Karbala International Journal of Modern Science

Volume 9 | Issue 1 Article 5

Using Ensemble Techniques Based on Machine and Deep Learning for Solving Intrusion Detection Problems: A Survey

Hadeel Qasem Gheni
Department of Software, Information Technology College, University of Babylon, Babylon, Iraq,,
wsci.hadeel.qasem@uobabylon.edu.iq

Wathiq Laftah Al-Yaseen
Karbala Technical Institute, Al-Furat Al-Awsat Technical University, 56001, Karbala, Iraq

Recommended Citation
Gheni, Hadeel Qasem and Al-Yaseen, Wathiq Laftah (2023) "Using Ensemble Techniques Based on Machine and Deep Learning for Solving Intrusion Detection Problems: A Survey," *Karbala International Journal of Modern Science*: Vol. 9 : Iss. 1 , Article 5.
Available at: https://doi.org/10.33640/2405-609X.3277

Using Ensemble Techniques Based on Machine and Deep Learning for Solving Intrusion Detection Problems: A Survey

Abstract

Obviously, the increasing threats to network security, which led to devastating network attacks, have taken a heavy toll on enterprises as a simple firewall cannot prevent complex and changing attacks. Therefore, companies should use intrusion detection systems in combination with other security devices to protect against corporate network security issues. In fact, intrusion detection is a system whose primary function is to protect network security by monitoring traffic, collecting and analyzing information, and then issuing an alert in cases where the output of the analysis represents a threat to network security. Intrusion Detection Systems (IDS) can stop unauthorized activity on a network or operating system, react automatically, stop the intrusion's source in time, record it, and alert the network administrator to ensure maximum system security. The process of detecting attacks using a single algorithm has not proven its worth. Therefore, several algorithms were used together by using ensemble learning. To elaborate, ensemble learning is a well-known predictive technique that involves training multiple algorithms to treat the same problem, after which the results are combined to produce a single, potent prediction that can provide performance better than that of a single algorithm. The primary goal of this study is to present an overview of the main ensemble techniques that are used to enhance the effectiveness of the intrusion detection system, as well as the research using these methods as published by Elsevier and Springer from 2018 until the time being. The results prove that the two easiest methods within ensemble learning to implement are majority voting and weighted averaging, which provide good results in terms of accuracy. In cases where the base models have a significant variance, the bagging method would be more beneficial, while the boosting method would be used in cases where the basic models are biased, and in order to lower bias by learning different algorithms, the stacking ensemble methods are used.

Keywords

Ensemble Learning Techniques; Intrusion Detection System; Intrusion Detection Dataset; Machine Learning; Deep Learning

REVIEW ARTICLE

Using Ensemble Techniques Based on Machine and Deep Learning for Solving Intrusion Detection Problems: A Survey

Hadeel Q. Gheni [a,*], Wathiq L. Al-Yaseen [b]

[a] Department of Software, Information Technology College, University of Babylon, Babylon, Iraq
[b] Karbala Technical Institute, Al-Furat Al-Awsat Technical University, 56001, Karbala, Iraq

Abstract

Obviously, the increasing threats to network security, which led to devastating network attacks, have taken a heavy toll on enterprises as a simple firewall cannot prevent complex and changing attacks. Therefore, companies should use intrusion detection systems in combination with other security devices to protect against corporate network security issues. In fact, intrusion detection is a system whose primary function is to protect network security by monitoring traffic, collecting and analyzing information, and then issuing an alert in cases where the output of the analysis represents a threat to network security. Intrusion Detection Systems (IDS) can stop unauthorized activity on a network or operating system, react automatically, stop the intrusion's source in time, record it, and alert the network administrator to ensure maximum system security. The process of detecting attacks using a single algorithm has not proven its worth. Therefore, several algorithms were used together by using ensemble learning. To elaborate, ensemble learning is a well-known predictive technique that involves training multiple algorithms to treat the same problem, after which the results are combined to produce a single, potent prediction that can provide performance better than that of a single algorithm. The primary goal of this study is to present an overview of the main ensemble techniques that are used to enhance the effectiveness of the intrusion detection system, as well as the research using these methods as published by Elsevier and Springer from 2018 until the time being. The results prove that the two easiest methods within ensemble learning to implement are majority voting and weighted averaging, which provide good results in terms of accuracy. In cases where the base models have a significant variance, the bagging method would be more beneficial, while the boosting method would be used in cases where the basic models are biased, and in order to lower bias by learning different algorithms, the stacking ensemble methods are used.

Keywords: Ensemble learning techniques, Intrusion detection system, Intrusion detection dataset, Machine learning, Deep learning

1. Introduction

Actually, information and communication technology (ICT) systems have had an active role in the majority of institutions and businesses, as well as other areas on which human activity relies. On the other hand, cybercrimes against ICT are widespread in cyberspace and have existed since the invention of computers [1]. Cybercrimes tend to adapt as ICT systems continue to develop, taking advantage of system flaws to carry out data thefts or completely destroy the infrastructure of the network [2]. Serious security issues have been highlighted by the rapid increase in data being communicated over a range of devices and communication protocols, such as viewing sensitive data without authorization, defacing a web server, copying a database containing credit card numbers, and many more [3]. Intrusions can be defined as any attempts made to access the network illegally and gain unauthorized

Received 28 August 2022; revised 16 October 2022; accepted 24 October 2022.
Available online 13 January 2023

* Corresponding author.
E-mail addresses: wsci.hadeel.qasem@uobabylon.edu.iq (H.Q. Gheni), wathiq@atu.edu.iq (W.L. Al-Yaseen).

https://doi.org/10.33640/2405-609X.3277

data, causing a threat to network security. Therefore, the importance of developing an advanced intrusion detection system (IDS) has increased. For detection purposes, an IDS is a security management system and a widely used method for identifying system-targeting internal intrusions as well as external ones. IDS works by gathering and examining data from networks and computers to see if any odd behaviors or suspicious activity exist, as well as anomalies that suggest potential intrusions [4]. IDS uses a variety of instruments and processes to keep an eye on network traffic and computer systems while also evaluating activities when looking for potential system intrusions [3]. The lack of existing IDSs to reveal unknown attacks leads researchers to concentrate on developing IDSs using machine learning techniques [5].

In fact, intrusion detection systems, which are beneficial for both individual computers and huge networks, can be divided into three types, namely: host intrusion detection systems (HIDS), network intrusion detection systems (NIDS), and hybrid systems [6]. The first, which is HIDS, is installed on the computer as a software application that aims to track and examine computer system activity, whereby each host is analyzed individually [7]. As for NIDS, it keeps track of the network's packet flow as it observes, evaluates, and classifies traffic based on tried-and-true methodologies and procedures to distinguish between normal and suspected traffic [8]. The third type is hybrid IDS, which combines NIDS and HIDS with high flexibility, resulting in a mechanism with stronger security [7].

Intrusion detection is performed through two techniques: anomaly-based intrusion and signature-based intrusion [9]. Signature-based IDS uses a precise definition (signature) of the attack stored in the internal database and compares incoming traffic and signatures stored [10]. This means that these systems can detect known attacks very accurately [11]. To identify active intrusion attempts, anomaly-based IDS maintains the normal state of system behavior and monitors the occurrence of any changes that will cause an alert to be generated [12]. Given the fact that it is based on common good behavior and spots any cases of abnormalities within it, anomaly-based IDS can identify unknown, zero-day attacks [13].

Ensemble learning has been used in many types of research and has been shown to be effective in the above challenges. It keeps the IDS from being out of date and makes it very good at finding new attacks at the lowest cost.

