Soft Computing (2021) 25:9731–9763
https://doi.org/10.1007/s00500-021-05893-0

FOUNDATIONS

Machine learning and deep learning methods for intrusion detection systems: recent developments and challenges

Geeta Kocher[1] · Gulshan Kumar[2]

Accepted: 17 May 2021 / Published online: 24 June 2021

Abstract

Deep learning (DL) is gaining significant prevalence in every field of study due to its domination in training large data sets. However, several applications are utilizing machine learning (ML) methods from the past several years and reported good performance. However, their limitations in terms of data complexity give rise to DL methods. Intrusion detection is one of the prominent areas in which researchers are extending DL methods. Even though several excellent surveys cover the growing body of research on this subject, the literature lacks a detailed comparison of ML methods such as ANN, SVM, fuzzy approach, swarm intelligence and evolutionary computation methods in intrusion detection, particularly on recent research. In this context, the present paper deals with the systematic review of ML methods and DL methods in intrusion detection. In addition to reviewing ML and DL methods, this paper also focuses on benchmark datasets, performance evaluation measures and various applications of DL methods for intrusion detection. The present paper summarizes the recent work, compares their experimental results for detecting network intrusions. Furthermore, current research challenges are identified for helping fellow researchers in the era of DL-based intrusion detection.

Keywords Intrusion detection system · Deep learning · Deep belief network · Recurrent neural network · Network intrusion detection system

1 Introduction

With the growth of the digital world, Internet has become an integral part of our lives. The dependence on Internet is growing day by day with the development of smart cities, autonomous cars, health monitoring via smartwatches and mobile banking etc. (Ziegler 2019; Taddeo et al. 2019; Serrano 2019). Although these technologies bring in many benefits to the users and society in general, they also pose several risks. Hackers can exploit the vulnerabilities resulting in theft and sabotage, affecting the lives of people globally. Figure 1 illustrates the most frequent targeted cyber warfare attacks between 2009 and 2019 (geopolitical-attacks 2019).

Cyberattacks can be costly for businesses next to financial loss; it also leads to loss of reputation (Ghose et al. 2019). Therefore network security has become an important topic. The conventional methods like firewalls, encryption and anti-virus software packages adopted by organizations play a significant role in securing network infrastructure. Still, these methods provide the first level of defence and cannot completely protect the networks and systems from progressive attacks and malware (Srinivas et al. 2019; Kandan et al. 2019). As a result, some intruders still manage to penetrate, resulting in a breach.

Organizations use intrusion detection systems (IDSs), which Denning proposed in 1987, as an additional security technique for securing their networks (Pradhan et al. 2020). The research efforts of Denning have given directions to construct detection models effectively and accurately. In literature, IDS methods are mainly classified as Knowledge-based, Statistical and ML methods (Kumar et al. 2010) as discussed in Sect. 2.2. Artificial intelligence (AI) and ML methods determine the models from the training dataset (Arrieta et al. 2020).

These ML methods have shown excellence to achieve high detection accuracy. Still, there are some limitations of ML methods like handling raw, unlabeled or high dimensional

✉ Gulshan Kumar
gulshanahuja@gmail.com

[1] Maharaja Ranjit Singh Punjab Technical University, Bathinda, Punjab, India

[2] Shaheed Bhagat Singh State Technical Campus, Ferozepur, Punjab, India

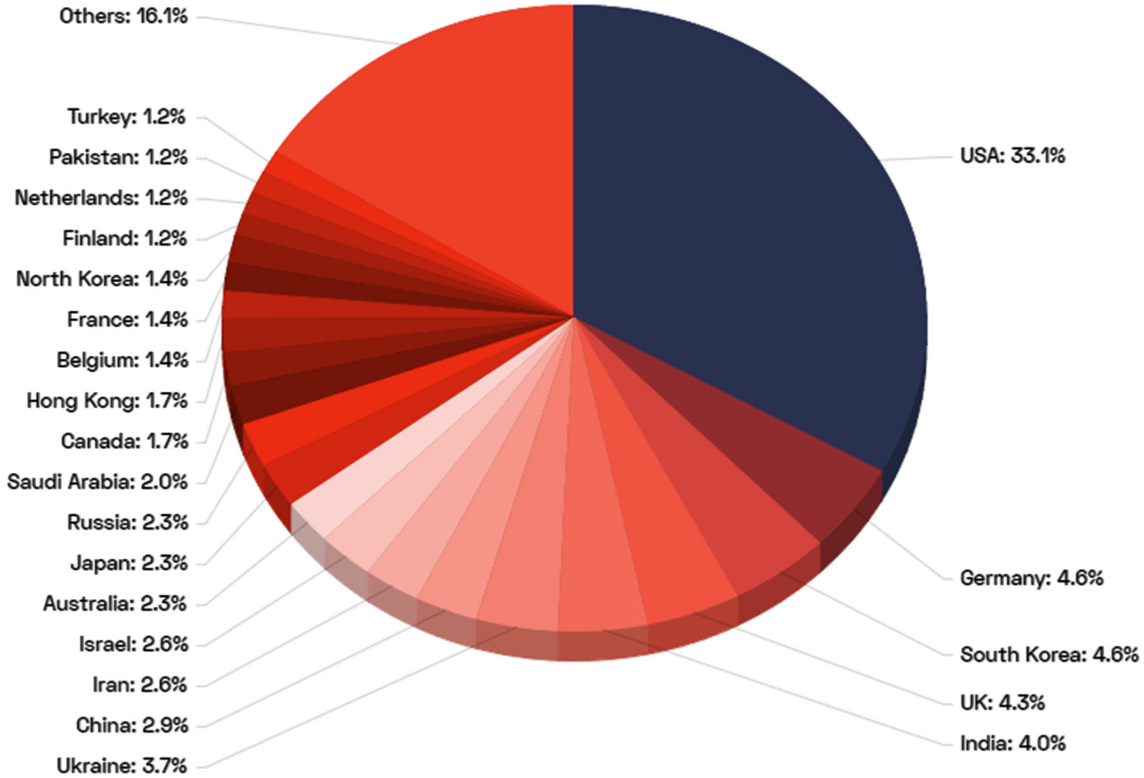


Fig. 1 Most frequent targeted cyber warfare attacks between 2009 and 2019 (geopolitical-attacks 2019)

data (Nguyen and Reddi 2019), degrades accuracy in case of a large dataset, manual feature extraction, requires expensive data labelling, time-consuming and tedious task, unable to detect multi-classification attacks (Alzaylaee et al. 2020; Meng et al. 2020). To combat these limitations, deep learning (DL)-based methods emerged in 2006. Fortunately, DL methods, known for their abilities to handle labelled or unlabelled data or solve complex problems with the help of the high powered GPU (Nguyen and Reddi 2019).

To simplify the use of ML and DL methods in intrusion detection, it is necessary to understand IDS, standard benchmark datasets, ML methods, their challenges and the reasons behind the evolution of DL methods (Nguyen and Reddi 2019; Chaabouni et al. 2019). The summarized review of ML/DL methods helps the researchers explore their advantages and disadvantages in IDS.

This paper has a dual objective. The first objective is to present a survey of recent contributions to ML and DL methods. The second objective is to explore the reasons behind the evolution of DL methods for intrusion detection.

The review paper is organized into different sections. Section 2 discusses IDSs and their taxonomy. Section 3 describes various benchmark datasets and performance evaluation measures of IDS. ML methods used for intrusion detection are discussed in Sect. 5. Section 6 introduces DL-based intrusion detection. The crucial challenges for accurate intrusion detection are discussed in Sect. 7. Finally, a conclusion is drawn in Sect. 8 at the end of this paper. To meet this paper's objectives, we attempt to answer the following research questions, given in Table 1.

2 Background

This section introduces intrusion detection systems followed by a taxonomy of IDS. The main motive of this section is to give an overview of the IDS and its taxonomy.

2.1 Intrusion detection systems

IDS can be a hardware or software system that is used to detect suspicious activity in the network. Monitoring the network, finding breaches and reporting to the administrator are some of the main functions performed by IDS (Vinayakumar et al. 2019; Almomani et al. 2020). Advanced IDS can also take actions when malicious activities are found like blocking the traffic from the source IP Address (Vinayakumar et al. 2019; Chevalier et al. 2020). IDSs can be divided based on different criteria like the technology used, the response of IDSs etc. IDSs can be classified into three categories based on the methodology used in intrusion detection, IDS' reaction, and IDS' architecture as depicted in Fig. 2.

Table 1 Description of research questions arises

RQ#	Research questions	Motivation
RQ1	What is IDS and its taxonomy?	Familiarization of IDS along with its taxonomy
RQ2	What are benchmark datasets used for intrusion detection?	Knowledge of various benchmark datasets used for intrusion detection
RQ3	What are most commonly used performance metrics for evaluating IDSs?	Evaluation of existing IDS using performance metrics
RQ4	What are promising ML methods for accurate intrusion detection?	Identification of ML methods for intrusion detection
RQ5	What are challenges of traditional ML based IDSs?	Finding the reasons for transition from ML to DL methods
RQ6	What are the primary advantages and disadvantages of DL methods in the field of IDS?	Determining the information about DL methods in the field of IDS
RQ7	What are the various challenges of DL methods for intrusion detection?	Identifying the limitations of existing DL methods

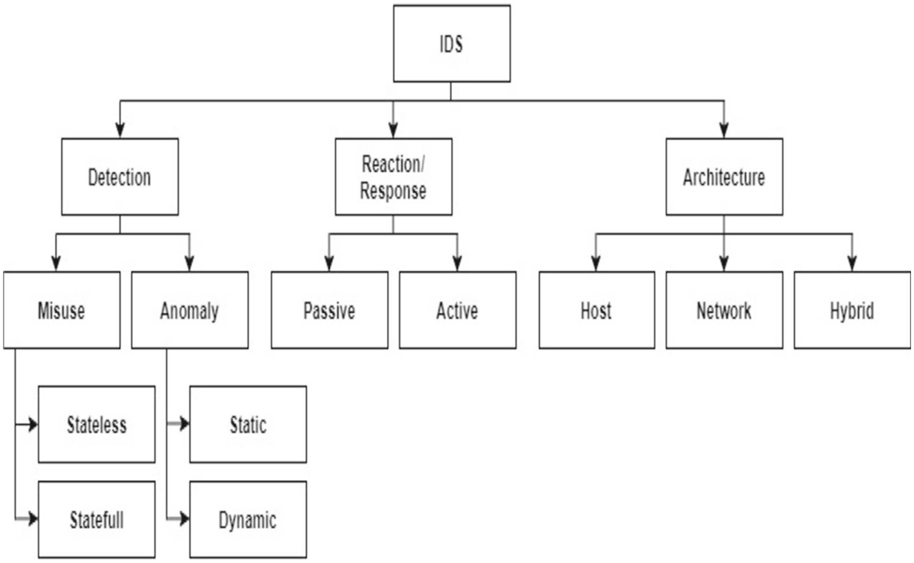


Fig. 2 Classification of IDS

2.1.1 IDSs classification based on detection method

IDS may be categorized into two classes, namely, misuse and anomaly detection. Misuse detection operates with predefined patterns of known attacks, also called signatures. It can be further divided into stateless and state-full IDS (Pandey et al. 2019). State-less methods use only existing signature, whereas state-full methods also use previous signatures and existing signatures (Pandey et al. 2019).

This approach provides high accuracy and low false alarm rates for known attacks but is not practical for detecting novel attacks. One of the known solutions to address this problem is regularly updating the database, which is a time-consuming and costly process. Hence it is not considered as feasible (Kurniabudi et al. 2019).

In contrast, anomaly detection deals with profiling user behaviour. In this approach, a particular model of regular user activity is defined, and any deviation from this model is known as anomalous. Anomaly detection methods can be further categorized into static and dynamic methods (Kurniabudi et al. 2019). The static anomaly detection method works only on the fixed part of the system. The dynamic anomaly detection method extracts patterns (also known as "profiles") from network usage history. This method can detect novel attacks but may lead to high FAR and lacks high accuracy (Kurniabudi et al. 2019; Mäkelä 2019). Another drawback of this system is that an attacker can slowly change its behaviour from abnormal to normal when he feels that he is being profiled. Researchers have also suggested hybrid approaches in the recent past for further improvement in intrusion detection (Guo et al. 2016; Kim et al. 2014).

2.1.2 IDSs classification based on reaction/response method

IDS can be classified into passive and active IDSs (Tidjon et al. 2019; Aljumah 2017) based on type of its response. Passive IDS is set up to only monitor and inform administrator about the intrusions by generating alerts. In contrast, an active IDS can act in real-time by blocking the suspected attack/intrusion (Tidjon et al. 2019; Kim et al. 2014).

2.1.3 IDSs classification based on architecture

IDSs can be divided into three categories based on their architecture, viz., host, network and hybrid IDSs (i.e. a mix of host and network). In a host-based IDS, an agent/sensor is installed on each computer system involved (Feng et al. 2019). It identifies intrusions by analyzing application logs, audit trails, system calls and other activities within the host. In case of a need to generate additional event information/logs, there is a dependency on the developer to modify the operating system kernel code. This approach increases cost which might be unacceptable for some customers (Arabo 2019). Also, deployment of the agent across all computer systems can be cumbersome.

In the network-based system, IDS is installed on the server. The sensors are deployed to identify intrusions by monitoring network traffic across multiple hosts (Chevalier et al. 2020). They are independent of the operating system, are highly portable and easy to implement. However, it shows limitations when high peaks in network traffic or high-speed data are involved. In a hybrid system, IDS is required on the server as well as on each client. It combines host and network approaches and is considered as the most effective and logical approach for intrusion detection (Chevalier et al. 2020; Kurniabudi et al. 2019).

