 2009), provided by the King Mongkuts's University of Technology Thonburi (KMUTT), Thailand. The first dataset was generated using various established tools and contained data related to several types of attacks. The second one was the dataset KDD99 [89]. They divided their proposed method into preprocessing, classification and post-processing. In the preprocessing step, they extracted different network packets such as TCP, UDP, ICMP, etc. Packet information was partitioned based on the source and destination IP address pairs of connections established connections. To form a record, they aggregated information received over 2 s. Based on extensive experiments, they found the key features that can be used to define normal and attack traffic. They used information gain technique to select 12 features. In the next step, they used different ML models to classify training data. In the final step, the goal was to remove outliers from the classification. They used majority voting algorithm on five consecutive results for data collected for each (source, destination) pair of IP addresses. If three out of five results are detected as attack, then it was considered as an attack; otherwise it is considered normal. They created their own traffic data and preprocessed them. In the lab created dataset, distribution of data is well balanced. Nevertheless, they did not consider using any publicly available datasets for validation. Therefore, assessing the performance of this model is difficult.

Singh et al. [90], addressed the issues related to processing large network traffic data. To address the space and time complexity in intrusion detection approaches, they proposed a method based on online sequential extreme machine learning (OS-ELM) [91], where they addressed data imbalance problem and memory overhead problem. The authors used two network traffic datasets (NSL-KDD and Kyoto University Benchmark Dataset 2009) to evaluate their model. They compared their model with ANN, AdaBoost, Naïve Bayes, and ELM. Their approach performed better than other models compared. On the first dataset, their model's accuracy was around 3% higher than the closest competitor model. On the second dataset, it achieved around 2% higher accuracy than the closest competitor model.

Vinayakumar et al. [92] proposed a hybrid DNN framework for intrusion detection, called scale-hybrid-IDS-AlertNet. This can detect both host and network level intrusions. To make this framework scalable, they used Apache Spark cluster computing platform117, developed over the Apache Hadoop Yet Another Resource Negotiator (YARN). They tested their model using publicly available benchmark datasets KDDCup 99, NSL-KDD, Kyoto, UNSW-NB15, WSN-DS, and CICIDS 2017. They have done extensive evaluation to compare their model with the other models. Overall performance of their model on all of the above datasets has been shown to be superior to all the competitor models compared. Table 8 summarizes the merits and demerits of the papers using both binary and multi-class classification, discussed in this section.

3.4. Research works that evaluated the effectiveness of ML-based IDSes

In this section we discuss some papers that evaluated the effectiveness of ML-based IDSes.

Wang [93] studied the effect of several adversarial attack methods on a DNN-based IDSes. To test the vulnerabilities of DNN-based model with respect to those state-of-the-art attack algorithms, the author used NSL-KDD dataset as experimental dataset. The author introduced malicious data samples into the dataset using those methods. Four common adversarial attack techniques were used to test the vulnerabilities of this MLP model. They were Fast Gradient Sign Method (FGSM) [94], Jacobin-based Saliency Map Attack (JSMA) [95], Deepfool [96], and Carlini and Wagner (CW) [97]. Their evaluation showed that their approach achieved 94% accuracy on the original dataset. After using those adversarial attacks, accuracy has significantly reduced. The CW attack was least effective on accuracy; it reduced accuracy to 80%. FGSM and JSMA reduced the accuracy to 44% and 50% respectively. The author also discussed how an attacker can manually change training data to facilitate adversarial attacks in real world.

Sikha et al. [98] conducted a study on botnet traffic where traffic data were collected in an IoT environment using benchmark machine learning classifiers. The dataset they used to do the classification experiments was prepared by UCI [99]. The authors addressed the imbalance problem in the dataset by selecting about equal amount of data for both benign and malicious attack

Table 8
Summary of the merits and demerits of papers that used both binary and multi-class classification.

Paper	Year	Method(s) used	Dataset(s) used	Type of classification used	Strengths and weaknesses
Sangkatsanee et al. [88]	2011	DT, ANN, Bayes and Ripper Rule	RLD09, KD99	Binary and Multi-class	Used custom dataset and filtered out important features.
Salama et al. [78]	2011	DBN, SVM and DBN-SVM	NSL-KDD	Binary and Multi-class	Feature selection technique is used to reduce the dimension of the dataset. Cross validation and data imbalance problems are not addressed.
Singh et al. [90]	2015	Alpha-FST-beta and OS-ELM	NSL-KDD and Kyoto 2009	Binary and Multi-class	Proposed to solve the problem of data imbalance and space problem in the system.
Javaid et al. [80]	2016	Soft-max regression and self-taught learning	NSL-KDD	Binary and Multi-class	Used feature learning technique before applying any classification algorithm. There is data imbalance problem in the NSL-KDD dataset.
Yin et al. [82]	2017	j48, Naïve Bayes, RF, SVM, Multi layer perceptron, NB tree and RNN	NSL-KDD	Binary and Multi-class	The resulting model is better compared with some of the other models. The data imbalance problem in the NSL-KDD dataset is not addressed.
Lee et al. [83]	2017	C-ELM	NSL-KDD	Binary and Multi-class	Constructing C-ELM in a way that hidden nodes and output weights can be updated in incremental way can save time.
Vinayakumar et al. [92]	2019	DNN	KDDCup 99, NSL-KDD, Kyoto, UNSW-NB15, WSN-DS, and CICIDS 2017	Binary and Multi-class	They used Apache Spark cluster computing platform to make their framework scalable. For multi-class classification, accuracy on some attack categories has dropped on some datasets.
Ge et al. [86]	2019	FNN	BoT-IoT	Binary and Multi-class	The authors used newly published IoT dataset to evaluate their model. They compared their method with only one method (SVC).
Almogren [85]	2020	DBN, SVM, ANN	UNSW-NB15	Binary and Multi-class	Addressed the data imbalance problem in the dataset. May have overfitting or underfitting problem because cross validation was not performed.