The current paper presents a study of recent research that deals with the methods of ensemble learning applied to the datasets that contain traces of attacks observed by intrusion detection systems, and it is organized in the following way: The second section includes a detailed explanation of the essential datasets used for intrusion detection. The third and fourth sections encompass a simplified description of machine learning and deep learning. The fifth section includes a detailed description of the ensemble learning method and its techniques. Finally, the sixth section states the conclusions of this review.

2. Intrusion detection datasets

The dataset is a unique compilation of information obtained from many distributed intrusion detection systems that work in concert to identify significant incidents of network security [14]. A remarkable role in intrusion detection is played by datasets [15], which are used to assess the model's suitability for accurately detecting attacks. The performance of NIDS is ultimately influenced by the quality of the dataset [16]. There are approximately 35 well-known cyber datasets, but the most frequently used databases in recent works are KDDCup1999, NSL_KDD, UNSW_NB15, and CICIDS 2017 [15], as explained below.

2.1. KDDCup99 dataset

Knowledge Discovery in Databases (KDD) was developed by the Defence Advanced Research Projects Agency (DARPA) in 1999, and even though KDD99 was created over 21 years ago, academic research still frequently uses it [17]. The training dataset for KDDCup99 involves about 4,900,000 singular connection vectors, each having 41 features. These features have been classified as either attack or normal, with only one class attribute: feature number 42 [18]. There are 21 classes in the class attribute that fall within four categories of network attacks: Denial of Service attack (DoS), Probe attack, User to Root attack (U2R), and Remote to Local attack (R2L) [19]. Table 1 illustrates the

Table 1. Different types and classifications for attacks in KDDCup99 dataset [20].

Category of Attack	Attack Name
Normal	Normal
DoS	Back, teardrop, Neptune, land, pod, smurf
U2R	Buffer_overflow, perl, load_module, rootkit
R2L	ftp_write, imap, multihop, guess_passwd, warezclient, phf, spy, warezmaster
Probe	Ipsweep, portsweep, nmap, satan

types of attacks found in the KDDCup99 dataset and their classifications, whereas Table 2 shows the size of the samples of each attack type in the KDDCup99 training and testing datasets.

A number of combined reasons created difficulties and reduced the efficiency of KDDCup99 in detecting intrusion. These involve the lack of contemporary attack patterns, advances in networking applications, and speed that have changed the nature of normal traffic. Because the KDDCup99 test set had some kinds of attacks that weren't in the training set [21], this led to the creation of a new set to get around these problems.

2.2. NSL_KDD dataset

As a result of some statistical flaws that impair the assessment of anomaly detection, which negatively impacts the effectiveness of the security analysis, the KDDCUP99 dataset has been developed into the Network Security Laboratory-Knowledge Discovery and Data Mining (NSL_KDD) dataset [23]. The NSL_KDD dataset is an upgraded version of the KDDCup99 [24]. The NSL_KDD has some characteristics that outperform KDDCup99, whereby the recurring records are omitted in the sets of training and testing, which will prevent classification systems from biasing toward these records [25]. The training and testing sets have a good number of records, so the tests can be run on the whole set without picking a small number at random [26].

The NSL_KDD is a generic dataset on network incidents with labeled intrusion events and has fascinating characteristics for the distribution of events and the interdependence among features. This dataset is much better suited to being used as a standard in intrusion detection research because it has a large number of both features and instances [27]. NSL_KDD has the same five classes (Dos, U2R, Probe, R2L, and Normal) as found in KDDCup99 and consists of 41 features, one labeled class, and one difficulty label for every traffic record [28]. The training set of the NSL_KDD involves 125,973 patterns, and the training is carried out using KDD Train data, which has 22 types of attacks. As for the testing set, it involves 22,544 patterns and testing is carried out using KDD Test data, which comprises 17 new attack types [29]. Table 3 demonstrates the primary classes and the number of patterns in each class for the NSL_KDD dataset.

2.3. UNSW_NB15 dataset

Since the prior datasets do not contain any modern attacks and the distribution of benchmark training and testing datasets differ in terms of the categories of data, this will lead the classifier to err and become less accurate. Over time, espionage and stealth attacks become more like everyday activities [30]. Therefore, in 2015, a variety of normal network traffic and recent attack events were combined by creating an artificial environment at the University of New South Wales Cyber Security Lab to form a new dataset called UNSW_NB15 [31]. The UNSW_NB15 dataset has nine categories of modernistic kinds of attack and 49 features that are made up of these different categories, and it also includes realistic actions of normal traffic [32]. The nine types of network attacks in UNSW_NB15 are DoS, Backdoor, Reconnaissance, Shellcode, Fuzzers, Worms, Generic, Analysis, and Exploits [33]. Table 4 shows the different classes of attacks and their distribution in training and testing sets in the UNSW_NB15 dataset.

2.4. CICIDS.2017 dataset

In 2017, the CICIDS.2017 dataset was developed after improving the ISCX 2012 dataset by the

Table 3. Primary classes and amount of patterns in each class for NSL_KDD [29].

Training Dataset			Testing Dataset	
Classes	No. Patterns	Per.%	No. Patterns	Per.%
Normal	67,343	53.458%	9711	**43.076%**
DOS	45,927	36.458%	7458	**33.082%**
R2L	995	0.79%	2754	**12.216%**
Probe	11,656	9.253%	2421	**10.739%**
U2R	52	0.041%	200	**0.887%**
Total	125,973	100%	22,544	100%

Table 2. The size of each Attack's samples in KDDCup99 dataset [22].

Dataset	Normal	DoS	Probe	R2L	U2R	Total
WholeKDD (Original KDD)	972,780	3,883,370	41,102	1126	52	4,898,430
10% KDD (Original KDD)	97,278	391,458	4107	1126	52	494,021
KDD corrected (Original KDD)	60,593	229,853	4166	16,347	70	311,029
KDD99Train+	87,832	54,572	2130	999	52	145,585
KDD99Test+	47,913	23,568	2678	3058	70	77,287
Train Set (For Model Selection)	8784	5458	213	100	6	14,561
Validation Set (For Model Selection)	8784	5458	213	100	6	14,561

Table 4. Attack classes in UNSW_NB15 dataset [34].

Classes	Training Dataset	Per.%	Testing Dataset	Per.%
DoS	12,264	6.994%	4089	4.966%
Backdoor	1746	0.996%	583	0.708%
Analysis	2000	1.141%	677	0.822%
Fuzzers	18,184	10.371%	6062	7.363%
Generic	40,000	22.813%	18,871	22.921%
Exploits	33,393	19.045%	11,132	13.521%
Reconnaissance	10,491	5.983%	3496	4.246%
Shell Code	1133	0.646%	378	0.459%
Worms	130	0.074%	44	0.053%
Normal	56,000	31.938%	37,000	44.940%
Total	175,341	100%	82,332	100%

College of Computer Science at New Brunswick University, as it was created using a real-time traffic generalization [35]. In CICIDS.2017, 83 features were included with 15 labels of class, one label for normal and 14 for attacks, and it has 3,119,345 samples [36]. The real reason for choosing the CICIDS.2017 dataset in the most recent research experience is that it accurately represents the current real-world network traffic [37]. CICIDS.2017 identifies a new set of attacks based on characteristics of actual network traffic, including DoS, Distributed Dos, XSS, brute force, SQL Injection, Web, Botnet, Portscan, and infiltration attacks [38]. The main problem in CICIDS.2017 is the huge amount of data that requires massive processing and extended processing time, which reduces the classification algorithm's effectiveness. It also contains missing and redundant data, which could bias the inputs used to train the prediction model [39]. Table 5 demonstrates the different classes of attacks in the CICIDS.2017 dataset after removing the missing values.

Table 5. CICIDS.2017 attack types and instances frequency [36].