2.2 IDS taxonomy

Figure 3 shows the proposed IDS taxonomy of IDS as per literature analysis. As mentioned in Sect. 2.1, Intrusion detection methods are divided into the anomaly and signature-based methods.

2.2.1 Signature based IDS

Organizations use Signature-based IDS to protect themselves from various known attacks whose signatures are available in the database. This IDS search audited pattern against a series of malicious bytes/known patterns. Signature-based IDS communicate the cause of intrusion alert (Jacob and Wanjala 2018). Signature-based IDS can easily detect known attacks, but it fails to work for new attacks where patterns are not known or not updated in the database. Regular updates of patterns in the database can deal with this issue. But when the user uses advanced technologies in mounting attacks like no operation (NOP) generators, payload encoders and encrypted data channels, signature-based detection does not work well. Its efficiency decreases significantly with creating a new signature for every variation (Rao and Raju 2019). Also, with the increase in the number of signatures, the performance of the system engine decreases. The failure to detect novel attacks and update the database for new patterns regularly are the causes to work in the field of anomaly detection IDS (Rao and Raju 2019; Kang and Kang 2016).

2.2.2 Anomaly based IDS

Anomaly-based IDS detect both network and computer intrusions by monitoring the system. After monitoring, instead of patterns or signatures, it uses heuristics/rules to classify the events as either normal or anomalous and attempts to detect abnormal operation (Farzaneh et al. 2019; Worku 2019). Anomaly detection methods can detect novel attacks but defining its ruleset is a cumbersome task. Anomaly-based IDS are further classified into three classes: knowledge-based, statistical, and machine learning methods as depicted in Fig. 3 described below.

Statistical anomaly IDSs were used for detecting intrusions in information systems earlier. Statistical tests were performed to check whether the observed behaviour is different from the expected behaviour. For statistical approaches, previous knowledge and frequent updates of the signatures are not required. It can detect low and slow attacks, especially DoS attacks. The statistical approach's limitation is the long lead time involved in learning to deliver accurate and valuable results. The most commonly used methods in this category include Markov method, deviation method, multivariate method, and time series method.

Knowledge-based IDS works by gathering knowledge about specific attacks and system vulnerabilities (Hussain and Khan 2020). They work by looking into its knowledge base to identify an attack. Expert System, Petri Nets, Signature Analysis and State Transition are the various examples of knowledge-based IDS. The accuracy rate of results produced using these methods is high, with a low false alarm rate. To keep knowledge-based IDS effective, attack data needs to be updated regularly. The updation of regular data is very time-consuming, which is the main limitation of knowledge-based IDS (Hussain and Khan 2020).

Machine learning is a large field of study that overlaps with and inherits ideas from many related fields such as artificial intelligence. The focus of the field is learning, that is, acquiring skills or knowledge from experience.

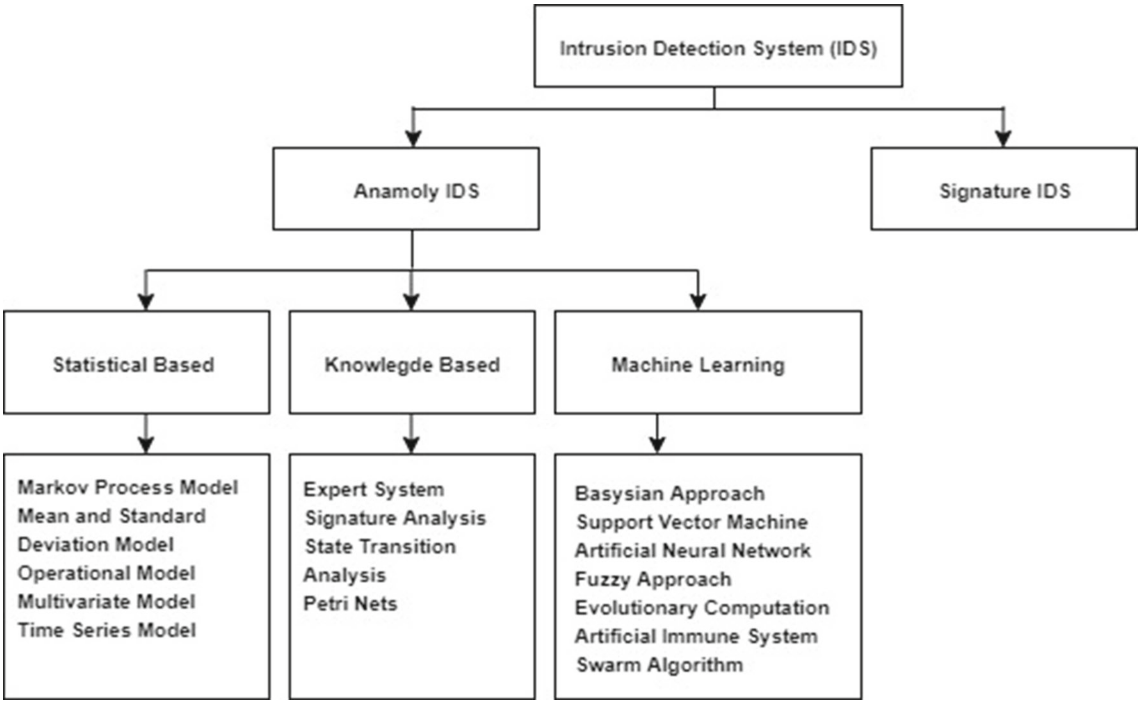


Fig. 3 IDS taxonomy

Most commonly, this means synthesizing practical concepts from historical data. Nowadays, most researchers focus on ML methods due to its built-in properties like robustness, resilience to noisy data and adaptability. Interested researchers can further explore the topic at (Lin et al. 2015; Liao et al. 2013; Aissa and Guerroumi 2016). ML methods proposed for intrusion detection are depicted in Fig. 3 and explained in Sect. 5.

3 Intrusion detection datasets

Several benchmark datasets have been designed to evaluate and compare the performance of IDSs. This section focuses on the most commonly used datasets for intrusion detection.

Training and evaluating IDS need data. So, data is gathered from different sources like network data packets, low-level system information like log files or system dumps etc which is used as benchmark dataset. Datasets are categorized into three types: Synthetic or Self-produced, Benchmark and Real-life datasets, as shown in Fig. 4.

3.1 Synthetic datasets

Synthetic datasets are used to fulfil particular demands or conditions in evaluating IDS. These datasets are used for designing any model for theoretical analysis. These designs can be refined to test and create various types of test scenarios. It facilitates the designers to construct realistic behaviour profiles to test a proposed system for attackers and regular users. It provides initial validation of a particular method; if the outcome proves satisfactory, the developers then continue to evaluate a method in a specific field of real-life data. The shortcomings of other datasets are the reasons for the origin of self-produced datasets. This results in the generation of artificial data and merger into training sets.

3.2 Benchmark datasets

Benchmark datasets include datasets like DARPA 98 KDD Cup99 (Uci 2019), NSL-KDD (Unb 2019a), DEFCON 2000/2002 dataset (Sharafaldin et al. 2018), UNM dataset, CAIDA2002/2016 datasets, LBNL dataset (Gharib et al. 2016), CDX 2009 (Sangster et al. 2009) dataset, Twente 2009 (Sperotto et al. 2009),UMASS 2011, ISCX 2012 (Unb 2019b), ADFA 2013 (Creech and Hu 2013)and CSE-CIC-2018 dataset. These datasets have been used commonly for evaluating IDSs in literature. Among these DARPA 98, KDD Cup99 and NSL-KDD are the most common ones used for the evaluation of IDS shown in Table 2.

DARPA 98, the base dataset of KDD 99 dataset contained raw TCP/ IP dump files. This dataset contained 38 attacks. The training size of dataset was 6.2 Gb and testing Size was 3.67 Gb. The training and testing of data was done for seven weeks and two weeks respectively for this purpose.

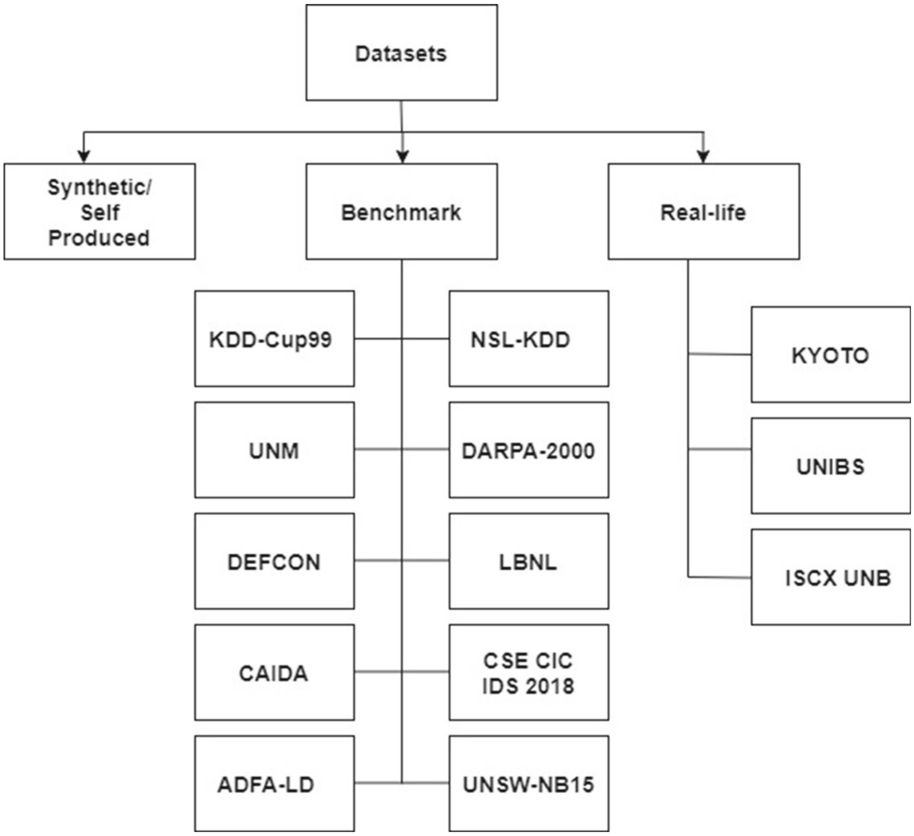


Fig. 4 Benchmark intrusion detection datasets

Table 2 Information regarding Darpa 98, KDD99 and NSL-KDD datasets

Name of dataset	DARPA 98 (base dataset)	KDD99	NSL-KDD
Training size	6,591,458 Kb (6.2 Gb)	4,898,431	125,973
Testing size	3,853,522 Kb (3.67 Gb)	311,029	22,544
Note	Raw TCP/IP dump files	Features extracted and preprocessed for ML	Reduced size by removing duplicates

In 1999, DARPA 98 dataset was summarized with 41-features which is known as KDD 99 benchmark dataset for intrusion detection. KDD 99 dataset covered Probing attacks, DoS attacks, U2R attacks and R2L attacks. KDD dataset was divided into labeled and unlabeled containing 4,898,431 records and 311,029 records respectively. The various types of attacks available in KDD99 dataset are described in Table 3.

Here the training size and testing size of attacks U2R and R2L was very small. This dataset contains huge number of redundant records as shown in Table 4.

The shortcomings of KDD99's related to IDS are well documented in literature (Brugger and Chow 2007; Mahoney and Chan 2003; Sommer and Paxson 2010).

The NSL-KDD dataset is a refined version of the KDD'99. (Tavallaee et al. 2009). Most researchers have applied different methods and tools on NSL-KDD dataset to build effective IDS. The NSL-KDD dataset's analysis using various ML methods available in the WEKA tool is described in Revathi and Malathi (2013). To train and test several novels and existing attacks, NSL-KDD dataset was used by K-means clustering algorithm (Kumar et al. 2013). In Sanjaya and Jena (2014) the comparative study on the KDD99 data set with NSL-KDD dataset was done using ANN and SOM. The ML algorithms are used to analyse various datasets like KDD99, NSL-KDD and GureKDD (Sanjaya and Jena 2014). The various types of attacks in NSL-KDD dataset are described in Table 5.

Improvements in KDD'99 dataset (Unb 2019a) The key advantages of NSL-KDD data set over the original KDD data set are:

1. The classifiers will not be biased toward frequent records due to not inclusion of redundant records in the training set.
2. The performance of the learners is not biased.

Table 3 KDD 99 dataset distribution

Attacks	Normal	DOS	Probe	U2R	R2L	Total
Training size	972,781	3,883,390	41,102	52	1106	4,898,431
%age	19.85	79.27	00.83	00.001	00.02	100
Test size	60,593	231,455	4166	245	14,570	311,029
%age	19.48	74.41	01.33	00.07	04.68	100

Table 4 Statistics of redundant records in KDD cup 99 training and testing datasets

	Attacks	Normal	Total
Training			
Original	3,925,650	972,781	4,898,431
Distinct	262,178	812,814	1,074,992
Redundancy (%)	93.32	16.44	78.05
Testing			
Original	250,436	60,591	311,027
Distinct	29,378	47,911	77,289
Redundancy (%)	88.26	20.92	75.15

3. Reasonable number of records in the train and test sets is available.

Since NSL-KDD is a refined version of KDD-99. So shortcomings of KDD99 also present in the NSL-KDD dataset. Some of the other benchmark datasets depicted in Fig. 4 are explained here.