types. Before applying machine learning classifiers, they normalized the dataset using z-score. They used LR, SVM, and RF to do the binary classification. According to their experimental results, all the classifiers reached above 99% accuracy.

Alqahtani et al. [100], analyzed widely used ML models on intrusion detection to test the effectiveness. They used KDD99 dataset to evaluate ML models. After doing initial preprocessing of the dataset, they employed ML models to do the classification. They used Bayesian Network (BN), NB, RF, DT, RT, Decision Table (DTb), and ANN as machine learning models. According to their analysis, the RF model based IDS had better performance in term of accuracy, precision, recall, and F_1 score than other classifiers. The authors have done extensive evaluation of the ML models, however, they did not address the data imbalance problem in KDD99 dataset.

4. Related work

In this section, we briefly discuss the survey papers on intrusion detection, published over the last decade, to bring out the difference between our survey paper and the existing ones. Ganapathy et al. [101] present a survey on feature selection and classification techniques used for intrusion detection. They also present two new algorithms — an attribute selection based feature selection algorithm and a rule-based multi-class support vector machine algorithm. Mitchell et al. [102] present a critical survey of 28 intrusion detection techniques for cyber–physical systems. They classify these IDSes as knowledge-based and behavior-based. They further classify each of these into two subclasses based on whether they use host-based audit data or network-based audit data for detecting intrusion. They also discuss the performance metrics used in IDSes, the characteristics of the IDSes and their merits and demerits. Butun et al. [103] present a survey of IDSes for wireless sensor networks. Milenkoski et al. [104] present a survey of the common practices used for evaluating IDSes. They categorized the IDSes based on (i) monitored platform (host based, network based or hybrid), (ii) attack detection method used (misuse based/signature based, anomaly based or hybrid) and (iii) deployment architecture (distributed or non distributed). Then, they survey the evaluation approaches and methods used by these IDSes with respect to workload, metrics used, and measurement methodology used.

Centralized IDSes do not scale well and are not capable of detecting many types of intrusions. To address this problem, Collaborative Intrusion Detection Systems (CIDSes) which consist of multiple components at the network level and host level for monitoring/analyzing and exchanging events to determine abnormal events have been proposed. Vasilomanolakis et al. [105] present a detailed framework of requirements and building blocks for CIDSes. Then, they present a critical analysis of CIDSes presented in the literature with respect to their framework.

Buczak et al. [5] present a detailed history of the evolution of ML and DM techniques, and the differences and similarities between these techniques. They also present a detailed description of various datasets used by the ML/DM approaches for cybersecurity applications. Then, they describe in detail various ML/DM methods used in cybersecurity applications such as Artificial Neural Networks, Association Rules and Fuzzy Association Rules, Bayesian Network, Clustering, Decision Trees, etc. For each method, they also present a critical survey of some of the seminal works which used that method for intrusion detection. Finally, they discuss the computational complexity of ML and DM methods and some open issues.

Liu et al. [106] observe that the ML algorithms used in IDSes and the data used for training the IDSes are vulnerable to various attacks and present a survey of the various security threats against a variety of learning algorithms such as Naïve Bayes, DT, SVM, etc. They also classify the defensive mechanisms presented against those threats. Benkhelifa et al. [107] discuss how traditional approaches for intrusion detection are not suitable for IoT and present a survey of intrusion detection techniques presented in the literature for IoT. Then, they propose an architecture for IDSes, especially suitable for IoT.

Nisioti et al. [108] present a survey of unsupervised and hybrid ML approaches for intrusion detection. They compare and analyze those approaches, identify their drawbacks and make recommendations for future research. How advanced data analytics techniques can be used for correlating attacks to identify attackers is also discussed in this paper. They also identify new classes of attacks that do not belong to any of the known classes of attacks and suggest methods for detecting them. Resende et al. [109] point out the absence of surveys of IDSes using RF-based methods for classification, feature selection, etc. and filled in this gap. Chaabouni et al. [110] present a survey of IDSes designed for Internet of Things. Khraisat et al. [111] classify and present a critical survey of some of the recent works on IDS and also discuss the commonly used datasets for evaluating the IDSes. Liang et al. [112] discuss in detail the benefits and drawbacks of using ML in designing and implementing IDSes.