Class Labels	No. Instances	Per.%
BENIGN	2,359,087	83.34406%
DDoS	41,835	1.477987%
DoS slowloris	5796	0.204767%
DoS Hulk	231,072	8.163531%
DoS GoldenEye	10,293	0.363641%
DoS Slowhttptest	5499	0.194274%
Infiltration	36	0.001272%
FTP-Patator	7938	0.280441%
SSH-Patator	5897	0.208335%
PortScan	158,930	5.61483%
Heartbleed	11	0.000389%
Bot	1966	0.069457%
Web Attack – Brute Force	1507	0.053241%
Web Attack – Sql Injection	21	0.000742%
Web Attack – XSS	652	0.023034%
Total	2,830,540	100%

3. Machine learning

Machine learning (ML) is a form of artificial intelligence (AI) approach that can extract useful information automatically from enormous datasets. Machine learning-based intrusion detection systems (IDSs) can achieve satisfactory detection levels when the data available for the training process is sufficient [40].

To distinguish a normal event from an abnormal one, ML algorithms are used to instantly and precisely identify the key differences between them and provide great generalizability, thereby enabling the detection of unknown attacks [5]. Despite the outstanding performance on small datasets, the machine learning algorithm has had trouble scaling to massive datasets [41].

4. Deep learning

Deep Learning (DL), a subfield of machine learning, could indeed provide impressive results when compared to more conventional machine learning methods. It outperforms conventional methods by creating meaningful information representations from huge datasets. Thus, it is appropriate for threat detection and classification in networks [42]. Deep learning enables computers to automatically extract, evaluate, and understand useful information from the original data [43]. The basis of deep learning is the computation of layered features, wherein the features on the top level are derived from those on the low level [44]. It can speed up the detection of any irregularities and provide a deeper examination of network data [45]. Fig. 1 demonstrates the deep neural network block diagram.

5. Ensemble learning methods

Detecting attacks using a single algorithm has not proven its worth. This is because the attacks are renewed and varied over time and the accuracy obtained from a single algorithm is low. Therefore, several algorithms were used together by using ensemble learning. The most advanced response to many machine learning problems is ensemble learning methods, which involve training numerous models and integrating their predictions to increase the predictive performance of an individual model [46]. One of the key aims that academics pursue when constructing an ensemble is to allow for as much individuality in the ensemble members as possible, especially in terms of misclassification [47]. Fig. 2 shows the ensemble learning diagram.

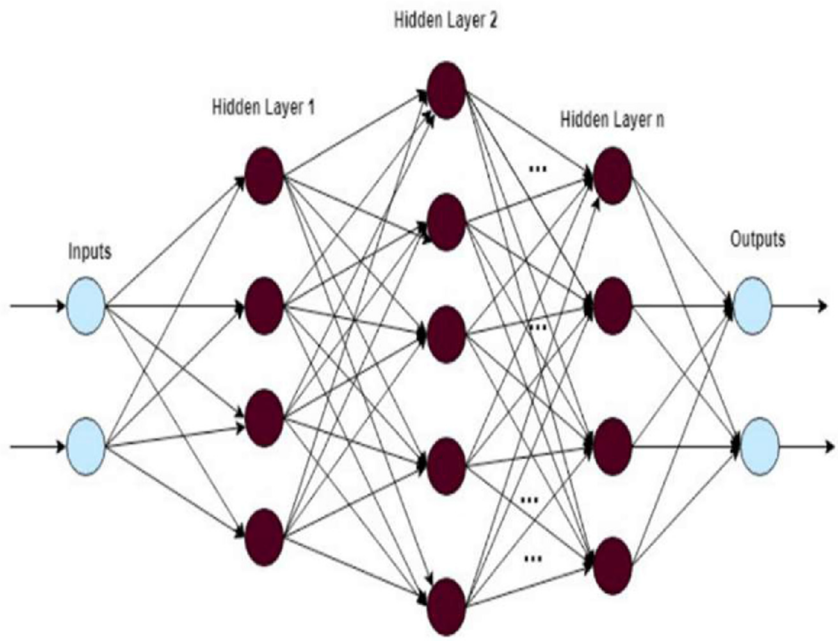

Fig. 1. Deep neural network block diagram [16].

Instead of deploying a single fit of the approach, the main idea behind ensemble methods is to build a linear combination of various model fitting techniques [49]. Fig. 3 shows the primary methods for ensemble learning.

The primary distinctions between the learning strategies mentioned in Fig. 4 are how they are trained and used. This implies that they have different applications in the real world. In the next section, these techniques will be elaborated on.]

5.1. Bagging method

Among the most advanced and simple approaches to achieving better efficiency based on the ensemble principle is bagging. It is an abbreviation for the Bootstrap Aggregating method. In the bagging method, one classifier is used and trained on various subsets of the same dataset [50]. By using bootstrapped copies of the training data, a range of outputs is generated. In particular, a large number of data subsets are picked at random with the replacement from the entire training dataset to build several learners in parallel [51]. The ultimate result of the ensemble classifier is derived by combining the results of the various basis classifiers. Typically, the results are merged using the majority voting method [52]. Fig. 4 explains the mechanism of the bagging ensemble technique [53].

There are two common types of bagging techniques:

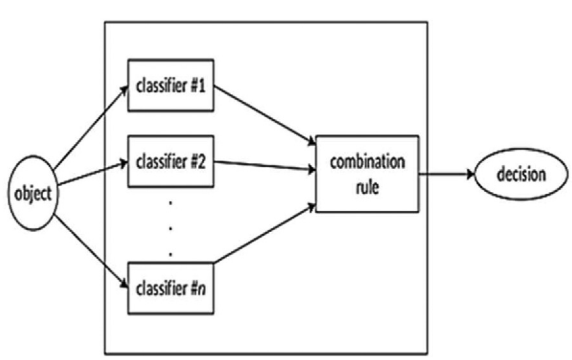

Fig. 2. Ensemble learning diagram [48].

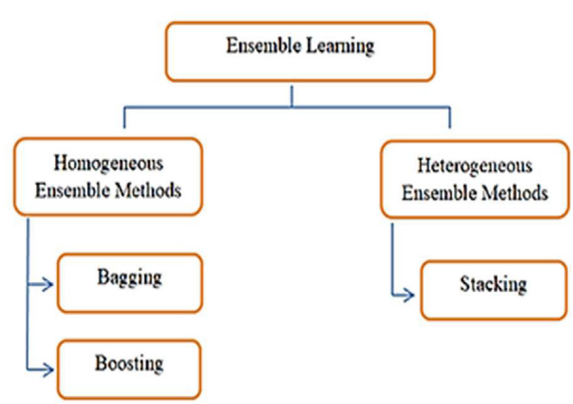

Fig. 3. Main ensemble learning techniques.

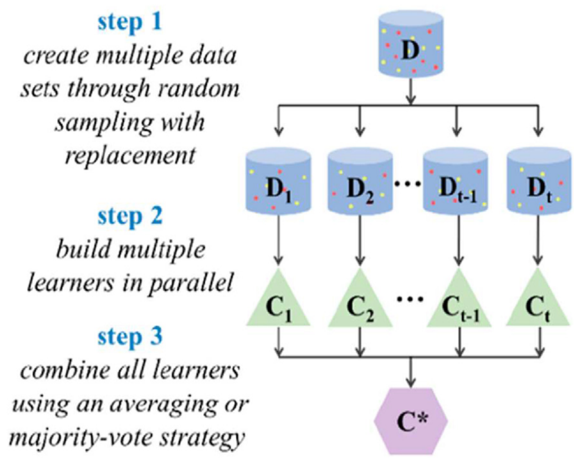

Fig. 4. Bagging ensemble technique [53].