LBNL (Lawrence Berkeley National Laboratory and ICSI—2004/2005) This dataset contains normal user behaviour. It is not labelled and suffers from heavy anonymization (Gharib et al. 2016).

UNM UNM dataset was proposed in 2004 and unable to fulfil current computer technology trends.

CDX 2009 This dataset was created during network welfare competition to generate a labelled dataset. Attackers used various tools like Web Scarab, Nikto, and Nessus to investigate and detect attacks automatically.IDS alert rules can also be tested by it. Volume and lack of traffic diversity are the limitations of this dataset (Sangster et al. 2009).

Twente—2009 It is a labelled and more realistic dataset. It captured data from a honey pot network and includes few unknown and uncorrelated alerts traffic. It also includes OpenSSH, Apache web server and Proftp using auth/ident on port 113. It suffers from diversity of attacks and lack of volume (Sperotto et al. 2009).

UMASS—2011 This dataset includes various trace files both from wireless applications and network packets (U. of massachusetts amherst 2019; Nehinbe 2011).It lacks a variety of traffic and attacks. This limitation is not beneficial for testing IDS methods (Prusty et al. 2011).

ADFA—2013 KDD and UNM datasets fails to fulfil the present needs of computer technology. ADFA Linux (ADFA-LD) was proposed as a new dataset by Creech and Hu (2013). This dataset is used for the evaluation of ML-based IDS. The attributes of ADFA-LD cybersecurity dataset are challenging to understand. Therefore, there is a need to improve its attributes for better understanding (Abubakar et al. 2015).

CSE-CIC-2018 Dataset This dataset uses a systematic approach to generate a benchmark dataset to detect intrusion. It is based on the creation of user profiles and behaviours seen on the network. This dataset includes seven different attack scenarios.

3.3 Real life datasets

This kind of datasets contains real-life data records. It includes Kyoto 2006/2009, ISCX2012, and UNSW-NB15 datasets.

Kyoto 2006/2009 This dataset contains 14 statistical features (Kyoto2006+ 2015; Song et al. 2011) derived from KDD Cup99 dataset ignoring redundant features. Besides, it also includes ten features for better evaluation and analysis of NIDS. It tries to overcome the limitations of KDD Cup99 dataset. In this dataset, only those attacks are directed at the honey-pots, so it provides a limited view of network traffic. The normal traffic used for simulation during the attacks does not represent normal traffic from the real world. There are no false positives, which are essential for reducing the number of alerts (Song et al. 2011; Sato et al. 2012; Chitrakar and Huang 2012). A comparison of various datasets is given in Al-Dhafian et al. (2015) this paper.

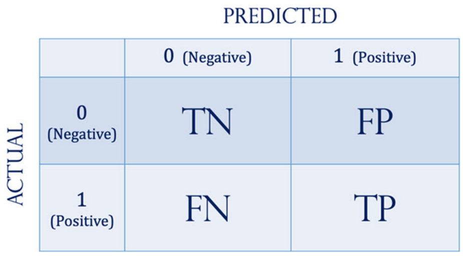
ISCX2012 The dynamic approach was used to generate this dataset. To generate realistic and practical evaluation datasets for IDS, the author presents good guidelines. Alpha and beta profiles are the two parts of this approach. This dataset comprises of relevant profiles and network traces. New network protocols are not considered in this dataset (Shiravi et al. 2012).

UNSW-NB15 TCP-dump tool was used to capture raw traffic. It was used for academic research purpose and contained a hybrid of normal activities and attack behaviours. To generate this dataset, twelve algorithms and tools were used.

Based on the literature, it is concluded that different researchers used different datasets as per their requirements.

Table 5 Distribution of instance in NSL-KDD dataset

Attacks	Normal	DOS	Probe	U2R	R2L	Total
Training size	67,343	45,927	11,656	52	995	125,973
%age	53.458	36.458	9.253	0.041	0.790	100
Test size	9711	7456	2421	200	2756	22,544
%age	43.076	33.073	10.739	0.887	12.225	100



		PREDICTED	
		0 (Negative)	1 (Positive)
ACTUAL	0 (Negative)	TN	FP
	1 (Positive)	FN	TP

Fig. 5 Confusion matrix

In the literature, KDD Cup99 and NSL-KDD dataset are primarily used for evaluating ML-based IDSs. KDD Cup99 dataset does not represent real traffic data. NSL-KDD is the refined version of KDD Cup99, but shortcomings of KDD Cup99 also present in NSL-KDD dataset. Both of these datasets are very old. So, there is a need to use more than one datasets to validate the performance of IDS (Table 6).

4 Performance metrics

IDS effectiveness can be judged by performance evaluation in terms of metrics. It can be evaluated based on different metrics computing using confusion matrix described below.

4.1 Confusion matrix

Confusion matrix often used to describe the performance of classification models. It summarizes performance of a classification algorithm by giving predicted result. It contains information regarding different combinations of actual and predicted classifications as shown in Fig. 5.

There are four components in confusion matrix True Positives (TP), False Positives (FP), True Negatives (FN) and True Negative (TN). TP means the actual class and the predicted class of data points both are 1 (true). It represents the attacks that the IDS successfully detects. FP refers to the normal behaviour being wrongly classified as attacks by IDS. FN means 0 (false) attack events that are missed by the IDS incorrectly classified as normal events 1 (true), and TN refers to the actual class and the predicted class of data points both are 0 (false). FP is referred to as Type I error and FN is referred to as Type II error. Confusion matrix is a powerful tool in classification, but its performance is not suitable for comparing IDS. To solve this problem, different performance metrics are described with the help of confusion matrix variables. The performance metrics gives output in the form of numeric values, which are easy to compare. The most common metrics are described below.

- *Accuracy* It describes how much the classifier is correct. It is the ratio of correct predicted samples to the total number of samples and can be computed as Eq. 1:

$$\frac{(\mathrm{TP}+\mathrm{TN})}{\text{Total Number of Instances}} \tag{1}$$

  It is a perfect metric for balanced data but diminishes its value in the case of imbalanced data.
- *Detection rate (DR)* It is also known as Sensitivity/Recall. It refers to the percentage of actual attacks correctly identified by the system and can be expressed as:

$$\frac{\mathrm{TP}}{(\mathrm{TP}+\mathrm{FN})} \tag{2}$$

  It provides information on the classifier's performance concerning false negatives.

$$\frac{\mathrm{FP}}{(\mathrm{TN}+\mathrm{FP})} \tag{3}$$

- *Specificity* It measures the proportion of negatives that are correctly identified by the system. This performance metric can be calculated with the help of Eq. 4:

$$\frac{\mathrm{TN}}{(\mathrm{TN}+\mathrm{FP})} \tag{4}$$

- *False alarm rate (FAR)* The ratio of false-negative samples to total positive samples is known as FAR and can be calculated by Eq. 5:

$$\frac{\mathrm{FN}}{(\mathrm{TP}+\mathrm{FN})} \tag{5}$$

- *Precision* It is an important metric and tells what percentage of our true precision is true. It helps to evaluate the model better and can be calculated with the help of Eq. 6:

$$\frac{\mathrm{TP}}{(\mathrm{TP}+\mathrm{FP})} \tag{6}$$

Table 6 Comparison of different datasets and their description

Dataset	Total instances	Attack type	Total Features	Ref	Advantages	Disadvantages
KDD-Cup 99	5,000,000 Imbalanced classes	Normal ,DoS, Probe, U2R, R2L	41	(Uci 2019)	Easily available	Imbalanced, not included modern attacks
NSL-KDD	Training set = 489,431 Test set = 311,027	Normal, DoS,Probe, U2R, R2L	41	(Unb 2019a)	Removes redundancy, eliminating the unbalancing problem in training and testing dataset	Does not represent the modern low foot print attack scenarios
Kyoto 2006+	Over all traffic (2006-2009)	Multiple	24	(Kyoto 2019)	Includes statistical features such as source byte and average count along with 10 additional features for IDS	Not effective on the hybrid features of the latest honey pots data sets
ISCX 2012	Training Dataset = 9 Testing Dataset = 9	Normal Attack	9	(Unb 2019b)	Allow dynamic attacks for the hybrid approach i.e Dos and SSH brute force	Not able to identify the characteristics based network errors
CICIDS2017	Contains total 5 days data, i.e. Monday to Friday	Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet and DDoS	80	(Unb 2019c)	Contains traffic based on bidirectional flow-based format and packet-based with additional 80 attributes	Unusable in case of application layeror modern Dos attacks
DEFCON	Contain only attack traffic during DEF-CON competition	Port Scan, BufferFlow attacks	–	(Sharafaldin et al. 2018)	Significant to avoid network interruption. Performed better as compared to CTF and MAC-CDC	It doesn't function for the normal background traffic as opposed to the intrusive traffic
DARPA	Multiple datasets	DoS, Probe, U2R, R2L	–	(Sharafaldin et al. 2018)	Mainly useful for web application activities such as sending and receiving files through FTP, browsing websites, sending and receiving mails and monitoring routers	The data set not deals with the noisy data injected artificially and the benign attack as well
UNSW-NB15	Training set = 175,341 Testing set = 82,332	Fuzzers Analysis, Backdoors, DoS, Exploits, Generic	49	(Cloudstor 2019)	It comprises major categories of benign attacks such as worms, fuzzers, exploits, and DoS. Useful when testing model against multi-IP addresses	Only functional for TCP and UDP connection. It can not handle a high number of DNS at a time

- *System utilization* It means the amount of CPU and memory utilization required for IDS.

4.2 Receiver operating characteristic (ROC)

ROC analysis is concerned with a field called "Signal Detection Theory" (Signal detection theory 2019). During World War II, electrical engineers and radar engineers first developed the ROC Curve to detect enemy objects on battlefields. The performance of different systems can be compared effectively with ROC Curves. It is a plot between TPR and FPR for the different possible cut-points of a diagnostic test. For many decades it is increasingly used in ML research. ROC Curve is used to count the detection costs and evaluates various detection learning methods in intrusion detection. DR and FAR are mainly used performance metrics. High DR and low FAR is preferred for IDS.

5 Machine learning methods for IDSs

ML is a branch of AI that learned or adapted to the new environment. It allows programs to finds and learns the patterns within data. It explores various methods, also called ML methods, that can learn from and then make predictions on data. ML methods usually operate based on the features that represent the characteristic of the object.

It is an interdisciplinary field that draws on ideas from various disciplines, including mathematics, science, and engineering. Face recognition, which allows users to tag and post images of their friends on social media, Optical character recognition (OCR), Recommendation engines, Self-driving vehicles, Image recognition, Speech recognition, Medical diagnosis, Virtual personal assistant, E-Mail spam and malware filtering, Online fraud detection, and several other problems have been solved with it.

In general, ML is divided into three sub-domains: supervised, unsupervised, and reinforcement learning as shown in Fig. 6.

Supervised learning requires labelled data for training (both inputs and desired outputs). It discovers the relationship between data and its class, while unsupervised learning is used when labelled data is not available. These methods find the hidden pattern in the data. Reinforcement learning is based on a feedback mechanism. Here, computer program interacts with the environment and learns by experience. Several ML methods have been proposed for accurate intrusion detection. The most commonly used methods are summarized in the following sub-sections.

5.1 Artificial neural networks (ANN)

ANNs are designed based on biological neural networks. They learn from examples and generalize from noisy and incomplete data to perform tasks. The original aim of the ANN approach was to solve problems similar to the human brain. Such systems are successfully employed for data-intensive applications. The various types of ANN and their contributions and performances on intrusion detection will be discussed in this section. Several ANN designs have been proposed based on different learning strategies as depicted in Fig. 7.

5.1.1 Supervised ANN models

In this learning, we train ANN model using labelled data and a new set of examples. ANN model analyzes training data and produces a correct outcome from labelled data. Feed-forward neural network and recurrent neural network (RNN) are examples of supervised learning.

Feed forward neural network was the first and most straightforward type of ANN. In this network, information is transferred from the input nodes to the hidden nodes and through hidden nodes to the output nodes in only one direction forward. This type of network does not form a cycle.

Single Layer Perception (SLP) consists of a single neuron with adjustable weights and bias. It is used to classify linearly separable patterns, and the training in the perception continuous until no error occurs. MLP and RBF are two examples of Feed-Forward ANNs used for modelling patterns. Static backpropagation is used to train the MLPs networks. This network's advantage is that they are easy to handle and can approximate any input/output map. The disadvantages of MLPs are slow training and requiring a lot of training data.

Several researchers used ANN in the supervised mode for detecting intrusions. For instance, Gupta et al. (2012) have used a feedforward neural network for predicting several zombies involved in flooding DDoS attacks. In this paper, the relationship between the zombies and sample entropy is identified. The zombies are predicted involved in a DDoS attack with significantly less test error. A generalization of the MLP over one or more layers is known as Generalized feedforward (GFF) networks. In real life, GFF networks often solve the problem much more efficiently than MLP. Akilandeswari and Shalinie (2012) used a Radial Basis Function Neural Network (RBFNN) for classifying DDoS attack traffic and regular traffic. This method achieves the highest accuracy for DDoS flooding attacks. RNNs are connectionist models that capture the dynamics of sequences via cycles. It is a sequential learning model and learns features from the memory of previous inputs. It shows promising results in ML tasks when input and output are of variable length.