Kiennert et al. [113] present a survey of intrusion detection techniques that use game theory for analyzing data and detecting intrusions. They point out the limitations of these approaches and also discuss some future directions. Al-Garadi et al. [114] present a survey of IDSes especially designed for IoT, and discuss their merits and demerits and also present some directions for future research. Wu et al. [115] point out that the existing in-vehicle network (IVN) designs for vehicles such as CAN (controller area network) did not take cybersecurity into consideration. Then, they present a survey of IDSes proposed in the literature for IVNs, highlights their drawbacks and present some directions for future research. Bridges et al. [2] present a survey of IDSes that use host based data sources such as system logs and other audit logs to detect intrusions. *Our survey focuses on IDSes based on ML only, presented in the last decade. So, this survey would serve as a supplement to the existing surveys as well as ready reference to work done on intrusion detection based on ML in the last decade.*

5. Discussion and open issues

In this paper, we reviewed the research papers on ML-based IDSes, presented in the literature over the last 10 years. First, we discussed some background related to ML approaches and also discussed some of the widely used datasets for evaluating the performance of ML-based IDSes and their drawbacks. Then, we classified the papers based on whether they used binary, multi-class, or a combination of both these two types of classification methods for labeling data. Within each of these three classes, we also classified the papers based on whether they used a single dataset or multiple datasets for evaluating the performance of their approach. We also presented a summary of the merits and demerits of the papers discussed in each class. Next, we discuss some open issues that still remain to be solved.

5.1. Some open issues and future directions for research

5.1.1. Parallelizing IDSes and distributed IDSes

Centralized IDSes are suitable only for small networks and hence are not scalable. However, ML based centralized IDSes which make use of parallel processing could help in early detection of network-based attacks including zero-day attacks. Distributed and collaborative IDSes are more scalable but may not detect all intrusions unless all events are collected and processed at a central location. Some IDSes use honeypots for detecting intrusions. However, they cannot help in detecting malwares such as logic bombs that may not react with honeypots.

5.1.2. Intrusion detection in virtualized environments

Virtualization plays an important role in efficient use of available resources in cloud computing environments and large data centers. However, such environments are vulnerable to different types of attacks. A virtual machine monitor (VMM), also called hypervisor, is capable of hosting several virtual machines (VMs) on a single host, creating a multi-tenant environment in which multiple clients can reside simultaneously on the same machine; this creates additional security vulnerabilities such as side channel attacks, snooping virtual machines, etc. Existing metrics for evaluating IDSes assume fixed set of hardware and hence are not suitable for evaluating IDSes designed for virtual environments. Datasets suitable for evaluating IDSes designed for virtual environments need to be generated and published so other researchers can use them.

5.1.3. Dealing with data imbalance problem

Khoshgoftaar et al. [116] did a comprehensive analysis of the performance of RF method and found it to be more robust compared to other methods for dealing with imbalanced datasets. More research needs to be done in finding efficient methods for dealing with data imbalance problem.

5.1.4. Labeling training data

ML models using supervised methods need to have labeled data for training; efficient methods for determining labels for training data are needed. Pattern matching may be one approach to create labels for supervised methods. Buczak et al. [5] recommend incremental learning for supervisory methods.

5.1.5. Issues related to datasets and training models used

We discussed some of the most popular and publicly available datasets used for training IDSes and their limitations in Section 2.3. For network intrusion detection, we need more robust datasets which are fully labeled and complete with original data, instead of lab generated traces. Generally, a given ML model is trained to solve a specific problem and training a ML model with appropriate dataset takes time. Moreover, applying single ML model cannot help in solving intrusion detection in cyber–physical systems due to their complex nature. So, several types of models in combination with diverse data are needed to detect intrusions in cyber–physical systems. Moreover, such models cannot handle the dynamically changing environment, because in such systems training data change continuously. To address this problem, on-the-fly and distributed training approaches need to be designed and implemented.

Another major problem with supervised ML based intrusion detection methods is the non-availability of training datasets which contain all types of attacks. This is especially true for IoT and other cyber–physical systems. So, it is important to generate datasets which include all types of attacks for different environments. Moreover, for detecting zero-day-attacks, training datasets and models need to be dynamically updated.

5.1.6. Security of machine learning algorithms

ML algorithms themselves are susceptible to attacks because the training datasets could be targets for potential attackers. Attackers could inject malicious samples to the datasets with incorrect labels so they can evade IDSes. These types attacks are possible especially against ML algorithms that dynamically update their training datasets.

5.1.7. Performance metrics

Most of the intrusion detection techniques presented in the literature focus only on detection rate, false positive and false negative rates and accuracy but do not talk about detection latency. Timely detection of intrusions will help in reducing recovery time and recovery overhead. So, detection latency is also an important metric for evaluating IDSes. Different adversaries behave differently. So, modeling the behavior of adversaries could also help in designing better IDSes and behavior based IDSes could also help in detecting zero-day attacks.