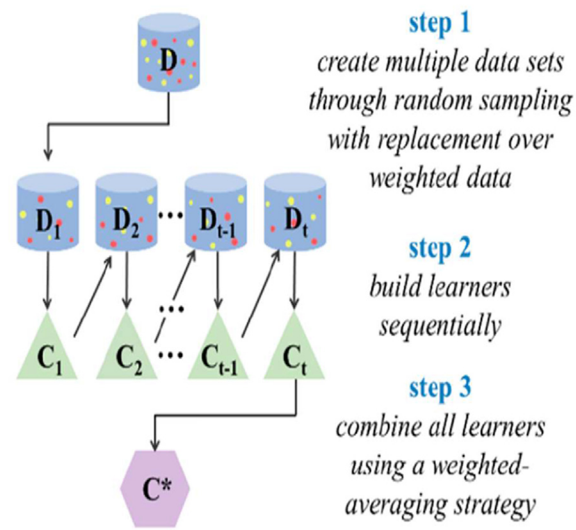

Fig. 5. Boosting ensemble technique [53].

5.1.1. Random Forest Method

Random Forest is a form of bagging that deploys decision trees to create a forest of decision trees. For node-to-node separation, these trees are formed by selecting attributes at random [54].

5.1.2. Random Forest Method

Another form of the bagging approach is the Wagging method, which is based on training instance extractions utilizing a non-uniform probability. The bagging method takes out instances from the current training dataset that have the same odds, while the wagging method takes out instances based on how their probabilities are weighted randomly [52].

5.2. Boosting method

The second technique of ensemble learning that produces a strong classifier by combining several poorly performing classifiers is called "boosting". In this case, the predictors are sequentially learned so that the first one learns from the entire collection of data, whereas the subsequent ones are learned from training sets, depending on how well the preceding one performed [55]. The boosting method learns several classifiers iteratively using various training data distributions constructed via random sampling with replacement over weighted data. By giving previously misclassified examples more weight, the modifications are directed at the training data to point further classifiers toward more challenging situations [56]. Fig. 5 explains the mechanism of the boosting ensemble technique.

There are three common types of boosting techniques:

5.2.1. Adaptive Boosting (AdaBoost)

AdaBoost, or adaptive boosting, is a generic method for producing a robust performer classifier from a series of weak performers that is effective even when the classifiers are drawn from a continuum of possible classifiers [57]. AdaBoost enables the designer to keep adding weak learners whose accuracy is only limited until a desirable low training error has been attained. It is considered "adaptive" in that it does not demand previous information on whether or not these assumptions are valid. Instead, it evaluates the validity of a base hypothesis at each iteration and adjusts its parameters as necessary [56].:

5.2.2. Gradient Boosting

One of the ensemble learning algorithms that is built from a mixture of weak-performing learners that can progressively learn from the prior misclassifications to construct a more powerful learning model is the Gradient Boosting method [58]. This method is a common supervised machine-learning technique for regression and classification tasks [59].

5.2.3. Extreme Gradient Boosting or XGBoost

One of the ensemble machine learning algorithms that have gained increasing popularity due to its scalability and performance is the XGBoost algorithm, with distributed or memory-constrained settings, which has been shown to be quicker than other well-known algorithms on a single computer when scaling to billions of samples [60]. Via continuous model iteration, the XG Boost classifier creates a model characterized by being highly accurate and a minimal false positive rate, by combining a large number of tree models that have lower classification accuracy. In terms of computation speed, generalization effectiveness, and scalability, XGBoost significantly outperforms the conventional Gradient Boosting Decision Tree (GBDT) algorithm, which combines two strategies to accelerate the algorithm [61]. Gradient Boosting Machine (GBM), Stochastic Gradient Boosting (SGB), and Regularized Gradient Boosting (RGB) are three primary gradient boosting methods that XGBoost supports [62].

5.3. Stacking method

Unlike the previous two methods, which are homogeneous ensemble methods, stacking is a distinct technique for ensemble methodology that combines numerous different classifiers, i.e., heterogeneous classifiers [63]. A stacked ensemble is implemented over two stages: base classifiers and meta classifiers. The fundamental idea behind stacking is to forecast samples using a meta-classifier that has been learned from base classifiers [64]. Stacking creates new training data to categorize unclassified data using several classifiers as base classifiers [63]. Fig. 6 explains the mechanism of the stacking ensemble technique.

In addition to the advanced methods mentioned earlier, ensemble learning also includes simple methods that are mentioned below, such as the Majority Voting method and the Weighted Averaging method.

5.4. Majority voting method

Majority voting, also called Max Voting, is a method that adheres to democratic principles, and the class determines the outcome with the most votes [66]. It is considered to be simple to implement so that the weight of all agents is equivalent. The result is determined by the ensemble agent's votes so that it can be regarded as the ensemble's final result when more than half of the ensemble agents concur [67].

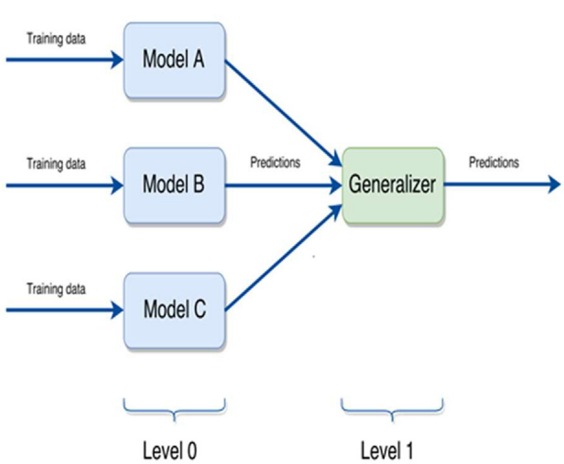

Fig. 6. Stacking ensemble technique [65].

5.5. Weighted averaging method

An ensemble approach called Weighted Averaging was applied to enhance the performance of the classification mode by aggregating the single classifier's classification results and choosing the group that received the most votes, depending on the weights assigned to the single classifiers [68]. The voting method with various class weights is utilized to get the best detection outcomes [69].

Table 6 is a summary of a lot of the research published by Elsevier and Springer between 2018 and 2022 that used ensemble methods. This is because the methods we just talked about are so important.

6. Discussion of results

The correct diagnosis of malicious behavior is critical. Even though there have been many positive changes in the area of intrusion detection to find attacks that affect the network in both multi- and binary-specification cases, the issue of performance is still being worked on because no algorithm has been found yet that gives good performance in intrusion detection.

Therefore, ensemble-based deep learning techniques have been used in many types of research to enhance the functionality of IDSs, where the main reason behind using the ensemble principle instead of a single algorithm lies in improving the performance, as it learns several algorithms and therefore gives much better performance and prediction results than a single algorithm.

The content of the datasets are alerts of network attacks that are observed by the intrusion detection system. All the techniques applied to these datasets are for classifying the data, and therefore the techniques differ from each other depending on the strength of the technology. Because of this, researchers used different methods to improve and develop the data classification process.

As will be shown in Table 6, the two easiest methods within ensemble learning to comprehend and implement are majority voting and weighted averaging, which provide good results in terms of accuracy. Bagging ensemble methods, which learn the models in parallel mode, are used when aiming to lower the variance and avoid overfitting, thus resulting in better accuracy. Therefore, if the base

Table 6. Papers used ensemble techniques for the period from 2018 to 2022.