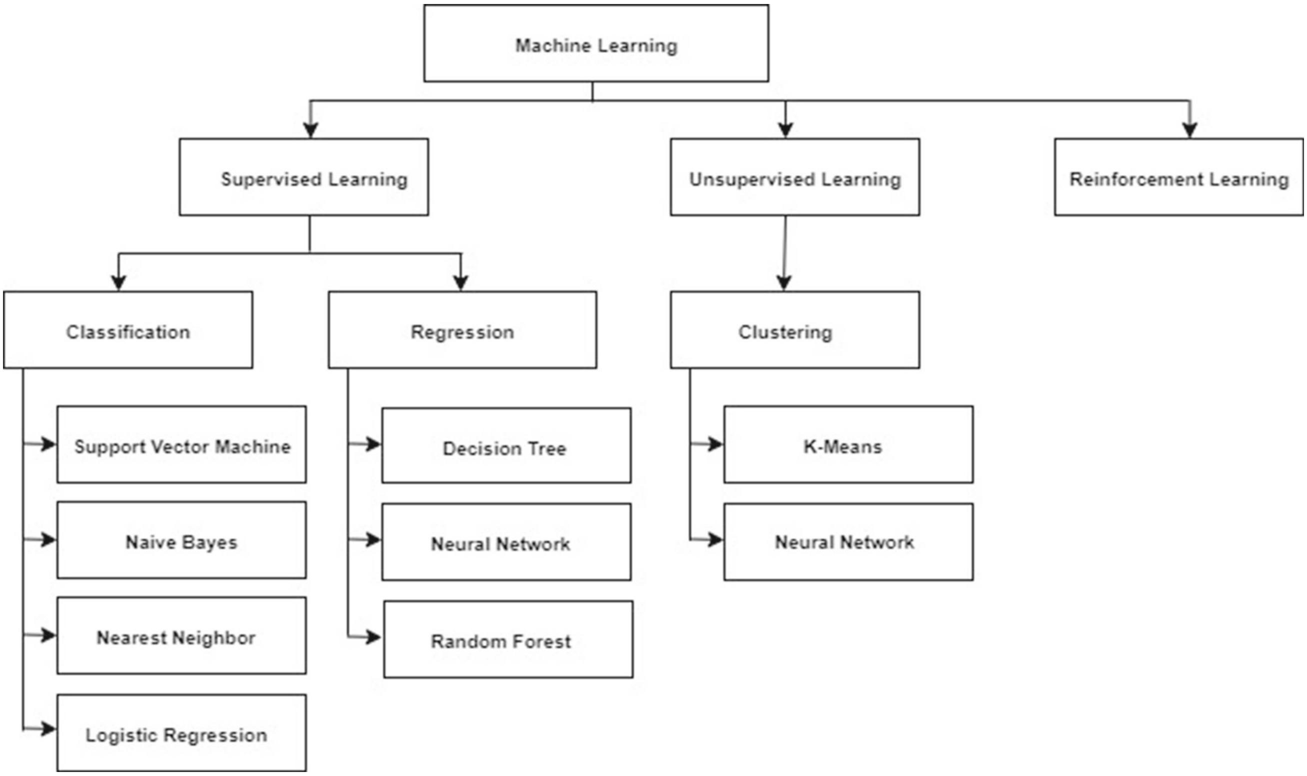


Fig. 6 Machine learning methods

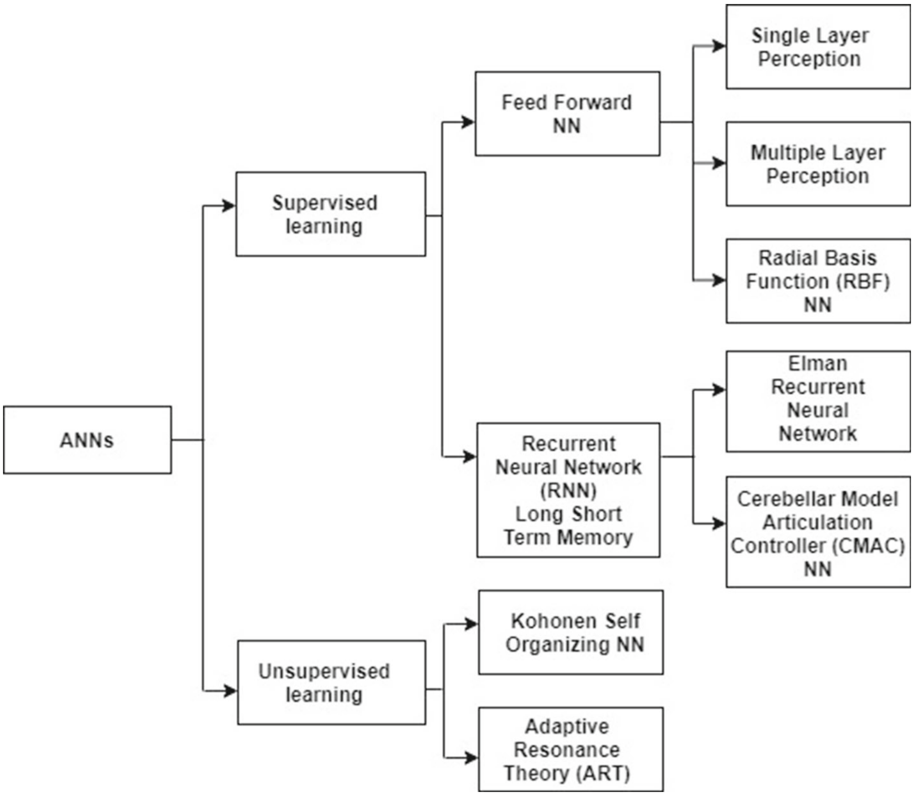

ANNs
Supervised learning
Unsupervised learning
Feed Forward NN
Recurrent Neural Network (RNN) Long Short Term Memory
Kohonen Self Organizing NN
Adaptive Resonance Theory (ART)
Single Layer Perception
Multiple Layer Perception
Radial Basis Function (RBF) NN
Elman Recurrent Neural Network
Cerebellar Model Articulation Controller (CMAC) NN

Fig. 7 ANN models

Tong et al. (2009) reported a hybrid RBF/ Elman neural network model for anomaly and misuse detection. Elman network is used to restore past events and RBF network memory as real-time pattern classification. The results show that the IDSs using this hybrid neural network improve the Detection Rate(DR) and effectively decrease the false positive rate. Aljumah (2017) has used a trained ANN algorithm to detect various attacks. A mirror image of a real-life environment was used for learning. The author got 98% detection accuracy. The old and up-to-date datasets were used to train the algorithm for further evaluation. This approach is not able to handle DDoS attacks.

5.1.2 Unsupervised ANN models

Training of a machine without a teacher is known as unsupervised learning. It uses information that is neither classified nor labelled. Kohonen Self Organizing Map and Adaptive Resonance Theory (ART) come under the category of unsupervised learning. SOM is used to build a 2D map of a problem space using unsupervised learning. It can generate a visual representation of data on a rectangular grid. Nonlinearity is the main advantage of SOM networks. It can preserve the topological structure of the data. It clusters the samples into predefined classes and then orders the classes into meaningful maps. It comprises two layers, i.e. input and output layer.

Several researchers used ANN in an unsupervised mode for detecting intrusions. For instance, Chen et al. (1996) described a multi-layered SOM algorithm, which permitted unlimited layers of Kohonen maps, also called M-SOM. This algorithm has been tested in many applications like internet entertainment-related home-pages and electronic brainstorming comments. According to Kalteh et al. (2008) SOM applications are based on ad-hoc approaches and featured by trial and error approaches. They perform better than other methods to solve various problems in cases like climate and environmental issues.

Ibrahim et al. (2013) implemented the SOM to detect anomalies on KDD dataset and NSL-KDD dataset. The author achieved 92.37% attack detection with KDD dataset and 75.49% with NSL-KDD dataset. The SOM network's advantage is its high speed and fast conversion rates compared with other learning methods.

Adaptive Resonance Theory (ART): These are self-organizing neural architectures. It clusters the pattern space and produces appropriate weight vector templates. Stephen Grossberg invented it in 1976. The resonance is related to the resonant state of a neural network. Conventional ANNs have failed to solve the stability-plasticity problem. ART algorithms solve the problem of plasticity, which is required to learn new patterns. It is an unsupervised learning model.

Many researchers, Aljumah (2017) used ANN to find DDos attacks while others (Tong et al. 2009) used a hybrid neural network for both misuse and anomaly detection. The following points can be concluded on the basis of contributions given by researchers:

1. Network data traffic can be filtered and modelled more efficiently using ANN. Always train the ANN with a new dataset instead of the old dataset. Otherwise, it will display poor results.
2. RBF takes less time to train compared to MLP.
3. Adhoc approaches are generally used in SOM applications.
4. SOM has high speed and fast conversion rates as compared with other learning methods.
5. Hybrid networks are required to improve the DR and decrease FPR.

5.2 Support vector machine (SVM)

It is a supervised model used for classification, regression and outlier detection. It linearly separates the data based on the hyperplane. SVM maps the data into feature space and divides it into classes using a hyperplane with the most significant margin between the classes' instances. It is a binary classifier that can also do multi-class classification. SVM is most useful when dealing with nonlinear data.

Several researchers used SVM for detecting intrusions. For instance, Wang et al. (2017) proposed SVM model for detecting network intrusions. To improve detection efficiency, the authors emphasized the importance of high-quality training data. The authors proposed an efficient IDS based on enhanced SVMs. To obtain new and better-quality SVM detection, they introduced a logarithm marginal density ratio transformation(LMDRT). The empirical results showed practical values such as high DR and good efficiency.

Wang's work was expanded by Gu et al. (2019) by introducing an ensemble-based intrusion detection model based on the LMDRT transformation, which also achieves competitive intrusion detection outcomes. Kabir et al. (2018) proposed optimum allocation based most miniature square support vector machine (OA-LS-SVM) based on the idea of sampling. This method can handle both static and incremental data. The suggested technique is explored and validated using the KDD 99 dataset. In terms of accuracy and performance, the proposed method achieves a realistic result. Similarly, Gu and Lu (2021) proposed an efficient IDS based on SVM and naive Bayes feature embedding. Four datasets UNSW-NB15, NSL-KDD, Kyoto 2006 and CICIDS2017 were selected for the experiment. The result showed that the proposed detection approach achieved strong and robust results, with an accuracy of 93.75% on the UNSW-NB15 dataset, 98.92% on the CICIDS2017 dataset, 99.35% on the

NSL-KDD dataset, and 98.58% on the Kyoto 2006+ dataset. Key findings of SVM studies include followings:

1. Training time is more in SVM.
2. SVM is most useful when dealing with nonlinear data.
3. Single SVM still has a significantly higher FAR.
4. Because of its promising performance in classification and prediction, the Support Vector Machine (SVM) is becoming more popular.

5.3 Naive Bayes (NB)

It is a classification algorithm defined based on Bayes Theorem. This classifier assumes that the probability of every feature belonging to a given class value is independent of other features. Prediction can be attained by calculating the instance probabilities of each class and by selecting the class value of the highest probability.

Several researchers used NB for detecting intrusions. For instance, Kevric et al. (2017) developed a combining classifier model using random tree and NBTree algorithms for NIDS. This algorithm was evaluated on NSL-KDD dataset and accuracy achieved was 89.24%. It was also concluded that combining the two best individual classifiers could not result in the best overall performance. Depending on the form of attack, a hybrid layered IDS was proposed by Çavuşoğlu (2019) that employed various ML methods. NSL-KDD dataset was used for training and testing. Transformation and normalization operations were performed on the dataset. In all attack types, the results revealed that the proposed method achieved high accuracy and low FPR.
The SVM and NB feature embedding was used by Gu and Lu (2021) to develop an efficient intrusion detection system as discussed in SVM section. Key findings of NB studies include the followings:

1. The NB classifier performs the best on real-time dataset.
2. On a variety of classification tasks, NB algorithms were found to be surprisingly accurate on small datasets.
3. The precision of NB does not scale up and decision trees in specific, more extensive databases.

5.4 k-nearest neighbour (kNN)

KNN used both for classification and regression problems, but it is most appropriate for classification problems. It is a lazy learner and simples stores all the training data. It uses this data to find the similarities between available data and new data. Based on the Euclidean distance, the test data is allotted to the class of kNN. This method is computationally expensive.

Several researchers used NB for detecting intrusions. For instance, Guo et al. (2016) developed a hybrid method to achieve a high DR with a low FPR. The system was based on a two-tier hybrid approach that includes two anomaly detection components and a misuse detection component. In stage 1, a low-complexity anomaly detection method was built and used to construct the detection portion. In order to construct the two detection components for stage 2, the k-nearest neighbour's algorithm was used. The stage 1 detection component was involved in creating the two-stage detection components that reduce the number of false positives and false negatives produced by the stage 1 detection component. The experimental results showed that this approach could effectively detect network anomalies with a low FPR on the KDD'99 dataset and the Kyoto University Benchmark dataset.

Saleh et al. (2019) proposed a hybrid IDS to handle the multi-class classification problem. It was based on a triple edged strategy due to its three main contributions, which were: (i) NBFS, employed for dimensionality reduction, (ii) OSVM, applied for outlier rejection, and (iii) PKNN, used for detecting input attacks. The KDD Cup '99, Kyoto 2006+ and NSL-KDD datasets were used to compare the HIDS against recent techniques. It was capable of detecting attacks rapidly and can be employed for real-time intrusion detection.

5.5 Logistic regression (LR)

LR estimates the discrete values in the form of 0 or 1 based on independent values. Fitting data will predict the event that will have occurred or not to the logistic function. 0.5 is considered a threshold, and the values greater than 0.5 are considered as 1 or lower than 0.5 is considered 0.