6. Conclusion

In this paper, we presented a critical survey of research work done on intrusion detection using ML techniques over the last decade. We also discussed some of the open issues that still remain to be addressed. This survey is complementary to other existing surveys on intrusion detection and will serve as a supplement to other surveys. It will also serve as ready reference to researchers working on intrusion detection using ML techniques.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] J.P. Anderson, Computer Security Technology Planning Study, Vol. 2, James P. Anderson & Co., Fort Washington, PA, 1972.
[2] R. Bridges, T. Glass-Vanderlan, M. Iannacone, M. Vincent, Q. Chen, A survey of intrusion detection systems leveraging host data, ACM Comput. Surv. 52 (6) (2020) 1–35.
[3] W. Stallings, L. Brown, Computer Security Principles and Practice (4th Edition), Pearson, 2018, ISBN-13: 978-0-13-479410-5.
[4] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall, 1999.
[5] A.L. Buczak, E. Guven, A survey of data mining and machine learning methods for cyber security intrusion detection, IEEE Commun. Surv. Tutor. 18 (2) (2016) 1153–1176, http://dx.doi.org/10.1109/COMST.2015.2494502.
[6] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice Hall, Englewood Cliffs, NJ, USA, 1988.
[7] E. Fix, J. Hodges, Discriminatory Analysis, Nonparametric Discrimination: Consistency Properties, Tech. Rep., USAF School of Aviation Medicine, Randolph Field, Texas, 1951.
[8] Y. Cheng, Mean shift, mode seeking, and clustering, IEEE Trans. Pattern Anal. Mach. Intell. 17 (8) (1995) 790–799, http://dx.doi.org/10.1109/34.400568.
[9] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: KDD'96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, ACM, 1996, pp. 226–231.
[10] D.A. Reynolds, Gaussian Mixture Models, Tech. Rep., MIT Lincoln Laboratory, 244 Wood St., Lexington, MA 02140, USA, 2009.