Ref	Year	Ensemble Tech.	Dataset	Algorithms	Work Summary
[70]	2018	Majority Voting	NSL-KDD	Support Vector Machine (SVM), Modified Naive Bayes (MNB), and Linear Programming Boost (LPBoost)	To identify the ideal attribute subset, a feature selection technique called Chi-Square is used, and by using the ensemble of Majority Voting, the accuracy of detecting normal, DoS and R2L is 99%, while the accuracy of Probe and U2R is 98% and 100% respectively.
[71]	2018	Weighted Voting	KDD Cup99	Core Vector Machine (CVM)	Four models of CVM (that is, CVMs for each type of attack in KDDCup99) are used. Each model filters out the necessary features before converting them to the required coordinates of x and y. After that, the distance from (x,y) points to the center of the core vector circle is computed and compared with the radius of that circle. Finally, the result of the entire system is acquired by a weighted voting method that predicts if the arriving connection is one of the attack types or not. Accuracy for DoS is 0.9905%, for Probe 0.9450%, for U2R 0.9371% and for R2L 0.7641%.
[72]	2019	Bagging	KDD Cup99	Marginal distance minimization (MDM)-based selective ensemble (MDMbSE) method	To determine different illicit uses and abuses of computer systems in actual time, the Adaptive network intrusion detection (ANID) method based on the selective ensemble of kernel extreme learning machines (KELMs) with random attributes (named ANID-SEoKELM) is used. Achieved 99.53% accuracy.
[73]	2019	Weighted Averaging	Real-World Datasets	Linear Regression	DELR, or double-level ensemble linear regression, has superior robustness and the ability to reduce the danger of information loss. At the first level, the goal is to lose as little information as possible. At the second level, the goal is to improve the ability to generalize.
[74]	2019	Bagging And Majority Voting Methods	Real-World Dataset	Set of Bayesian Network	It suggested a detection method for XSS attacks based on an ensemble learning strategy learned together with attack intelligence and knowledge domain. The accuracy achieved was 98.54%.
[75]	2019	AdaBoost	KDD Cup99	ACO + MCC-based GFR + Ensemble of decision trees	Using a well-known ensemble method to integrate several decision trees, building an adequate training set by using ant colony optimization, and selecting the proper subset of starting features by utilizing an effective feature selection strategy, results in a model with a high degree of accuracy, detection quality for imbalanced classes, stability, and consistency. The accuracy is 99.92%.

[76]	2019	SVM ensemble	NSL-KDD	Different SVM classifiers	The K-distinct SVM classifiers are trained in the lower layer to create an ID model for a binary class, and the output from these classifiers is then fed into the SVM in the upper layer, where the accuracy is 99.41%
[77]	2019	Voting	ISCX2012, NSL-KDD, Kyoto 2006+	SVM, Instance-Based Learning Algorithms (IBK), and Multi-Layer Perceptron (MLP)	A hybrid strategy combining information gain (IG) and principal component analysis (PCA) is suggested to keep the best attribute subset and eliminate unnecessary features. Then, the ensemble model, which is based on SVM, IBK, and MLP, is utilized and obtains 99.01% accuracy on ISCX 2012, 98.24% on NSL_KDD, and 98.95% on Kyoto 2006+.
[63]	2020	Stacking	NSL-KDD	Gradient Descent (GD), Random Forest (RF)	The stacking ensemble achieves greater accuracy, recall, and detection rates where the DR for DoS is 99.77%, Probe is 38.83%, R2L is 88.98%, and U2R 76.12%. The recall for Dos is 81.85%, Probe is 96.11%, R2L is 97.75%, U2R is 89.47%, and the accuracy for the ensemble method is 91.06%.
[6]	2020	Stacking	UNSW-NB15	RF, SVM, Naive Bayes (NB)	Obtaining 95% accuracy by applying the stacking method and logistic regression as a meta-classifier for integrating methods.
[78]	2020	Bagging	KDDcup99, NSL-KDD	Decision Tree	A new ensemble approach called the ET classifier is utilized to create separate classifiers, train these classifiers, and combine the results to produce a decisive judgment. The accuracy is 99.97% on KDDCup99 and 99.32% on NSL_KDD.
[79]	2020	Majority Voting	NSL-KDD, AWID, CICIDS 2017	C4.5, RF, Forest by Penalizing Attributes (Forest PA)	To choose the best subset depending on the correlation among features, the Correlation-based Feature Selection-Bat algorithm (CFS-BA) is proposed where the ensemble classifier obtains accuracy equal to 99.81% on NSL_KDD, 99.52% on the Aegean Wifi Intrusion Dataset (AWID), and 99.89% on CICIDS 2017.

(*continued on next page*)

Table 6. (continued)

Ref	Year	Ensemble Tech.	Dataset	Algorithms	Work Summary
[80]	2020	Hybrid Ensemble	NSL-KDD	IBk (K-Nearest Neighbor (K-NN)), Random Tree (RT), REPTree, j48graft, RF	A filter-based attribute evaluation method and an ensemble classifier both gave 99.72% accuracy for the binary class and 99.68% accuracy for the multi-class class.
[81]	2020	Majority Voting	KDD Cup99	Random Subspace Algorithm	An accuracy of 98.9% was achieved via a new ID method based on a discriminant classifier ensemble. This model uses the Random Subspace Algorithm to build an ensemble of discriminant classifiers. The goal of the ensemble approach is to compare various independent classifiers and add them together to get a single estimated classifier. This method weighs the separate perspectives before combining them to make a judgment.
[82]	2021	Max Voting	CICIDS 2017	Boosted tree, Bagged tree, Random Under-Sampling (RUS) boosted tree, Subspace Discriminant	The Max voting approach and ensemble learning techniques have been developed for network-based cloud IDS. The accuracy after implementation is 97.24%.
[83]	2021	Weighted Voting	NSL-KDD	Decision Tree	The Effective Online Bagging classification performance is superior to the C4.5 and Under Over Bagging (UOB) methods and comparable to the AdaBoost approach.
[84]	2021	Boosting (XGB)	KDD Cup99	Decision Tree	Created an incremental IDS classifier that utilizes the Drift Detection concept in the advanced data, where, every time the performance deteriorates, the proposed technique adjusts the classifier, thus improving the accuracy and recall. The accuracy is 0.99078.
[85]	2022	Stacking	CICIDS 2017	Decision Tree, NB, Logistic Regression (LR)	The accuracy of the proposed model is 88.96% in the multi-class and 88.92% in the binary-class.
[86]	2022	Voting	UNSW-NB15	Multiple SVM	The SVM ensemble and Chaos Game Optimization (CGO) method are integrated to improve the ID process by managing the basic complexity of the big data related to various forms of heterogeneity of the security data, thus obtaining 96.29% accuracy.

models have a significant variance, the bagging method would be more beneficial. Boosting ensemble methods, which learn the models in sequential mode, are used when dealing with bias problems, which are reduced to obtain a better performance. Thus, the Boosting method would be more beneficial in the event that the basic models are biased. As for the stacking ensemble methods, they are used to learn distinct learning algorithms to lower the bias via learning different learners' strengths and filling in their inadequacies.

7. Conclusion

The assumption behind intrusion detection is that the intruder's behavior tends to be different from that of an authorized user in quantifiable ways. However, no clear and precise distinction could be assumed between an intruder's attack and an authorized user's regular use of resources due to the fact that there is some overlap between natural and malicious behavior. Therefore, any behavioral change will be viewed as an intrusion and result in false alarms at high rates. Many IDSs have a high rate of false alarms, which means that attacks that pose a severe threat are often ignored. This makes it hard to figure out what the new attacks are.

Deep learning is one of the innovative methods recently widely used by IDS to improve their effectiveness in protecting the network of computers. Deep learning methods are important and valuable because, unlike traditional approaches, their architecture includes multiple levels of data processing for entering data, turning it into information, and finally producing the results.

Because of the importance of this topic, a research paper was presented that includes a review of the essential methods of ensemble learning for both machine and deep learning algorithms, including homogeneous methods such as bagging and boosting techniques and heterogeneous methods such as stacking techniques. Also, the most important research papers that used these methods and were published in international journals affiliated with Springer and Elsevier from 2018 to the present have been reviewed so that they are easy to find, and a summary of their work is made.