Several researchers used NB for detecting intrusions. For instance, Palmieri (2019) introduced a novel network anomaly detection approach focused on nonlinear invariant properties of Internet traffic. The overall findings showed that the method effectively isolates a wide range of volumetric DoS attacks in the sense of complex traffic flows with high accuracy and precision.

Key findings of LR studies include the followings:

1. LR is known for its high performance, low computational burden, and good interpretability.
2. It also produces well-calibrated prediction probabilities without requiring any scaling or tuning of its input features.
3. LR outperforms other probabilistic classifiers by being more tolerant of feature correlation, allowing it to make better predictions even though multiple correlated features are present.

5.6 Decision tree (DT)

DT is used for both regression and classification problems, but it is mainly used for classification problems. A regression tree is one with continuous values, whereas a decision tree is one with a range of symbolic labels. It classifies a sample through a sequence of decisions represented in a tree structure, in which the current decision helps to make the subsequent decision. Such a sequence of decisions is represented in a tree structure. Classification and Regression Tree (CART) is a popular program for constructing decision trees.

Several researchers used DT for detecting intrusions. For instance, Kim et al. (2014) proposed a hybrid intrusion detection method based on the misuse and anomaly detection. The experiment was conducted on NSL-KDD dataset. The proposed method was better in terms of DR, low FPR and reduced time complexity. The proposed method's ability to reduce time was not as good as it could be. As a result, future research will concentrate on improving the C4.5 decision tree algorithm. Similarly, Mousavi et al. (2019) also proposed IDS based on ant colony optimization and decision trees' ensemble. In this method,16 essential features were selected for representing different network visits using a gradually feature removal method. The accuracy of 99.92% was obtained using the proposed method.

5.7 Random forest (RF)

RF, as the name suggests, constructs a forest with several decision trees. It is created by combining several decision trees, which predicts by averaging the predictions of each component tree. It is generally much more accurate than a single indicator. In general, the more trees in a forest, the more robust it appears.

Several researchers used RF for detecting intrusions. For instance, Farnaaz and Jabbar (2016) proposed a model based on RF classifier for intrusion detection. RF was used as an ensemble classifier and outperformed other conventional classifiers in terms of successful attack classification. The results showed that the proposed model was efficient with low FAR and high DR. Belavagi and Muniyal (2016) proposed a model for intrusion detection using ML classifiers on NSL-KDD dataset. The results concluded that the RF classifiers outperformed other classifiers, and the accuracy obtained was 99%. Hasan et al. (2019) discussed several ML models' accuracy for predicting attacks and anomalies on IoT systems. The accuracy obtained for DT, RF and ANN classifiers was 99.4%, but in terms of other performance metrics, RF classifier outperformed other classifiers. Saranya et al. (2020a) explored the comparative study of ML algorithms used in IDS on KDD cup dataset. The accuracy obtained was 99.65%, 98.1% and 98% for RF, LDA, and CART algorithms. It was observed from the results that RF outperformed other classifiers in terms of accuracy and concluded that the classifiers' performance was also dependent on the application used and the size of the dataset. Key findings of RF studies include the followings:

1. While increasing the trees, the RF adds more randomness to the model. When splitting a node, it looks for the best function among a random subset of features rather than the most appropriate feature. As a consequence, there is a lot of variation, which leads to a better model.
2. Random forest's versatility is one of its most appealing features. It can be used for both regression and classification tasks, and the relative importance it assigns to the input features can be easily viewed.
3. Overfitting is one of the most common problems in ML, but RF classifier will not overfit the model if there are enough trees in the forest.

5.8 K-means clustering method

It is one of the unsupervised ML algorithms. Like an unsupervised algorithm, there is no labelled data in this method. This algorithm works based on the finding groups in the data. It groups objects into clusters based on their similarities and differences with objects in other clusters. K-means algorithm is highly used in time series data for pattern matching. The K-Means algorithm has the disadvantage of not applying to non-spherical results.

Several researchers used K-means method for detecting intrusions. For instance, Mohamad Tahir et al. (2015) proposed a hybrid ML method for NIDS centred on a combination of K-means clustering and SVM classification. The NSL-KDD dataset was used for evaluation and the results obtained were a positive DR and reduced FAR. In another work, Al-Yaseen et al. (2017) suggested a changed K-means method for reducing the training dataset's size and balancing the data for SVMs and Extreme Learning Machines training (ELMs). The experimental results obtained were 95.75% accuracy with a FAR of 1.87%.

5.9 Fuzzy systems

In the early 1960s, Zadeh initiated fuzzy set theory to deal with problems like incomplete information. It is an essential tool used to analyze the security of a place and begin for scientific applications. Fuzzy logic was introduced for intrusion detection, mainly due to quantitative features and security (Luo 1999). Fuzzy set theory assigns values ranging from 0 to 1 (Tsoukalas and Uhrig 1997). An object can belong to different classes simultaneously in fuzzy logic, which is beneficial when the difference between classes is not adequately defined. Due to this concept, fuzzy theory can be applied in intrusion detection when the differences between the normal and abnormal classes are not well defined (Gomez and Dasgupta 2002). Fuzzy sets help in recognizing dangerous events and reducing false alarms level during intrusion detection.

Several researchers used Fuzzy logic in detecting intrusions. For instance, Porras et al. (2002) proposed the EMER-

ALD Mission Impact Intrusion Report Correlation System or M-Correlator to alert prioritization and aggregation. It is an alert ranking technique. These methods work better for misuse-based IDSs than anomaly-based IDSs. Qin and Lee (2003) discussed the alert score to describe the cruelty of attack and its applicability. Yu et al. (2004) presented multiple IDS to detect real-time network intrusions. In this paper, a novel IDS alert management system known as FuzMet was discussed (Alsubhi et al. 2012). It extends the works of Porras et al. (2002), Yu et al. (2004) which is used both for misuse and anomaly-based IDSs. Kudłacik et al. (2016) presented a fuzzy-based intrusion detection method. It consists of two profiles of the user's activity, i.e. local profile and fuzzy profile. This method has low computational complexity, and due to this, the monitoring server can process a large number of incoming local profiles in real-time.

Key findings of the review of fuzzy logic-based methods are that fuzzy logic builds flexible patterns for detecting intrusion. The fuzzy theory can differentiate between abnormal and normal class in intrusion detection. It enhanced the readability as well as the understanding ability of some ML algorithms. Various researchers used fuzzy logic or fuzzy sets to recognize the dangerous events and reduce false alarm rates.

5.10 Evolutionary computation

Evolutionary computation (EC) is a problem-solving technique of computational intelligence motivated by natural and biological evolution. Traditional systems are unable to solve complex problems. So, researchers have been using evolutionary computational methods to solve such problems. EC is an idea through which a computer can develop its solutions to problems rather than write the computer program manually by going through complicated steps. As a result computer program could be ready in a matter of minutes. It enables computers to solve complex real-world problems that are difficult for a human being to tackle. The researchers have used EC for automatic model design, optimization, and even learning for classification in intrusion detection. In this section, some critical issues like the working of EC, EC methods, and algorithms used in EC will be discussed. After the initialization of candidate solutions, new solutions are created by applying mutation and crossover operators. The resulting solution's evaluation is done based on their fitness, and after this, the selection is applied to find solutions for the next generation. A flow chart of EC is depicted in Fig. 8.

Genetic algorithms (GA), genetic programming (GP), grammatical evolution (GE), evolutionary algorithms (EA), evolutionary programming, evolution strategy, learning classifier system etc., are examples of EC methods. These methods can be differentiated based on representing the individuals like GP uses trees; GE uses the Backus-Naur Form (BNF) grammar. GA is implemented as chromosome-like data structures and uses parameters, operators and processes like selection, crossover, mutation and fitness function to arrive at a particular solution. Several researchers used EC methods for detecting intrusions. For instance, Li (2004) has applied GA to identify anomalous network behaviours. Crosbie et al. (1995) applied the multiple agent technology and GP to detect network anomalies. The proposed methodology has the advantage when many small autonomous agents are used. The training process can be time-consuming if the agents are not correctly initialized or communication occurs among the agents. Abdullah et al. (2009) has used GAs for getting classification rules for intrusion detection. Ojugo et al. (2012), has applied GAs to build rule-based intrusion detection. Maniyar and Musande (2016) revised the genetic algorithm to generate the rules to detect or classify attacks using network audit data, and fitness function is used for the selection of rules. GA based IDS can be implemented in two steps, i.e. to generate classification rules and use these rules for intrusion detection. For intrusion detection, GA has to go through a series of steps which are discussed below:

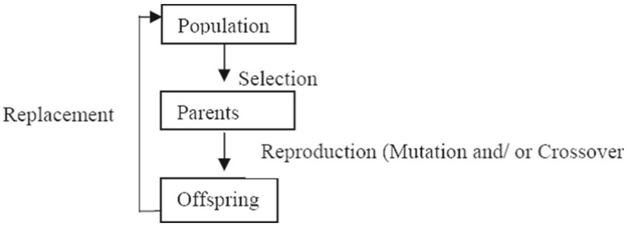


Fig. 8 Flow chart of evolutionary algorithm

1. Information about the network traffic is collected by the sniffer present in the IDS.
2. On this captured data, IDS applies GA. The collected information is used to frame classification rules.
3. The set of rules of the previous phase are then applied to the incoming traffic by IDS, resulting in population initialization. A new population having good qualities is generated as a result. After this evaluation is performed on this population, and a new generation with better qualities is generated. Then genetic operators are applied to the newly created generation until the most suitable individual is found.

The implementation of GA is depicted in Fig. 9.

In the literature, GP is the most popular technique of EC. GP is the extension of GA and was introduced by Koza in 1992. It is a domain-independent method, and to solve a problem, GP genetically breeds a population of computer programs. Le Goues et al. (2011) described and evaluated genetic Program Repair technique based on existing test cases. It automatically generates repairs for real-world

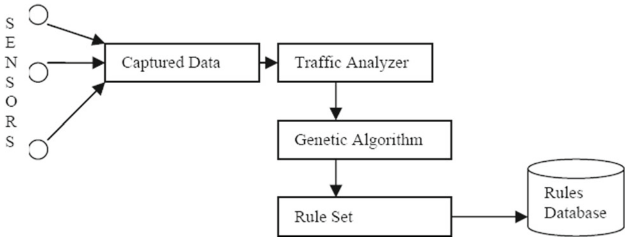


Fig. 9 Genetic algorithm's working

bugs in legacy applications. GenProg can efficiently repair programs containing multiple errors drawn from multiple domains.

Jebur and Nasereddin (2015) introduced Fuzzy-genetic IDS combined with feature selection. It allows the system to develop an optimal subset of an attribute in the middle of enormous network information. To reduce the training time, the author uses 15 features to describe the rules. Fuzzy logic is used to generate rules. The soft computing approach generates more efficient rules than complex computing. Further, a GA is applied to generate essential rules by tuning. The feature selection strategies perform poorly in the case of unbalanced data.

To solve this problem, Viegas et al. (2018) proposed a new feature selection technique based on Genetic Programming that works well with balanced and unbalanced data. It is capable of selecting a set of discriminative features. Biological and Textual datasets are used for evaluation. The solution proposed by the author improves the efficiency of the learning process and also bringing down the size of the data space. Besides GA and GP, Grammatical Evolution (GE) is a technique based on biological process. With GE's help, complete programs can be generated in an arbitrary language by developing programs written in BNF grammar. The evolution process can be performed on variable-length binary strings instead of actual programs. This transformation provides mapping, which simplifies the application of search.

Şen and Clark (2009) applied GE technique on route disruption and DoS attacks on MANETs. Intrusion detection programs are developed for each attack and distributed to every node on the network. GE technique shows good performance on evolving efficient detectors for known attacks. Nyathi and Pillay (2018) compared the GA to GE to automate GP classification algorithms' design. This approach is trained and tested using real-world binary and multi-class data. The result shows that GE is suitable for binary classification while the GA is suitable for multi-class classification.

Evolutionary Algorithms (EAs) are black-box search optimization methods based on population and not required assumptions like continuity or differentiability. They are very appropriate for dealing with MOPs (Yang et al. 2013). To summarize above, our findings are that due to having simple structures, EC has shown excellence to represent the possible solutions to a large variety of problems. It plays a vital role for classifiers learning, optimization and automatic model design. These algorithms are easily transferable from one application to another. The important application areas of evolutionary algorithms are numerical and combinatorial optimization. Black-box optimization is the most challenging. Following features make the EAs attractive:

1. They make no explicit assumptions about the problem. Due to this, they are widely applicable and can be transferable at a low cost.
2. They are flexible and can be easily used in collaboration with existing methods.
3. They are strong due to randomized choices.
4. They are also less sensitive to noise.
5. Algorithm terminates with several solutions and not focused on a single solution.

5.11 Swarm intelligence

Beni and Wang (1993) firstly triggered the term "Swarm Intelligence" (SI) for cellular robotics system and later on for problem-solving in AI. It provides a distributed solution to complex problems by interactions between agents and their environment. Self-organization and division of labour are the two necessary properties of SI. Self-organization is the capability of a system to devolve its agents without any external help and labour division. It refers to the parallel execution of feasible and straightforward tasks that enable it to solve complex problems. The two popular swarm inspired methods are ACO and PSO. ACO simulates ants' behaviour and suitable for discrete optimization problems whereas PSO simulates the behaviour of flocks of birds and is used to solve nonlinear optimization problems. The ants foraging behaviour inspires ACO. The indirect communication between the ants utilizing chemical pheromone trails enables them to find short paths between their nest and food sources. ACO algorithms are used to solve computational and discrete optimization problems. Researchers have applied ACO algorithms to solve complex problems like Traveling Salesman, Vehicle Routing and Telecommunication network etc.