[11] T.M. Cover, P.E. Hart, Nearest neighbor pattern classification, IEEE Trans. Inform. Theory 13 (1) (1967) 21–27, http://dx.doi.org/10.1109/TIT.1967.1053964.
[12] J.R. Quinlan, Induction of decision trees, Mach. Learn. 1 (1986) 81–106, http://dx.doi.org/10.1007/BF00116251.
[13] J.R. Quinlan, C4.5: PRograms for Machine Learning, Morgan Kaufmann, 1993.
[14] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (5) (1988) 359–366, http://dx.doi.org/10.1016/0893-6080(89)90020-8.
[15] I.H. Witten, M.A.H. Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, San Mateo, CA, USA, 2011.
[16] S.M. Abd Elrahman, A. Abraham, A review of class imbalance problem, J. Netw. Innov. Comput. 1 (2013) (2013) 332–340.
[17] N.V. Chawla, N. Japkowicz, A. Kotcz, Special issue on learning from imbalanced data sets, ACM SIGKDD Explor. Newsl. 6 (1) (2004) 1–6.
[18] C.-E. Sarndal, J. Wretman, B. Swensson, Model Assisted Survey Sampling, Springer, 1992.
[19] KDD cup 1999 data, 2021, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, Ocotber 2007. (Accessed 21 August 2021).
[20] S.J. Stolfo, W. Fan, W. Lee, A. Prodromidis, P.K. Chan, Cost-based modeling for fraud and intrusion detection: Results from the JAM project, in: Proceedings DARPA Information Survivability Conference and Exposition, DISCEX'00, vol. 2, IEEE, 2000, pp. 130–144.
[21] M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani, A detailed analysis of the KDD CUP 99 data set, in: Proceedings of 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, IEEE, 2009, pp. 1–6.
[22] R.P. Lippmann, D.J. Fried, I. Graf, J.W. Haines, K.R. Kendall, D. McClung, D. Weber, S.E. Webster, D. Wyschogrod, R.K. Cunningham, et al., Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation, in: Proceedings DARPA Information Survivability Conference and Exposition, DISCEX'00, vol. 2, IEEE, 2000, pp. 12–26.
[23] A. Özgür, H. Erdem, A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015, PeerJ Preprints 4 (2016) e1954v1.
[24] NSL-KDD dataset, 2021, https://www.unb.ca/cic/datasets/nsl.html. (Accessed 21 August 2021).
[25] H.H. Pajouh, G. Dastghaibyfard, S. Hashemi, Two-tier network anomaly detection model: a machine learning approach, J. Intell. Inf. Syst. 48 (1) (2017) 61–74.
[26] N. Moustafa, J. Slay, UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set), in: Proceedings of 2015 Military Communications and Information Systems Conference, MilCIS, IEEE, 2015, pp. 1–6.
[27] TCPDUMP, 2021, https://www.tcpdump.org/. (Accessed 21 August 2021).
[28] Argus tool, 2021, https://openargus.org/. (Accessed 21 August 2021).
[29] Bro-IDS tool, 2021, https://zeek.org/. (Accessed 21 August 2021).
[30] N. Moustafa, J. Slay, The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set, Inf. Secur. J. Glob. Perspect. 25 (1–3) (2016) 18–31.
[31] Y. Yang, K. Zheng, C. Wu, X. Niu, Y. Yang, Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks, Appl. Sci. 9 (2) (2019) 238, http://dx.doi.org/10.3390/app9020238.
[32] UNSW-NB15 dataset, 2021, https://research.unsw.edu.au/projects/unsw-nb15-dataset. (Accessed 21 August 2021).
[33] I.T. Jolliffe, J. Cadima, Principal component analysis: a review and recent developments, Phil. Trans. R. Soc. A 374 (2065) (2016) 20150202.
[34] W.S. Noble, What is a support vector machine? Nature Biotechnol. 24 (12) (2006) 1565–1567.
[35] T. Joachims, SVM^{light}: Support vector machine, 2008, http://svmlight.joachims.org/.
[36] R.M. Elbasiony, E.A. Sallam, T.E. Eltobely, M.M. Fahmy, A hybrid network intrusion detection framework based on random forests and weighted k-means, Ain Shams Eng. J. 4 (4) (2013) 753–762.
[37] A. Abusitta, M. Bellaiche, M. Dagenais, T. Halabi, A deep learning approach for proactive multi-cloud cooperative intrusion detection system, Future Gener. Comput. Syst. 98 (2019) 308–318, http://dx.doi.org/10.1016/j.future.2019.03.043.
[38] Y.K. Ever, B. Sekeroglu, K. Dimililer, Classification analysis of intrusion detection on NSL-KDD using machine learning algorithms, in: Proceedings of International Conference on Mobile Web and Intelligent Information Systems, in: Lecture Notes in Computer Science, vol. 11673, Springer, Cham, 2019.
[39] M. Begli, F. Derakhshan, H. Karimipour, A layered intrusion detection system for critical infrastructure using machine learning, in: Proceedings of IEEE 7th International Conference on Smart Energy Grid Engineering, SEGE, IEEE, Oshawa, ON, Canada, Canada, 2019, http://dx.doi.org/10.1109/SEGE.2019.8859950.
[40] M.N. Chowdhury, K. Ferens, M. Ferens, Network intrusion detection using machine learning, in: Proceedings of International Conference on Security Management, SAM, Las Vegas, USA, 2016, pp. 1–7.
[41] R. Abdulhammed, M. Faezipour, A. Abuzneid, A. AbuMallouh, Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic, IEEE Sens. Lett. 3 (1) (2019) http://dx.doi.org/10.1109/LSENS.2018.2879990.
[42] N.V. Chawla, Data mining for imbalanced datasets: An overview, in: Data Mining and Knowledge Discovery Handbook, Springer, 2009, pp. 875–886.
[43] N. Japkowicz, et al., Learning from imbalanced data sets: a comparison of various strategies, in: AAAI Workshop on Learning from Imbalanced Data Sets, vol. 68, AAAI Press Menlo Park, CA, 2000, pp. 10–15.
[44] M. Ring, S. Wunderlich, D. Grüdl, D. Landes, A. Hotho, Creation of flow-based data sets for intrusion detection, J. Inf. Warfare 16 (4) (2017) 41–54.
[45] G.M. James, Majority Vote Classifiers: Theory and Applications, Stanford University, 1998.
[46] J. Han, M. Kamber, J. Pei, Data mining concepts and techniques third edition, Morgan Kaufmann Ser. Data Manag. Syst. 5 (4) (2011) 83–124.
[47] B. Zenko, L. Todorovski, S. Dzeroski, A comparison of stacking with meta decision trees to bagging, boosting, and stacking with other methods, in: Proceedings 2001 IEEE International Conference on Data Mining, IEEE, 2001, pp. 669–670.
[48] D. Bhamare, T. Salman, M. Samaka, A. Erbad, R. Jain, Feasibility of supervised machine learning for cloud security, in: Proceedings of International Conference on Information Science and Security, ICISS, IEEE, 2016, http://dx.doi.org/10.1109/ICISSEC.2016.7885853.
[49] S. Saad, I. Traore, A. Ghorbani, B. Sayed, D. Zhao, W. Lu, J. Felix, P. Hakimian, Detecting P2P botnets through network behavior analysis and machine learning, in: Proceedings of 2011 Ninth Annual International Conference on Privacy, Security and Trust, IEEE, 2011, pp. 174–180.
[50] M. Injadat, A. Moubayed, A.B. Nassif, A. Shami, Multi-stage optimized machine learning framework for network intrusion detection, IEEE Trans. Netw. Serv. Manag. (2020) http://dx.doi.org/10.1109/TNSM.2020.3014929, (Early Access).
[51] Z. Chen, Q. Yan, H. Han, S. Wang, L. Peng, L. Wang, B. Yang, Machine learning based mobile malware detection using highly imbalanced network traffic, Inform. Sci. 433 (2018) 346–364.
[52] R.S.B. Krishna, M. Aramudhan, Feature selection based on information theory for pattern classification, in: 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies, ICCICCT, IEEE, 2014, pp. 1233–1236.
[53] B. Bonev, Feature Selection Based on Information Theory, Universidad de Alicante, 2010.
[54] J. Li, K. Cheng, S. Wang, F. Morstatter, R.P. Trevino, J. Tang, H. Liu, Feature selection: A data perspective, ACM Comput. Surv. 50 (6) (2017) 1–45.
[55] M.A. Hall, Correlation-based feature selection for machine learning, (Ph.D. thesis), The University of Waikato, Hamilton, NewZealand, 1999.
[56] I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization, in: Proceedings of 4th International Conference on Information Systems Security and Privacy, ICISSP, vol. 1, 2018, pp. 108–116, http://dx.doi.org/10.5220/0006639801080116.
[57] G. Andresini, A. Appice, N. Di Mauro, C. Loglisci, D. Malerba, Multi-channel deep feature learning for intrusion detection, IEEE Access 8 (2020) 53346–53359.