References

[1] R. Vinayakumar, K.P. Soman, P. Poornachandran, A comparative analysis of deep learning approaches for network intrusion detection systems (N-IDSs): deep learning for N-IDSs, Int. J. Digital Crime Foren. (IJDCF) 11 (2019) 65—89. http://doi:10.4018/IJDCF.2019070104.

[2] M. Kavitha, K. Elamukhil, R. Ajeeth, R. Ashwin, V. Balasubramaniam, Distributed ensemble based deep learning architecture for intrusion detection against cyber attacks, in: Journal of Physics: Conference Series, IOP Publishing. 2021 012080, https://doi.org/10.1088/1742-6596/1916/1/012080.

[3] A. Aldweesh, A. Derhab, A.Z. Emam, Deep learning approaches for anomaly-based intrusion detection systems: a survey, taxonomy, and open issues, Knowl. Base. Syst. 189 (2020) 105124, https://doi.org/10.1016/j.knosys.2019.105124.

[4] X.K. Li, W. Chen, Q. Zhang, L. Wu, Building Auto-Encoder Intrusion Detection System based on random forest feature selection, Comput. Secur. 95 (2020) 101851, https://doi.org/10.1016/j.cose.2020.101851.

[5] H. Liu, B. Lang, Machine learning and deep learning methods for intrusion detection systems: a survey, Appl. Sci. 9 (2019) 4396, https://doi.org/10.3390/app9204396.

[6] M.S. Abirami, U. Yash, S. Singh, Building an ensemble learning based algorithm for improving intrusion detection system, in: Artificial Intelligence and Evolutionary Computations in Engineering Systems, Springer. 2020, pp. 635—649, https://doi.org/10.1007/978-981-15-0199-9_55.

[7] S.N. Mighan, M. Kahani, A novel scalable intrusion detection system based on deep learning, Int. J. Inf. Secur. 20 (2021) 387—403, https://doi.org/10.1007/s10207-020-00508-5.

[8] A. Boukhalfa, A. Abdellaoui, N. Hmina, H. Chaoui, LSTM deep learning method for network intrusion detection system, Int. J. Electr. Comput. Eng 10 (2020) 3315—3322, https://doi.org/10.11591/ijece.v10i3.pp3315-3322.

[9] A. Rashid, M.J. Siddique, S.M. Ahmed, Machine and deep learning based comparative analysis using hybrid approaches for intrusion detection system, in: 2020 3rd International Conference on Advancements in Computational Sciences (ICACS), IEEE. 2020, pp. 1—9, https://doi.org/10.1109/ICACS47775.2020.9055946.

[10] F. Erlacher, F. Dressler, On high-speed flow-based intrusion detection using snort-compatible signatures, in: IEEE Transactions on Dependable and Secure Computing, 2022, pp. 495—506, https://doi.org/10.1109/TDSC.2020.2973992.

[11] S. Elhag, A. Fernández, A. Altalhi, S. Alshomrani, F. Herrera, A multi-objective evolutionary fuzzy system to obtain a broad and accurate set of solutions in intrusion detection systems, Soft Comput. 23 (2019) 1321—1336, https://doi.org/10.1007/s00500-017-2856-4.

[12] S. Jose, D. Malathi, B. Reddy, D. Jayaseeli, A survey on anomaly based host intrusion detection system, in: Journal of Physics: Conference Series, Institute of Physics Publishing. 2018 012049, https://doi.org/10.1088/1742-6596/1000/1/012049.

[13] A.A. Shah, M.K. Ehsan, K. Ishaq, Z. Ali, M.S. Farooq, An efficient hybrid classifier model for anomaly intrusion detection system, IJCSNS 18 (2018) 127—135. http://paper.ijcsns.org/07_book/201811/20181117.pdf.

[14] M. Husák, M. Žádník, V. Bartoš, P. Sokol, Dataset of intrusion detection alerts from a sharing platform, Data Brief 33 (2020) 106530—106541, https://doi.org/10.1016/j.dib.2020.106530.

[15] M.A. Ferrag, L. Maglaras, S. Moschoyiannis, H. Janicke, Deep learning for cyber security intrusion detection: approaches, datasets, and comparative study, J. Inf. Secur. Appl. 50 (2020) 102419, https://doi.org/10.1016/j.jisa.2019.102419.

[16] S. Choudhary, N. Kesswani, Analysis of KDD-cup'99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT, in: Procedia Computer Science, Elsevier B.V.. 2020, pp. 1561—1573, https://doi.org/10.1016/j.procs.2020.03.367.

[17] J. v Hansen, P.B. Lowry, R.D. Meservy, D.M. McDonald, Genetic programming for prevention of cyberterrorism through dynamic and evolving intrusion detection, Decis. Support Syst 43 (2007) 1362—1374, https://doi.org/10.1016/j.dss.2006.04.004.

[18] M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani, A detailed analysis of the KDD CUP 99 data set, in: IEEE Symposium on Computational Intelligence for Security and Defense