Several researchers applied SI for detecting intrusions. For instance, Tabakhi et al. (2014) described an unsupervised feature selection method based on ant colony optimization (UFSACO). This technique is used to find the optimal feature subset with several iterations without using any learning algorithms. The redundancy is minimized by the computation of feature relevance based on the similarity between features. Hence it is classified as a filter-based multivariate method. It exhibits low computational complexity. The result indicates that the method outperforms the unsupervised methods and comparable with the supervised methods. Aghdam and Kabiri (2016) applied ACO for intrusion detection problem area using dimensionality reduction. Due to solid search capability, it could efficiently found minimal feature subset. This technique uses KDD Cup 99 and NSL-KDD benchmark data sets for intrusion detection and obtained higher accuracy with a lower false alarm. Hajimirzaei and Navimipour (2019) proposed a hybrid approach for intrusion detection. NSL-KDD dataset and CloudSim simulator are used, root mean square error (RMSE), mean absolute error (MAE), and the kappa statistic is chosen for evaluation criteria. This hybrid approach gives better results than earlier methods. Kennedy, Eberhart and Shi introduced the PSO as an optimization technique to guide the particles to seek optimal global solutions. Various researchers in intrusion detection using this technique due to several advantages like ease to implement,simplicity, robustness,scalability, fast finding optimal solution and flexibility, etc. To improve the accuracy of attack detection, Bamakan et al. (2015) presented a new method based on multiple criteria linear programming(MCLP) and PSO. MCLP is a classification method and is capable of solving real-life data mining problems. It is based on mathematical programming. To improve the performance of MCLP classifier, PSO, a robust and straightforward technique was used. KDD CUP 99 Benchmark Datasets are used to evaluate the performance. PSO-MCLP model shows the high accuracy of 99.13 percentage and a low FAR of 1.947 percentage.

Similarly, Bamakan et al. (2016) again proposed a time-varying chaos particle swarm optimization method (TVCPSO). This technique is based on two conventional classifiers, i.e. MCLP and SVM, to detect intrusion using NSL-KDD dataset. This method shows high accuracy in detecting intrusions with a more discriminative feature subset.

Ali et al. (2018) proposed a PSO-FLN for intrusion detection problem. KDD99 benchmark dataset was used for validation. This model was compared with algorithms, i.e. ELM, and FLN classifier. This technique provides high testing accuracy, which can be further increased by increasing the number of hidden neurons in the ANN.

After reviewing ACO and PSO methods, it can be concluded that discrete and nonlinear optimization problems can be easily solved with SI methods. Researchers used these SI methods for the generation of classification rules or to discover clusters for Anomaly Detection. Some researchers used hybrid approaches for the enhancement of intrusion detection. These approaches showed better results than traditional or single approaches. Due to self-organization and division of labour like properties, challenging problems can be decomposed into smaller ones and handed over to an agent to work in parallel. So by adopting SI methods, real-life problems can be easily solved. The comparative study of ML methods is shown in Table 7.

5.12 Challenges of ML methods and its remedies

It can be concluded from the literature mentioned above that ML methods have been widely used to detect various types of attacks. It helps the network administrator to take the counter steps to deal with attacks. However, most conventional ML methods belong to shallow learning (SL) and often focus on feature engineering and selection. The learning capacity of traditional detection approaches is limited, and learning efficiency further decreases as the network structure complicated. They only represent partial information, i.e. one or two levels of information and cannot effectively solve the real network application problem. The multi-classification task will lead to decreased accuracy due to the dynamic growth of data sets.

Shallow learning methods require a vast quantity of training data for the operation, which become a challenge in a heterogeneous environment. Besides, shallow learning is expensive and labour intensive and not suited for forecasting high-dimensional learning requirements with massive data. When dealing with a large number of multi-type variables, logistic regression is easy to underfit, and the accuracy is low; decision trees are prone to overfitting and neglect the problems caused by inter-data correlation; SVM is inefficient when dealing with large samples, and it can be challenging to find a suitable kernel function that can deal with missing data.

To address these limitations, DL methods, an advanced subset of ML, are receiving interest across multiple domains. It has attracted researchers due to its several advantages over ML methods like automatic feature learning, flexible adaptation to novel problems which make it possible to work upon big data etc. Its superior layer feature learning ability can show improved or at least the same ML methods performance, as shown in Table 8.

6 Deep learning methods for IDSs

DL methods come into existence in 2006 and have become a prominent research topic. The word deep stands for many hidden layers in the neural network. It is a subcategory of ANN and has a more number of hidden layers than traditional neural networks, which goes up to 150. Although it is a branch of ML, complexity in the structure and learning data representations makes it a broader version of ML. DL deals with algorithms that learn from examples the same as in ML. The performance of ML and DL algorithm varies as the scale of the data increases. To find the network patterns, DL algorithms require massive data, whereas ML algorithms require lesser data. The structure can be made deep by adding one or more hidden layers in ANNs, and since the data is processed at each layer, thus, making the learning task deeper.

Table 7 Comparative study of ML methods for IDS (2016-2020)

Study	Dataset	Method used	Advantage(s)/result(s)	Limitation(s)/future scope
Mehmood and Rais (2016)	KDD-99	SVM, J.48, NB, DT	J48 outperforms other algorithms in terms of accuracy and misclassification rate	Feature selection methods can be used in future
Belavagi and Muniyal (2016)	NSL-KDD	SVM, LR, NB, RF	RF outperforms other algorithms in terms of highest TPR and lowest FPR	Multiclass classification is required in future
Aburomman and Reaz (2016)	KDD-99	PCA, LDA	Overall accuracy = 0.92162 FP = 0.0196 , FN = 0.10849	
Ashfaq et al. (2017)	NSL-KDD	Semi supervised learning (SSL) approach based on fuzziness	–	Limited only for binary classification tasks
Al-Yaseen et al. (2017)	KDD-99	SVM, EVM	95.75% accuracy,shorter training time	Efficient classifiers are required for novel attacks
Othman et al. (2018)	KDD-99	Chi-Square, SVM with SGD	High Performance, Low FPR	Can be extended to multi class model
Gautam and Doegar (2018)	KDD-99	Naive Bayes, Ensemble methods,Adaptive Boost and PART	Accuracy-99.97,Recall—99.98, Precision—99.99	Limit to 2 class attack
Hasan et al. (2019)	Kaggle	ANN, SVM, LR, DT, RF	DT, RF and ANN showed accuracy of 99.4% but in terms of other performance metrics RF outperforms other classifiers	More focus is needed on real time data
Saranya et al. (2020b)	KDD-99	LDA, CART, RF	RF outperforms other classifiers in terms of 99.65% accuracy	Realtime dataset can be used in future

The DL models are applied in the research of computer vision, audio recognition, natural language processing, speech recognition, face recognition, image recognition, information retrieval, failure prediction, handwriting recognition, feature learning , social network filtering, machine translation, dimensionality reduction, intrusion detection and so on. Table 17 shows the architecture and application areas of DL methods.

DL methods are categorized into supervised learning and unsupervised learning. Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) comes under the category of supervised learning, and Auto-Encoder(AE) and Deep Belief Network (DBN)comes under the category of unsupervised learning. There exist many other DL models as variants of these basic models (Fig. 10).

6.1 Supervised DL models

In supervised learning, the training of the machine is done with labelled data. After that, to analyze the training data, the machine is trained with a new set of examples and produce a correct result from labelled data. One popular network under supervised learning is CNN.

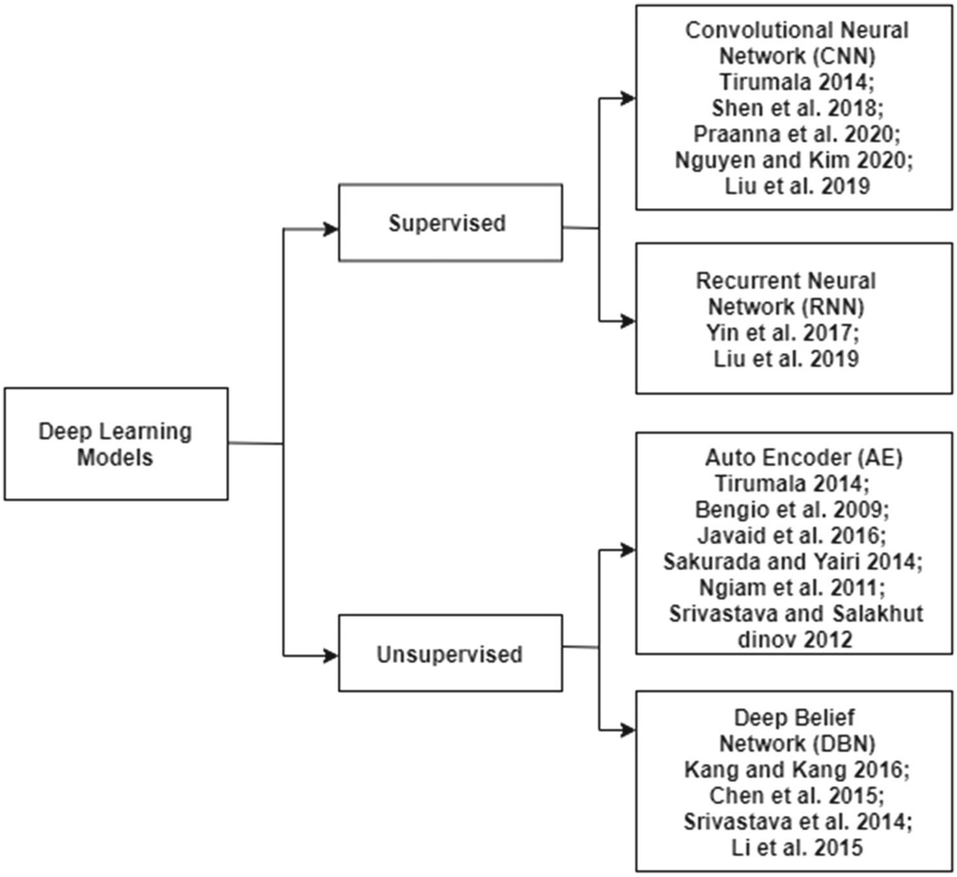
Table 8 Shallow ML versus DL

Sr. no.	Approach	Steps in learning	Result
1	Shallow ML	Input—Hand design Features—Mapping from Features	Output
2	DL	Input—Simple Features—Complex Features—Mapping From Features	Output

6.1.1 Convolutional neural network (CNN)

CNN are a particular form of feed-forward ANNs which works under supervised learning. These networks are made up of neurons with learn-able biases and weights. These models process data that comes in multiple arrays and eliminates the need for manual feature extraction. It works by withdrawing relevant features directly from images without retaining

Fig. 10 Deep learning models


Deep Learning Models
Supervised
Unsupervised
Convolutional Neural Network (CNN) Tirumala 2014; Shen et al. 2018; Praanna et al. 2020; Nguyen and Kim 2020; Liu et al. 2019
Recurrent Neural Network (RNN) Yin et al. 2017; Liu et al. 2019
Auto Encoder (AE) Tirumala 2014; Bengio et al. 2009; Javaid et al. 2016; Sakurada and Yairi 2014; Ngiam et al. 2011; Srivastava and Salakhut dinov 2012
Deep Belief Network (DBN) Kang and Kang 2016; Chen et al. 2015; Srivastava et al. 2014; Li et al. 2015

them. Feature identification is the main application of ConvNets. The automated feature extraction of ConvNets makes it highly accurate for computer tasks (Tirumala 2014). The architecture of CNN is shown in Fig. 11. The limitation of ConvNets is its limited ability to process natural data in its raw form.

Several researchers used CNN method for detecting intrusions. For instance, Shen et al. (2018) proposed a new compressed CNN model for image classification called CS-CNN that incorporates compressive sensing theory at the input layer of CNN models both minimize resource consumption and improve accuracy. MINST and CIFAR-10 datasets were used for the evaluation. This method improved the training speed and classification accuracy. Praanna et al. (2020) proposed a method that combines the CNN algorithm and the LSTM algorithm. The proposed method was evaluated with KDD99. According to the experiments' results, the proposed model outperformed SVM, CNN and DBN with 99.78% accuracy. Nguyen and Kim (2020) proposed a novel algorithm for a NIDS based on genetic algorithm (GA)-based exhaustive search and fuzzy C-means. The most successful CNN structure, called the deep feature extractor, was chosen using a GA-based optimization process. It was concluded from the results that deploying the proposed algorithm on real-world internet networks would boost computer network security by clustering criminal activities. The proposed algorithm performed better for multiple classifications. The time needed to implement a GA-based exhaustive search method to select a specific feature subset and an appropriate CNN structure was a limitation of this study.

CNN can accommodate image translation, rotation, size difference, and other types of deformations while providing accurate classification results. In a nutshell, it has good generalization potential when interacting with noisy inputs. The following factors contribute to the performance of CNN models in classification tasks:

1. The availability of comprehensive ground truth training sets with labels, e.g., ImageNet.
2. Implementations of high-speed GPU clusters for training a vast number of parameters.
3. Regularization techniques like dropout, which are carefully planned, increase generation capacity.