[58] S.-J. Horng, M.-Y. Su, Y.-H. Chen, T.-W. Kao, R.-J. Chen, J.-L. Lai, C.D. Perkasa, A novel intrusion detection system based on hierarchical clustering and support vector machines, Expert Syst. Appl. 38 (1) (2011) 306–313, http://dx.doi.org/10.1016/j.eswa.2010.06.066.
[59] T. Zhang, R. Ramakrishnan, M. Livny, BIRCH: An efficient data clustering method for very large databases, in: Proceedings of 1996 ACM SIGMOD International Conference on Management of Data, SIGMOD '96, ACM, 1996, pp. 103–114, http://dx.doi.org/10.1145/233269.233324.
[60] J. Li, Z. Zhao, R. Li, H. Zhang, AI-based two-stage intrusion detection for software defined IoT networks, IEEE Internet Things J. 6 (2) (2018) 2093–2102.
[61] H. Kim, N. Feamster, Improving network management with software defined networking, IEEE Commun. Mag. 51 (2) (2013) 114–119.
[62] A.-C. Enache, V. Sgârciu, A feature selection approach implemented with the binary BAT algorithm applied for intrusion detection, in: Proceedings of 2015 38th International Conference on Telecommunications and Signal Processing, TSP, IEEE, 2015, pp. 11–15.
[63] J. Wang, J. Liao, Y. Zhou, Y. Cai, Differential evolution enhanced with multiobjective sorting-based mutation operators, IEEE Trans. Cybern. 44 (12) (2014) 2792–2805.
[64] M.H. Ali, B.A.D. Al Mohammed, A. Ismail, M.F. Zolkipli, A new intrusion detection system based on fast learning network and particle swarm optimization, IEEE Access 6 (2018) 20255–20261.
[65] S.K. Sahu, S. Sarangi, S.K. Jena, A detail analysis on intrusion detection datasets, in: Proceedings of 2014 IEEE International Advance Computing Conference, IACC, IEEE, 2014, pp. 1348–1353.
[66] V.K. Mishra, A. Sengupta, MO-PSE: Adaptive multi-objective particle swarm optimization based design space exploration in architectural synthesis for application specific processor design, Adv. Eng. Softw. 67 (2014) 111–124.
[67] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: a new learning scheme of feedforward neural networks, in: Proceedings of 2004 IEEE International Joint Conference on Neural Networks, vol. 2, IEEE, 2004, pp. 985–990.
[68] Z.W. Geem, J.H. Kim, G. Loganathan, A new heuristic optimization algorithm: Harmony search, Simulation 76 (2) (2001) 60–68, http://dx.doi.org/10.1177/003754970107600201.
[69] L. Jia, Z. Li, An ameliorated teaching-learning based optimization algorithm for nonlinear bilevel programming, in: Proceedings of 2016 12th International Conference on Computational Intelligence and Security, CIS, IEEE, 2016, pp. 52–56, http://dx.doi.org/10.1109/CIS.2016.0021.
[70] Y. Zhang, P. Li, X. Wang, Intrusion detection for IoT based on improved genetic algorithm and deep belief network, IEEE Access 7 (2019) 31711–31722, http://dx.doi.org/10.1109/ACCESS.2019.2903723.
[71] X. Gao, C. Shan, C. Hu, Z. Niu, Z. Liu, An adaptive ensemble machine learning model for intrusion detection, IEEE Access 7 (2019) 82512–82521, http://dx.doi.org/10.1109/ACCESS.2019.2923640.
[72] S. Moualla, K. Khorzom, A. Jafar, Improving the performance of machine learning-based network intrusion detection systems on the UNSW-NB15 dataset, Comput. Intell. Neurosci. 2021 (2021).
[73] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, J. Artificial Intelligence Res. 16 (2002) 321–357.
[74] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1–3) (2006) 489–501.
[75] H. Hindy, E. Bayne, M. Bures, R. Atkinson, C. Tachtatzis, X. Bellekens, Machine learning based IoT intrusion detection system: An MQTT case study (MQTT-IoT-IDS2020 dataset), in: B. Ghita, S. Shiaeles (Eds.), Selected Papers from the 12th International Networking Conference, INC 2020, in: Lecture Notes in Networks and Systems, Springer, Cham, 2021.
[76] OASIS-Standard, MQTT Version 5.0, 2019, https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html.
[77] N. Shone, T.N. Ngoc, V.D. Phai, Q. Shi, A deep learning approach to network intrusion detection, IEEE Trans. Emerg. Top. Comput. Intell. 2 (1) (2018) 41–50, http://dx.doi.org/10.1109/TETCI.2017.2772792.
[78] M.A. Salama, H.F. Eid, R.A. Ramadan, A. Darwish, A.E. Hassanien, Hybrid intelligent intrusion detection scheme, in: Soft Computing in Industrial Applications, Springer, 2011, pp. 293–303.