Applications, CISDA. 2009, pp. 1–6, https://doi.org/10.1109/CISDA.2009.5356528.
[19] L.M. Ibrahim, D.T. Basheer, M.S. Mahmod, A comparison study for intrusion database (Kdd99, Nsl-Kdd) based on self organization map (SOM) artificial neural network, J. Eng. Sci. Technol. 8 (2013) 107–119. https://jestec.taylors.edu.my/V8Issue1.htm.
[20] A. Kumar Shrivas, S. Jabalpur, An ensemble model for classification of attacks with feature selection based on KDD99 and NSL-KDD data set, Int. J. Comput. Appl. 99 (2014) 8–13, https://doi.org/10.5120/17447-5392.
[21] M. Al, M. Hasan, M. Nasser, M. al Mehedihasan, B. Pal, On the KDD'99 dataset: support vector machine based intrusion detection system (IDS) with different kernels, Int J. Electron Commun. Comput. Eng. 4 (2013) 1164–1170. https://www.researchgate.net/publication/267736960.
[22] 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. A Glob. Perspect. 25 (2016) 18–31, https://doi.org/10.1080/19393555.2015.1125974.
[23] S. Revathi, A. Malathi, A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection, Int. J. Eng. Res. Technol. 2 (2013) 1848–1853, https://doi.org/10.17577/IJERTV2IS120804.
[24] L. Dhanabal, S.P. Shantharajah, A study on NSL-KDD dataset for intrusion detection system based on classification algorithms, Int. J. Adv. Res. Comp. Commun. Eng. 4 (2015) 446–452, https://doi.org/10.17148/IJARCCE.2015.4696.
[25] N. Moustafa, J. Slay, The Significant Features of the UNSW-NB15 and the KDD99 Data Sets for Network Intrusion Detection Systems, Institute of Electrical and Electronics Engineers (IEEE). 2017, pp. 25–31, https://doi.org/10.1109/badgers.2015.014.
[26] G. Meena, R.R. Choudhary, A review paper on IDS classification using KDD 99 and NSL KDD dataset in WEKA, in: 2017 International Conference on Computer, Communications and Electronics, COMPTELIX, Institute of Electrical and Electronics Engineers Inc.. 2017, pp. 553–558, https://doi.org/10.1109/COMPTELIX.2017.8004032.
[27] D.H. Deshmukh, T. Ghorpade, P. Padiya, Improving classification using preprocessing and machine learning algorithms on NSL-KDD dataset, in: 2015 International Conference on Communication, Information & Computing Technology (ICCICT), IEEE. 2015, pp. 1–6, https://doi.org/10.1109/ICCICT.2015.7045674.
[28] K. Wu, Z. Chen, W. Li, A novel intrusion detection model for a massive network using convolutional neural networks, IEEE Access 6 (2018) 50850–50859, https://doi.org/10.1109/ACCESS.2018.2868993.
[29] B. Ingre, A. Yadav, Performance analysis of NSL-KDD dataset using ANN, in: International Conference on Signal Processing and Communication Engineering Systems - Proceedings of SPACES 2015, in Association with IEEE, Institute of Electrical and Electronics Engineers Inc.. 2015, pp. 92–96, https://doi.org/10.1109/SPACES.2015.7058223.
[30] D. Jing, H.-B. Chen, SVM based network intrusion detection for the UNSW-NB15 dataset, in: 2019 IEEE 13th International Conference on ASIC (ASICON), IEEE. 2019, pp. 1–4, https://doi.org/10.1109/ASICON47005.2019.8983598.
[31] N. Moustafa, J. Slay, UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set), in: 2015 Military Communications and Information Systems Conference, MilCIS 2015 - Proceedings, Institute of Electrical and Electronics Engineers Inc.. 2015, pp. 1–6, https://doi.org/10.1109/MilCIS.2015.7348942.
[32] L. Zhiqiang, G. Mohi-Ud-Din, L. Bing, L. Jianchao, Z. Ye, L. Zhijun, Modeling network intrusion detection system using feed-forward neural network using unsw-nb15 dataset, in: 2019 IEEE 7th International Conference on Smart Energy Grid Engineering (SEGE), IEEE. 2019, pp. 299–303, https://doi.org/10.1109/SEGE.2019.8859773.
[33] S.M. Kasongo, Y. Sun, Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset, J. Big Data 7 (2020) 1–20, https://doi.org/10.1186/s40537-020-00379-6.
[34] A. Divekar, M. Parekh, V. Savla, R. Mishra, M. Shirole, Benchmarking datasets for anomaly-based network intrusion detection: KDD CUP 99 alternatives, in: 2018 IEEE 3rd International Conference on Computing, Communication and Security, ICCCS). 2018, pp. 1–8, https://doi.org/10.1109/CCCS.2018.8586840.
[35] A. Yulianto, P. Sukarno, N.A. Suwastika, Improving AdaBoost-based intrusion detection system (IDS) performance on CIC IDS 2017 dataset, in: Journal of Physics: Conference Series, Institute of Physics Publishing. 2019, p. 12018, https://doi.org/10.1088/1742-6596/1192/1/012018.
[36] R. Panigrahi, S. Borah, A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems, Int. J. Eng. Technol. 7 (2018) 479–482. https://www.researchgate.net/publication/329045441.
[37] D. Kurniabudi, D. Stiawan, M.Y. bin bin Idris, A.M. Bamhdi, R. Budiarto, CICIDS-2017 dataset feature analysis with information gain for anomaly detection, IEEE Access 8 (2020) 132911–132921, https://doi.org/10.1109/ACCESS.2020.3009843.
[38] A. Thakkar, R. Lohiya, A review of the advancement in intrusion detection datasets, in: Procedia Computer Science, Elsevier B.V.. 2020, pp. 636–645, https://doi.org/10.1016/j.procs.2020.03.330.
[39] Z. Pelletier, M. Abualkibash, Evaluating the CIC IDS-2017 dataset using machine learning methods and creating multiple predictive models in the statistical computing language R, Int. Res. J. Adv. Eng. Sci. 5 (2020) 187–191. https://irjaes.com/volume-5-issue-2/.
[40] I.H. Sarker, Y.B. Abushark, F. Alsolami, A.I. Khan, IntruDTree: a machine learning based cyber security intrusion detection model, Symmetry 12 (2020) 754, https://doi.org/10.3390/SYM12050754.
[41] S. Ma, M. Belkin, Diving into the shallows: a computational perspective on large-scale shallow learning, Adv. Neural Inf. Process Syst. 30 (2017) 1–30, https://doi.org/10.48550/arXiv.1703.10622.
[42] Y.N. Kunang, S. Nurmaini, D. Stiawan, B.Y. Suprapto, Attack classification of an intrusion detection system using deep learning and hyperparameter optimization, J. Inf. Secur. Appl. 58 (2021) 102804, https://doi.org/10.1016/j.jisa.2021.102804.
[43] N.K. Chauhan, K. Singh, A review on conventional machine learning vs deep learning, in: 2018 International Conference on Computing, Power and Communication Technologies (GUCON), IEEE. 2018, pp. 347–352, https://doi.org/10.1109/GUCON.2018.8675097.
[44] N.T. Van, T.N. Thinh, An anomaly-based network intrusion detection system using deep learning, in: 2017 International Conference on System Science and Engineering (ICSSE), IEEE. 2017, pp. 210–214, https://doi.org/10.1109/ICSSE.2017.8030867.
[45] T. Nguyen, V. Dinh, N. Shone, T. Nguyen Ngoc, V. Dinh Phai, Q. Shi, A deep learning approach to network intrusion detection, IEEE Trans. Emerg. Topics Comput. Intell. 2 (2018) 41–50, https://doi.org/10.1109/TETCI.2017.2772792.
[46] X. Dong, Z. Yu, W. Cao, Y. Shi, Q. Ma, A survey on ensemble learning, Front. Comput. Sci. 14 (2020) 241–258, https://doi.org/10.1007/s11704-019-8208-z.
[47] H.M. Gomes, J.P. Barddal, A.F. Enembreck, A. Bifet, A survey on ensemble learning for data stream classification, ACM Comput. Surv. 50 (2017) 1–36, https://doi.org/10.1145/3054925.
[48] B. Krawczyk, L.L. Minku, J. Gama, J. Stefanowski, M. Woźniak, Ensemble learning for data stream analysis: a survey, Inf. Fusion 37 (2017) 132–156, https://doi.org/10.1016/j.inffus.2017.02.004.
[49] L. Li, W. Yaonan, Z. Yexin, A handwritten digit recognizer using ensemble method, in: 2015 Chinese Automation