6.1.2 Recurrent neural network (RNN)

RNN is an extension of a conventional feed-forward network. It is a sequential learning model and is appropriate for sequential tasks like speech and language. It learns features from previous inputs' memory and has cyclic connections making

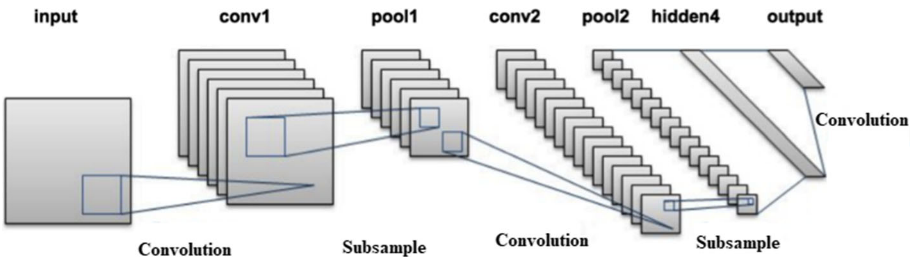


Fig. 11 Architecture of CNN

them robust for modelling sequences. RNNs are very good at predicting the next character in the text or the next word in a sequence, but they can also be used for more complex tasks. The RNN can capture arbitrary-length dependencies from a theoretical point of view which is difficult to handle and hard to train. However, it makes the gradient exploding or vanishing while training with Back Propagation Training Time algorithm. LSTM models are presented to prevent gradient exploding. RNN obtains the best performance in many applications such as speech recognition, natural language processing and machine translation.

Several researchers used RNN method for detecting intrusions. For instance, Yin et al. (2017) applied a DL-based RNN approach on NSL-KDD dataset to find various attacks in the network. After that, the results were compared with traditional classification methods like SVM, ANN proposed by previous researchers and found that RNN-IDS was very appropriate for modelling a classification model with high accuracy, and its performance was superior to ML classification methods in both binary and multi-class classification but to reduce the time, training time using GPU acceleration needs to be focused in future.

In another research work, Liu et al. (2019) suggested a payload classification approach to analyze payloads based on PL-CNN and PL-RNN use in attack detection. The proposed methods help end-to-end detection by learning feature representations from original payloads without requiring feature engineering. When applied to the DARPA1998 dataset, PL-CNN and PL-RNN techniques achieved accuracies of 99.36% and 99.98%, respectively. PL-RNN outperformed the PL-CNN on a variety of datasets. There were two issues with these models. First, unlike conventional ML models, these models had more parameters. Consequently, changing model parameters was complex, i.e., model training was difficult and required specific skills. Second, these methods were not well-interpreted.

6.2 Unsupervised DL models

In this learning, no teacher is available for guidance or training. Here the machine is trained using information that is neither labelled nor classified. The unsorted information is grouped by machine according to patterns, similarities and differences without any previous data training. Therefore, the machine can find the hidden structure in unlabeled data by self-learning. AE and DBN come under the category of unsupervised learning.

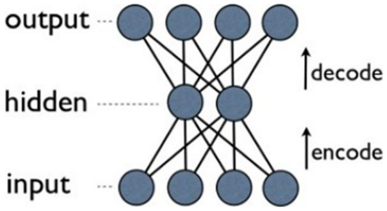


Fig. 12 Architecture of SAE

6.2.1 Auto encoder (AE)

In AE, its input is copied to its output. The reduction in dimensionality or feature learning is made by transforming high dimensional data into lower dimensional code. Data will be recovered from the code by a decoder network. Initially, random weights are assigned to both encoder and decoder networks. The training of AE is done by observing the difference between input and output obtained from encoding and decoding. Then the error is fed back to the decoder and encoder network, respectively (Tirumala 2014). This model's significant change is done by Bengio et al. (2009) changing the unsupervised training to supervised for identifying the significance of the training paradigm. The stacked autoencoders(SAE) with unsupervised training are more efficient than the SAE with supervised pre-training. The performance of SAE based on deep architecture is slightly less than the performance of RBMs based architecture because SAE is unable to ignore random noise in its training data. The architecture of SAE is shown in Fig. 12.

Several researchers used RNN method for detecting intrusions. For instance, Javaid et al. (2016) described a DL-based approach for developing a flexible and efficient NIDS on NSL-KDD benchmark datasets. In this paper, STL scheme based on unsupervised learning has been applied to training data using a sparse-auto encoder. The trained features were used on a labelled test dataset for classification into the normal and attack. N-fold cross-validation methods were used for performance evaluation, and the result obtained was reasonable. Accuracy, Precision, Recall, and F-measure values metrics were used for performance evaluation. The results were also compared with the soft-max regression (SMR) when applied directly to the dataset without feature learning. After evaluation, it was found that the performance of STL was better as compared to the previous work.

The autoencoders were also applied for anomaly detection (Sakurada and Yairi 2014), in which nonlinear feature reduction by autoencoders was used to train normal network profile. In their study, the authors analyzed the learned features in the hidden layer of AE. They found that AE learned the normal state properly and activated it differently with anomalous input.

Many historical evidence shows that SAE can perform better classification tasks and multiple levels of higher-quality representation in terms of feature learning than their shallow counterparts. But the limitation of SAE is difficulty in effectively performing feature learning on "Big data", having a large number of heterogeneous data due to the use of vectors to represent every hidden layer's input data and learning features. A vector cannot model the highly nonlinear distribution of the input data. To solve this problem, multi-modal DL models have been proposed by Ngiam et al. (2011), Srivastava and Salakhutdinov (2012). Firstly, feature learning from each modality is performed using conventional DL models and then integrate the learned features at different levels as shared representations of multi-model data. Multi-model DL models help capture the high-order correlations across multiple modalities to form the hierarchical representations of multi-modal data. However, they cannot model the nonlinear distribution of the heterogeneous input data since they learn features from different modal data independently, leading to the failure in learning useful features on big data.

Sakurada and Yairi (2014) proposed a tensor DL model for heterogeneous data. The stacking of multiple tensor autoencoder models was used to build the data computation model. This model achieved higher classification accuracy for heterogeneous data than multi-modal DL models.

6.2.2 Deep belief network (DBN)

DBN model was designed by Hinton et al. in 2006. It is based on MLP model with greedy layer-wise training and can learn feature representations from both the labelled and unlabeled data. It comprises many interconnected hidden layers in which each layer acts as an input to the next layer and is visible only to the next layer. Each layer in a DBN has no lateral connection between its nodes present in that layer. It first takes the benefit of an efficient layer by layer greedy learning strategy to initialize the deep network and then fine-tune all the weights jointly with the desired outputs. It optimizes its weights at time complexity linear to the depth and size of the networks. In this model, unsupervised pre-training and supervised fine-tuning strategies are used. Developing a DBN model is computationally expensive. The DBN architecture proposed by Hinton et al. is depicted in Fig. 13. Several researchers used DBN method for detecting intrusions. For instance, Kang and Kang (2016) used DBN for intrusion detection in-vehicular network. DBN based unsupervised pre-training models could improve intrusion detection accuracy, as demonstrated by various researchers. The limitation of this model is that its centralized approach might limit its practicality in fog networks.

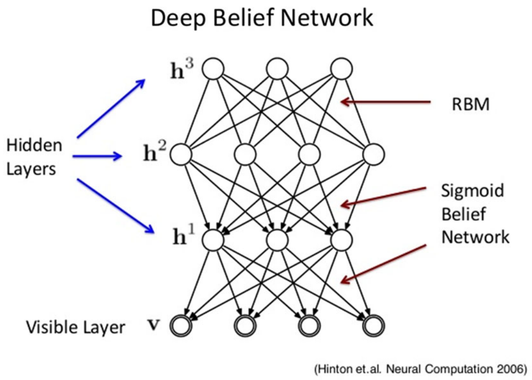


Fig. 13 Architecture of DBN

Boltzmann Machines (BM) is the form of log-linear Markov Random Field (MRF), where the energy function is linear in its free parameters. The hidden nodes can be introduced to make them robust enough. The modelling capacity of the BM can be increased by introducing more hidden variables. RBM is the most popular version of BM.

For regular RBM, the relationship between visible units and hidden units is limited to constants that certainly downgrade the representation capability of the RBM. To avoid this error and enhance DL capability, the fuzzy restricted Boltzmann machine (FRBM) and its learning algorithm are proposed by Chen et al. (2015). Here, the parameters governing the model are replaced by fuzzy numbers. As per results, the representation capacity of FRBM is better than traditional RBM, and when the noise-contaminated the training data, FBRM reveals better robustness property than RBM. To train the DNNs with many parameters creates an overfitting problem to solve the overfitting problem (Srivastava et al. 2014) introduced the dropout Restricted Boltzmann Machine model, which performs better than standard RBM. Dropout is a technique for dropping out units in a neural network. Dropping a unit out means temporarily removing it from the network.

Denoising AutoEncoder (DAE) is the process that uses similar input and output data. The denoising power is produced by adding noise to the training procedure. DAE are an essential and critical tool for feature selection and extraction. Variational AutoEncoders (VAEs) are a deep learning tool that can be used to learn latent representations. It is appropriate for sensor failure detection, application of IoT device security, and intrusion systems' security. The VAEs executes the visualization, recognition, representation, and denoising task.

Li et al. (2015), the malicious code detection was performed using AE for feature extraction and DBN as a classifier. KDDCUP'99 benchmark dataset was used for the experiment. The results have shown that the hybrid approach is more effective and accurate in time and detection accuracy than a single DBN. The advantage of these networks is that they are more beneficial than shallow ones in cyber-attack detection. The dataset required in this research should be the latest, which is its main drawback.

Table 9 summarizes DL methods for IDS (2015-2021).

6.3 Comparative analysis of experimental results for intrusion detection

A comparison of several ML and DL algorithms used for the IDS on benchmark datasets is presented in this section. The evaluation metrics used for the comparisons are accuracy, precision and recall. Table 10 shows the evaluation comparison of several ML classifiers using tenfold cross-validation on KDD99 dataset. Similarly, Table 11 shows the evaluation comparison of several ML classifiers using tenfold cross-validation on the UNSW-NB15 dataset.

We can see from Table 10 that HT, KNN, DT,and RF performed well in terms of classifying normal and abnormal traffic and achieved an accuracy of 99.22%, 99.83%, 99.86% and 99.94% respectively. The same order of superiority is retained (refereed to Table 11) by HT, KNN, DT, and RF using tenfold cross validation with accuracy of 93.53%, 93.71%, 95.54% and 96.07% respectively on UNSW-NB15 dataset.

Tables 12 and 13 present the evaluation comparison of several ML classifiers using tenfold cross validation on the supplied UNSW-NB15 and KDD99 dataset of the testing phase.

The empirical analysis of classifying spam traffic using supplied data sets from Tables 12 and 13 demonstrate that RF outperformed KNN, SMO and DT with a smaller margin. At the same time, its superiority is more significant as compared to other state of the art algorithms with a large margin. The accuracy obtained by SMO, KNN, DT and RF are 95.11%, 96.01%,96.22%,96.79%, respectively. The result analysis from Tables 9–12 shows that RF classifier gives better performance in most cases because while increasing the trees, the RF adds more randomness to the model. When dividing a node, it looks for the best feature among a random subset of features rather than the most significant feature. As a result, there is a lot of variety, which leads to a better model.

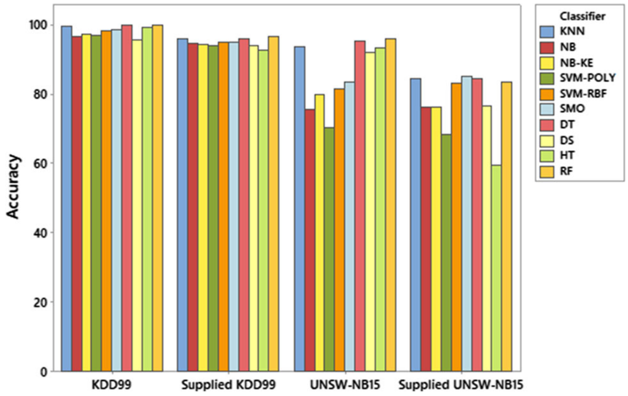


Fig. 14 Bar chart comparison for state of the art algorithms in terms of accuracy

Time complexity of all classifiers used for the testing and training on UNSW-NB15 and KDD99 datasets are given in Table 14. It is observed from the results that RF classifier requires more time for training because it uses many decision trees to define the class.

The accuracy comparison for all data set using ML state of the art algorithms can be visualized in Fig. 14. From, Fig. 14 it is verified that RF, DT, KNN and HT gives high results in the tenfold cross-validation test mode of both datasets compared to the other classifiers, while the RF, DT, KNN and SMO achieve high results in the supplied test mode of both datasets compared to the other classifiers.

We carried out a comparison of several DL models used for the IDS based on cybersecurity. The training time and accuracy of DL supervised and unsupervised models with various hidden nodes and learning rate using the CSE-CIC-2018 dataset is presented in Table 15. The presented results are directly taken from Ferrag et al. (2020). Similarly, the training time and accuracy of DL supervised and unsupervised models with various hidden nodes and learning rate using the Bot-IoT dataset is manifested in Table 16.

Contrasted with both profound neural network and RNN, the CNN gets a higher precision of 97.38% (referred to Table 15), in the presence of 100 hidden nodes and the learning rate is 0.5. Furthermore, Table 15 demonstrates the precision and preparation time of generative/solo models in the CSE-CIC-IDS2018 dataset with various hidden nodes variants and learning rates. The profound DA get a higher precision of 97.37% when there are 100 hidden nodes, and the learning rate is 0.5 contrasted with the other three algorithms, i.e. DBM, DBN, and RBM.