[79] D.E. Rumelhart, J.L. McLelland (Eds.), Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations, Paul Smolensky - Chapter 6: Information Processing in Dynamical Systems: Foundations of Harmony Theory, MIT Press, 1986, pp. 194–281.
[80] A. Javaid, Q. Niyaz, W. Sun, M. Alam, A deep learning approach for network intrusion detection system, in: Proceedings of the 9th EAI International Conference on Bio-Inspired Information and Communications Technologies (Formerly BIONETICS), 2016, pp. 21–26.
[81] D. Perez, M.A. Astor, D.P. Abreu, E. Scalise, Intrusion detection in computer networks using hybrid machine learning techniques, in: Proceedings of 2017 XLIII Latin American Computer Conference (CLEI), Cordoba, 2017, pp. 1–10, http://dx.doi.org/10.1109/CLEI.2017.8226392.
[82] C. Yin, Y. Zhu, J. Fei, X. He, A deep learning approach for intrusion detection using recurrent neural networks, IEEE Access 5 (2017) 21954–21961.
[83] C.-H. Lee, Y.-Y. Su, Y.-C. Lin, S.-J. Lee, Machine learning based network intrusion detection, in: Proceedings of 2nd IEEE International Conference on Computational Intelligence and Applications, ICCIA, IEEE, Beijing, China, 2017, http://dx.doi.org/10.1109/CIAPP.2017.8167184.
[84] G.-B. Huang, X. Ding, H. Zhou, Optimization method based extreme learning machine for classification, Neurocomputing 74 (1–3) (2010) 155–163.
[85] A.S. Almogren, Intrusion detection in Edge-of-Things computing, J. Parallel Distrib. Comput. 137 (2020) 259–265, http://dx.doi.org/10.1016/j.jpdc.2019.12.008.
[86] M. Ge, X. Fu, N. Syed, Z. Baig, G. Teo, A. Robles-Kelly, Deep learning-based intrusion detection for IoT networks, in: Proceedings of IEEE 24th Pacific Rim International Symposium on Dependable Computing, PRDC, IEEE, Kyoto, Japan, 2019, http://dx.doi.org/10.1109/PRDC47002.2019.00056.
[87] N. Koroniotis, N. Moustafa, E. Sitnikova, B. Turnbull, Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset, Future Gener. Comput. Syst. 100 (2019) 779–796.
[88] P. Sangkatsanee, N. Wattanapongsakorn, C. Charnsripinyo, Practical real-time intrusion detection using machine learning approaches, Comput. Commun. 34 (18) (2011) 2227–2235, http://dx.doi.org/10.1016/j.comcom.2011.07.001.
[89] W. Lee, S.J. Stolfo, K.W. Mok, Mining in a data-flow environment: Experience in network intrusion detection, in: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1999, pp. 114–124.
[90] R. Singh, H. Kumar, R.K. Singla, An intrusion detection system using network traffic profiling and online sequential extreme learning machine, Expert Syst. Appl. 42 (22) (2015) 8609–8624, http://dx.doi.org/10.1016/j.eswa.2015.07.015.
[91] N.Y. Liang, G.-B. Huang, P. Saratchandran, N. Sundararajan, A fast and accurate online sequential learning algorithm for feedforward networks, IEEE Trans. Neural Netw. 17 (6) (2006) 1411–1423.
[92] R. Vinayakumar, M. Alazab, K.P. Soman, P. Poornachandran, A. Al-Nemrat, Deep learning approach for intelligent intrusion detection system, IEEE Access 7 (2019) 41525–41550, http://dx.doi.org/10.1109/ACCESS.2019.2895334.
[93] Z. Wang, Deep learning-based intrusion detection with adversaries, IEEE Access 6 (2018) 38367–38384.
[94] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, 2014, arXiv preprint arXiv:1412.6572.
[95] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z.B. Celik, A. Swami, The limitations of deep learning in adversarial settings, in: 2016 IEEE European Symposium on Security and Privacy, EuroS&P, IEEE, 2016, pp. 372–387.
[96] S.-M. Moosavi-Dezfooli, A. Fawzi, P. Frossard, Deepfool: A simple and accurate method to fool deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2574–2582.
[97] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: Proceedings of 2017 IEEE Symposium on Security and Privacy, SP, IEEE, 2017, pp. 39–57.
[98] S. Bagui, X. Wang, S. Bagui, Machine learning based intrusion detection for IoT botnet, Int. J. Mach. Learn. Comput. 11 (6) (2021).
[99] UCI machine learning repository, 2021, https://archive.ics.uci.edu/ml/datasets/detection_of_IoT_botnet_attacks_N_BaIoT. (Accessed 21 August 2021).