Congress (CAC), IEEE. 2015, pp. 469—473, https://doi.org/10.1109/CAC.2015.7382546.
[50] A. Khraisat, A. Alazab, A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges, Cybersecurity 4 (2021) 1—27, https://doi.org/10.1186/s42400-021-00077-7.
[51] AA, M.B.I.R. Aburomman, A survey of intrusion detection systems based on ensemble and hybrid classifiers, Comput. Secur. 65 (2017) 135—152, https://doi.org/10.1016/j.cose.2016.11.004.
[52] G. Kumar, K. Thakur, M.R. Ayyagari, MLEsIDSs: machine learning-based ensembles for intrusion detection systems—a review, J. Supercomput. 76 (2020) 8938—8971, https://doi.org/10.1007/s11227-020-03196-z.
[53] X. Yang, Y. Wang, R. Byrne, G. Schneider, S. Yang, Concepts of artificial intelligence for computer-assisted drug Discovery, Chem. Rev. 119 (2019) 10520—10594, https://doi.org/10.1021/acs.chemrev.8b00728.
[54] N. Farnaaz, M.A. Jabbar, Random forest modeling for network intrusion detection system, in: Procedia Computer Science, Elsevier B.V.. 2016, pp. 213—217, https://doi.org/10.1016/j.procs.2016.06.047.
[55] I. Syarif, E. Zaluska, A. Prugel-Bennett, G. Wills, Application of bagging, boosting and stacking to intrusion detection, in: International Workshop on Machine Learning and Data Mining in Pattern Recognition, Springer, Berlin, Heidelberg. 2012, pp. 593—602, https://doi.org/10.1007/978-3-642-31537-4_46.
[56] T. Phuoc Tran, L. Cao, D. Tran, C. Duc Nguyen, Novel intrusion detection using probabilistic neural network and adaptive boosting, Int. J. Comput. Sci. Inf. Secur. 6 (2009) 83—91, https://doi.org/10.48550/arXiv.0911.0485.
[57] X. Li, L. Wang, E. Sung, AdaBoost with SVM-based component classifiers, Eng. Appl. Artif. Intell. 21 (2008) 785—795, https://doi.org/10.1016/j.engappai.2007.07.001.
[58] D. Upadhyay, J. Manero, M. Zaman, S. Sampalli, Gradient boosting feature selection with machine learning classifiers for intrusion detection on power grids, IEEE Trans. Network Service Manag. 18 (2021) 1104—1116, https://doi.org/10.1109/TNSM.2020.3032618.
[59] O. Faker, E. Dogdu, Intrusion detection using big data and deep learning techniques, in: ACMSE 2019 - Proceedings of the 2019 ACM Southeast Conference, Association for Computing Machinery, Inc.. 2019, pp. 86—93, https://doi.org/10.1145/3299815.3314439.
[60] A. Gouveia, M. Correia, Network intrusion detection with XGBoost, in: Recent Advances in Security, Privacy, and Trust for Internet of Things (IoT) and Cyber-Physical Systems (CPS), Chapman and Hall/CRC. 2020, pp. 137—166, https://doi.org/10.1201/9780429270567-6.
[61] Z. Chen, F. Jiang, Y. Cheng, X. Gu, W. Liu, J. Peng, XGBoost classifier for DDoS attack detection and analysis in SDN-based cloud, in: Proceedings - 2018 IEEE International Conference on Big Data and Smart Computing (BigComp), Institute of Electrical and Electronics Engineers Inc.. 2018, pp. 251—256, https://doi.org/10.1109/BigComp.2018.00044.
[62] P. Verma, S. Anwar, S. Khan, S.B. Mane, Network intrusion detection using clustering and gradient boosting, in: 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), IEEE. 2018, pp. 1—7, https://doi.org/10.1109/ICCCNT.2018.8494186.
[63] H. Rajadurai, U.D. Gandhi, A stacked ensemble learning model for intrusion detection in wireless network, Neural. Comput. Appl. 34 (2020) 15387—15395, https://doi.org/10.1007/s00521-020-04986-5.
[64] S. Rajagopal, P.P. Kundapur, K.S. Hareesha, A Stacking Ensemble for Network Intrusion Detection Using Heterogeneous Datasets, Security and Communication Networks, 2020, pp. 1—9, https://doi.org/10.1155/2020/4586875, 2020.
[65] F. Divina, A. Gilson, F. Goméz-Vela, M.G. Torres, J.F. Torres, Stacking ensemble learning for short-term electricity consumption forecasting, Energies 11 (2018) 949, https://doi.org/10.3390/en11040949.
[66] R. Khurram Shahzad, N. Lavesson, Comparative analysis of voting schemes for ensemble-based malware detection, J. Wireless Mobile Network. Ubiquit. Comput. Depend. Appl. 4 (2013) 98—117.
[67] L. Yu, W. Yue, S. Wang, K.K. Lai, Support vector machine based multiagent ensemble learning for credit risk evaluation, Expert Syst. Appl. 37 (2010) 1351—1360, https://doi.org/10.1016/j.eswa.2009.06.083.
[68] V.C. Osamor, A.F. Okezie, Enhancing the weighted voting ensemble algorithm for tuberculosis predictive diagnosis, Sci. Rep. 11 (2021) 1—11, https://doi.org/10.1038/s41598-021-94347-6.
[69] 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, https://doi.org/10.1109/ACCESS.2019.2923640.
[70] I.S. Thaseen, C.A. Kumar, A. Ahmad, Integrated intrusion detection model using chi-square feature selection and ensemble of classifiers, Arabian J. Sci. Eng 44 (2019) 3357—3368, https://doi.org/10.1007/s13369-018-3507-5.
[71] T.H. Divyasree, K.K. Sherly, A network intrusion detection system based on ensemble CVM using efficient feature selection approach, in: Procedia Computer Science, Elsevier B.V.. 2018, pp. 442—449, https://doi.org/10.1016/j.procs.2018.10.416.
[72] J. Liu, J. He, W. Zhang, T. Ma, Z. Tang, J.P. Niyoyita, W. Gui, ANID-SEoKELM, Adaptive network intrusion detection based on selective ensemble of kernel ELMs with random features, Knowl. Base. Syst. 177 (2019) 104—116, https://doi.org/10.1016/j.knosys.
[73] J. Zhang, Z. Li, K. Nai, Y. Gu, A. Sallam, DELR: a double-level ensemble learning method for unsupervised anomaly detection, Knowl. Base. Syst. 181 (2019) 104783, https://doi.org/10.1016/j.knosys.
[74] Y. Zhou, P. Wang, An ensemble learning approach for XSS attack detection with domain knowledge and threat intelligence, Comput. Secur. 82 (2019) 261—269, https://doi.org/10.1016/j.cose.2018.12.016.
[75] S.M. Mousavi, V. Majidnezhad, A. Naghipour, A new intelligent intrusion detector based on ensemble of decision trees, J. Ambient. Intell. Hum. Comput. 13 (2022) 3347—3359, https://doi.org/10.1007/s12652-019-01596-5.
[76] J. Gu, L. Wang, H. Wang, S. Wang, A novel approach to intrusion detection using SVM ensemble with feature augmentation, Comput. Secur. 86 (2019) 53—62, https://doi.org/10.1016/j.cose.2019.05.022.
[77] F. Salo, A.B. Nassif, A. Essex, Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection, Comput. Network 148 (2019) 164—175, https://doi.org/10.1016/j.comnet.2018.11.010.
[78] BS, C.S.R. Bhati, Ensemble based approach for intrusion detection using extra tree classifier, in: Intelligent Computing in Engineering, Advances in Intelligent Systems and Computing, Springer. 2020, pp. 213—220, https://doi.org/10.1007/978-981-15-2780-7_25.
[79] Y. Zhou, G. Cheng, S. Jiang, M. Dai, Building an efficient intrusion detection system based on feature selection and ensemble classifier, Comput. Network 174 (2019) 107247, https://doi.org/10.1016/j.comnet.2020.107247.
[80] M. Dua Kunal, Attribute selection and ensemble classifier based novel approach to intrusion detection system, in: Procedia Computer Science, Elsevier B.V.. 2020, pp. 2191—2199, https://doi.org/10.1016/j.procs.2020.03.271.
[81] B.S. Bhati, C.S. Rai, B. Balamurugan, F. Al-Turjman, An intrusion detection scheme based on the ensemble of

discriminant classifiers, Comput. Electr. Eng. 86 (2020) 106742, https://doi.org/10.1016/j.compeleceng.2020.106742.
[82] P. Singh, V. Ranga, Attack and intrusion detection in cloud computing using an ensemble learning approach, Int. J. Inf. Technol. 13 (2021) 565−571, https://doi.org/10.1007/s41870-020-00583-w.
[83] H. Du, Y. Zhang, K. Gang, L. Zhang, Y.C. Chen, Online ensemble learning algorithm for imbalanced data stream, Appl. Soft Comput. 107 (2021) 107378, https://doi.org/10.1016/j.asoc.2021.107378.
[84] D. Mulimani, S.G. Totad, P. Patil, S. v Seeri, Adaptive ensemble learning with concept drift detection for intrusion detection, in: Data Engineering and Intelligent Computing, Advances in Intelligent Systems and Computing, Springer. 2021, pp. 331−339, https://doi.org/10.1007/978-981-16-0171-2_31.
[85] A. Abbas, M.A. Khan, S. Latif, M. Ajaz, A.A. Shah, J. Ahmad, A new ensemble-based intrusion detection system for internet of things, Arabian J. Sci. Eng. 47 (2022) 1805−1819, https://doi.org/10.1007/s13369-021-06086-5.
[86] A. Ponmalara, V. Dhanakotib, An intrusion detection approach using ensemble Support Vector Machine based Chaos Game Optimization algorithm in big data platform, Appl. Soft Comput. 116 (2022) 108295, https://doi.org/10.1016/j.asoc.2021.108295.