[100] H. Alqahtani, I.H. Sarker, A. Kalim, S.M.M. Hossain, S. Ikhlaq, S. Hossain, Cyber intrusion detection using machine learning classification techniques, in: International Conference on Computing Science, Communication and Security, Springer, 2020, pp. 121–131.
[101] S. Ganapathy, K. Kulothungan, S. Muthurajkumar, M. Vijayalakshmi, P. Yogesh, A. Kannan, Intelligent feature selection and classification techniques for intrusion detection in networks: a survey, EURASIP J. Wireless Commun. Networking 2013 (1) (2013) 1–16, http://dx.doi.org/10.1186/1687-1499-2013-271.
[102] R. Mitchell, I.R. Chen, A survey of intrusion detection techniques for cyber-physical systems, ACM Comput. Surv. 46 (4) (2014) http://dx.doi.org/10.1145/2542049, Article No.: 55.
[103] I. Butun, S.D. Morgera, R. Sankar, A survey of intrusion detection systems in wireless sensor networks, IEEE Commun. Surv. Tutor. 16 (1) (2014) 266–282.
[104] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer, B.D. Payne, Evaluating computer intrusion detection systems: A survey of common practices, ACM Comput. Surv. 48 (1) (2015) 1–41, http://dx.doi.org/10.1145/2808691.
[105] E. Vasilomanolakis, S. Karuppayah, M. Mühlhäuser, M. Fischer, Taxonomy and survey of collaborative intrusion detection, ACM Comput. Surv. 47 (4) (2015) 1–33.
[106] Q. Liu, P. Li, W. Zhao, W. Cai, S. Yu, V.C.M. Leung, A survey on security threats and defensive techniques of machine learning: A datadriven view, IEEE Access 6 (2018) 12103–12117, http://dx.doi.org/10.1109/ACCESS.2018.2805680.
[107] E. Benkhelifa, T. Welsh, W. Hamouda, A critical review of practices and challenges in intrusion detection systems for IoT: Toward universal and resilient systems, IEEE Commun. Surv. Tutor. 20 (4) (2018) 3496–3509, http://dx.doi.org/10.1109/COMST.2018.2844742.
[108] A. Nisioti, A. Mylonas, P.D. Yoo, V. Katos, From intrusion detection to attacker attribution: A comprehensive survey of unsupervised methods, IEEE Commun. Surv. Tutor. 20 (4) (2018) 3369–3388.
[109] P. Resende, A. Drummond, A survey of random forest based methods for intrusion detection systems, ACM Comput. Surv. 51 (3) (2018) 1–36.
[110] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, P. Faruki, Network intrusion detection for IoT security based on learning techniques, IEEE Commun. Surv. Tutor. 21 (3) (2019) 2671–2701, http://dx.doi.org/10.1109/COMST.2019.2896380.
[111] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, Survey of intrusion detection systems: techniques, datasets and challenges, Cybersecurity 2 (2019) 20.
[112] F. Liang, W.G. Hatcher, W. Liao, W. Gao, W. Yu, Machine learning for security and the internet of things: The good, the bad, and the ugly, IEEE Access 7 (2019) 158126–158147, http://dx.doi.org/10.1109/ACCESS.2019.2948912.
[113] C. Kiennert, Z. Ismail, H. Debar, H. Leneutre, A survey on game-theoretic approaches for intrusion detection and response optimization, ACM Comput. Surv. 51 (5) (2019) 1–31.
[114] M.A. Al-Garadi, A. Mohamed, A.K. Al-Ali, X. Du, I. Ali, M. Guizani, A survey of machine and deep learning methods for internet of things (IoT) security, IEEE Commun. Surv. Tutor. 22 (3) (2020) 1646–1685, http://dx.doi.org/10.1109/COMST.2020.2988293.
[115] W. Wu, R. Li, G. Xie, J. An, Y. Bai, J. Zhou, K. Li, A survey of intrusion detection for in-vehicle networks, IEEE Trans. Intell. Transp. Syst. 21 (3) (2020) 919–933.
[116] T.M. Khoshgoftaar, M. Golawala, J.V. Hulse, An empirical study of learning from imbalanced data using random forest, in: Proceedings of 19th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2007, IEEE, 2007, pp. 310–317, http://dx.doi.org/10.1109/ICTAI.2007.46.