Table 16 presents the exactness and preparation time of profound discriminative models in the Bot-IoT dataset with multiple hidden nodes and learning rates. The CNN receives a higher exactness 98.37% in 100 hidden nodes and a 0.5 learning rate. CNN increases the system performance and accuracy due to its unique features like shared weights and

Table 9 Summarized review of DL methods for NIDS (2015-2021)

Study	Dataset	Method used	Advantage(s)/result(s)	Limitation(s)/future scope
Zhao et al. (2015)	(Three real world datasets) Animal-10 NUS-WIDE-Object MSRA-MM	multi-modal deep neural networks feature selection with sparse group LASSO	Efficient in selecting the relevant features and attains competitive classification performance	Framework applied on single-label multi-class classification problem, it might be further extended to multi-label categorization or retrieval tasks
Eesa et al. (2015)	KDD-99	Cuttlefish optimization algorithm	Higher Detection Rate, Higher accuracy, lower false alarm rate	CFA can be used as a rule generator for future work
Srimuang and Intara-sothonchun (2015)	KDD-99	Weighted-ELM	Takes less time for working	R2L and Probing attacks had lower effectiveness
Li et al. (2015)	KDDCUP'99	Hybrid (AE, DBN)	Improves malicious code detection accuracy, reduces time complexity	
Zhang et al. (2015a)	STL-10 and NUS-WIDE	Deep computation model uses the BGV encryption scheme to encrypt the private data	Efficiently deal with deep computation model for big data feature learning	Additional overhead to perform the data encryption/decryption and communication between the client and the cloud, little lower performance accuracy. For future work focus should be given to incremental deep computation model
Zhang et al. (2015b)	STL-10, CUAVE, SANE and INEX datasets	Tensor DL model	The model is successful to perform feature learning for heterogeneous data due to its capability of learning abstract representations of multiple modal data	Takes more times to train the parameters than SAE and multimodels DL. The focus is given to improve the efficiency of deep computational model
Tang et al. (2016)	NSL-KDD	(DNN) Deep Neural Network in SDN Environment	75.75% accuracy rate, AOC = .86	Implement this approach in real SDN environment with real traffic is still needed to improve the performance
Dong and Wang (2016)	KDD-99	Deep coding/SVM-RBM, SVM, Decision Tree, Naïve Bayes C4.5	SVM—RBM gets the better precision, gives accurate information on the anomalous behavior	
Ma et al. (2016)	KDD-99 NSL-KDD	(SC-DNN), spectral clustering, deep neural network	Suitable in complex networks, improves detection accuracy in real security system, more capable of classifying sparse attacks cases	Cluster parameters are to be determined empirically and not through mathematical theory
Aminanto and Kim (2016a)	KDD-99	ANN (used for Feature Selection) SAE (used for Classifier)	Detection Rate 99.4% = IDS-T 99.9% = IDS-All IDS-T Training time = I Minute IDS-All Training Time = 10 min	Difficult to implement in Wireless System

Aminanto and Kim (2016b)	AWID	ANN (used for Feature Selection)
		SAE (used for Classifier)
			Detection Rate = 65.18%
			FAR = 0.14%
			Accuracy = 98.59%
			Precision = 94.53%
			F1 = 77.16%
				Limited to impersonation attacks only

Table 9 continued

Study	Dataset	Method used	Advantage(s)/result(s)	Limitation(s)/future scope
Nskh et al. (2016)	KDD-CUP 1999	SVM, PCA	RBF kernel exhibits better results with better DR and detection speed is faster in polynomial kernel based SVM	

Javaid et al. (2016)	NSL-KDD	Self Taught Learning (STL) association with (Sparce auto encoder, Soft-max)	*2 Class*
			Precision = 85.44%
			Recall = 95.95%
			F-Measure = 90.4%
			Accuracy = 88.39%
			5 Class
			F-Measure = 75.76%
			Accuracy = 79.10%
			All Class
			Accuracy = 98%
				Implementation of real time NIDS with more efficient and active feature learning is required

Seo et al. (2016)	KDD 99	Restricted Boltzmann Machine	Accuracy = 99.4%
			Precision = 99.8%
				Parameters like batch size, learning rate and no. of iteration can be revised

Aminanto and Kim (2017)	AWID	SAE is used as a Feature Extraction and selection Method	Detection Rate = 92.18 %
			FAR = 4.40%
			Accuracy = 94.81%
			Precision = 86.15%
			F1 = 89.06%
				In future SAE will be used as outlier detection for detecting unknown attacks

Effendy et al. (2017)	NSL-KDD	k-means clustering, Information gain	Provides high accuracy value	
Wisesty et al. (2017)	KDD-CUP 1999	Conjugate Gradient algorithm	93.2% accuracy in two class classification, 54.13% in case of multi class classification	In future, sampling method can be used to enhance the performance of classification system
Yin et al. (2017)	NSL-KDD	Proposed RNN-IDS and compared with ANN, J48, placePlaceNameRandom PlaceTypeForest, SVM	High accuracy , High DR, low FPR	In future attention will be given to reduce the training time using GPU acceleration
Hodo et al. (2017)	UNB-CIC	CFS-ANN based classifier, ANN SVM	Detects nonTor traffic with an accuracy of 99.8%, DR 100% and FPR of 1.2%	In future, performance will be analyzed in classification of 8 different types of traffic in UNB-CIC Tor Network Traffic dataset
Zhao et al. (2017)	KDD Cup 99	PCA Model k-NN Softmax Regression	Softmax regression shown better time performance	Calculation of memory size is ignored
Manzoor et al. (2017)	KDD-99	ANN	Increased DR, reduced FAR	Manually preprocessing work, number of features in reduced feature set can be made optimal
Aminanto et al. (2017)	Aegean Wi-Fi Intrusion Dataset (AWID)	D-FES (Deep Feature extraction and Selection) Used as an Clustering	Detection accuracy = 99.918% FAR = 0.012%	Extend D-FES to all attack classes
Liu et al. (2019)	DARPA-98	PL-CNN PL-RNN	Higher DR	Model training is difficult due to more parameters and interpretation of models is difficult
He et al. (2018)	KDD CUP 1999	Kernel clustering algorithm	Higher DR, Lower FAR, fit for most attack types	

Table 9 continued

Study	Dataset	Method used	Advantage(s)/result(s)	Limitation(s)/future scope
Napiah et al. (2018)	Simulated Dataset	CHA-IDS Multi-Agent System SVM, J48,MLP Naïve Bayes, Logistic, Random Forest Feature Selection: BPS-CFS, GS-CFS	J48 algorithm shows 99% TPR, consumed low energy overhead and memory, high capability in detecting routing attacks	Unable to precisely identify the attacker
Ali et al. (2018)	KDD-99	(PSO-FLN) Model Particle Swarm Optimization Fast Learning Network Model and Outperformed ELM and FNN Classifier in the testing accuracy	R2L has obtained less accuracy due to limited samples	
Muna et al. (2018)	NSL-KDD and UNSW-NB15	Deep-Auto encoder, Deep Feed Forward Neural Network	99% Detection Rate, 1.8% False Alarm Rate	Need to train the algorithm on real data collected from IoT systems

Kim et al. (2016)	CSIC-2010 HTTP	LSTM-RNN with Adam Optimizer	Accuracy = 99.97%
			Recall = .995%
			Precision = .995%
				To Evaluate LSTM Performance with different optimizers

Shone et al. (2018)	KDD-99, NSL-KDD	Nonsymmetric deep autoencoder (NDAE), RF Classification Algorithm, DL, Stacked NDAEs	Accuracy = 89.22%
			Precision = 92.97%
			Recall = 89.22%
			F-Score = 90.76%
			False Alarms = 10.78%
				Not perfect to handle zero-day attacks

Caminero et al. (2019)	NSL-KDD, AWID	AE-RL Adversarial environment reinforcement learning	Accuracy = 80.16%
			Precision = 79.74%
			Recall = 80.16%
			F-Score = 79.40%	

Yang et al. (2019)	NSL-KDD, UNSW-NB15	Modified density peak clustering algorithm (MDPCA) and deep belief networks (DBNs)	Accuracy = 82.08%
			DR = 70.51%
			FPR = 2.62%
				To synthesize R2L and U2R attacks for increasing the performance of the Model

Feng et al. (2019)	KDD 99	DNN , CNN and LSTM	Accuracy = 98.5%
			Precision = 97.63%
			Recall = 99.59%
			F-Score = 98.6%
				Limited to DoS ,XSS and SQL attacks

Gamage and Samarabandu (2020)	KDD 99, NSL-KDD, CIC-IDS2017,CIC-IDS2018	AE , DBN and LSTM	Empirical results with difference of 2.5 to 3% in comparison to reference papers	–
Zhang et al. (2020)	KDD 99	AN-LSTM	Improved accuracy and detection performance	More processing time
Sohi et al. (2021)	Publicly available datasets	RNNIDS	Improvement in the detection rate upto 16.67%	To minimize the false positives in the future

Table 10 Comparative analysis using KDD99 dataset (Khan and Gumaei 2019)

Classifier	SMO	DT	DS	HT	RF
Accuracy %	98.8289	99.8661	95.7757	99.2293	99.9437
precision	0.988	0.999	0.960	0.992	0.999
recall	0.988	0.999	0.958	0.992	0.999
Classifier	KNN	NB	NB-KE	SVM-POLY	SVM-RBF
Accuracy %	99.8393	96.589	97.337	97.214	98.4367
precision	0.998	0.966	0.974	0.973	0.984
recall	0.998	0.966	0.973	0.972	0.984

Table 11 Evaluation comparison of ML classifiers using tenfold cross validation on the UNSW-NB15 dataset (Khan and Gumaei 2019)

Classifier	SMO	DT	DS	HT	RF
Accuracy %	83.588	95.5413	92.0629	93.5349	96.0791
precision	0.837	0.955	0.928	0.935	0.961
recall	0.836	0.955	0.921	0.935	0.961
Classifier	KNN	NB	NB-KE	SVM-POLY	SVM-RBF
Accuracy %	93.7134	75.749	79.9157	70.44	81.708
precision	0.937	0.831	0.848	0.707	0.817
recall	0.937	0.757	0.799	0.704	0.817

Table 12 Evaluation comparison of ML classifiers using tenfold cross validation on the supplied KDD99 dataset of the testing phase (Khan and Gumaei 2019)

Classifier	SMO	DT	DS	HT	RF
Accuracy %	95.1125	96.218	93.9811	92.6586	96.7926
precision	0.952	0.962	0.944	0.926	0.969
recall	0.951	0.962	0.940	0.927	0.968
Classifier	KNN	NB	NB-KE	SVM-POLY	SVM-RBF
Accuracy %	96.0065	94.6799	94.4287	94.0411	94.9474
precision	0.962	0.947	0.948	0.943	0.951
recall	0.960	0.947	0.944	0.940	0.949

Table 13 Evaluation comparison of ML classifiers using tenfold cross validation on the supplied UNSW-NB15 dataset of the testing phase (Khan and Gumaei 2019)

Classifier	SMO	DT	DS	HT	RF
Accuracy %	85.3411	84.554	76.6324	59.4423	83.6333
precision	0.863	0.864	0.835	0.763	0.869
recall	0.853	0.846	0.766	0.594	0.836
Classifier	KNN	NB	NB-KE	SVM-POLY	SVM-RBF
Accuracy %	84.4872	76.3907	76.2219	68.3379	83.2216
precision	0.855	0.782	0.768	0.689	0.835
recall	0.845	0.764	0.762	0.683	0.832

Table 14 Time complexity comparison (in seconds) for training phase on KDD99 and UNSW-NB15 dataset (Khan and Gumaei 2019)

KNN	NB	NB-KE	SVM-Poly	SVM-RBF	SMO	DT	DS	HT	RF
KDD99 dataset									
0.06	1.03	1.01	228.88	198.69	789.77	40.78	2.15	5.50	128.51
UNSW-NB15 dataset									
0.18	1.84	3.06	793.48	748.36	531.11	76.13	3.95	8.27	542.97

Table 15 Training time and accuracy of DL supervised and unsupervised models with various hidden nodes and learning rate using the CSE-CIC-2018 dataset (Ferrag et al. 2020)

Parameters	Metric	CNN	RNN	DNN	DA	DBM	DBN	RBM
LR = 0.5	Time	331.2	334.7	390.2	341.3	351.5	344.7	390.1
HN = 100	ACC	97.38%	97.31%	97.28%	97.37%	97.37%	97.30%	97.28%
LR = 0.1	Time	332.5	336.9	391.1	331.7	330.1	334.8	390
HN = 100	ACC	97.31%	97.23%	97.19%	97.31%	97.30%	97.23%	97.19%
LR = 0.01	Time	338.9	341.5	395.2	337.11	339.1	340.4	394.1
HN = 100	ACC	97.22%	97.11%	97.10%	97.22%	97.21%	97.11%	97.10%
LR = 0.5	Time	182.6	190.6	177.7	181.4	181.4	190.5	177.6
HN = 60	ACC	96.99%	96.96%	96.95%	96.99%	96.99%	96.96%	96.95%
LR = 0.1	Time	189.1	192.2	179.3	189.1	189	192.1	179.1
HN = 60	ACC	96.98%	96.97%	96.92%	96.97%	96.97%	96.97%	96.92%
LR = 0.01	Time	192.2	197.5	180.2	191.4	191.1	196.5	180.1
HN = 60	ACC	96.92%	96.90%	96.70%	96.91%	96.91%	96.88%	96.69%
LR = 0.5	Time	87.9	90.3	86.1	87.1	87.9	90.3	86.1
HN = 30	ACC	96.93%	96.89%	96.66%	96.92%	96.93%	96.89%	96.66%
LR = 0.1	Time	88.5	90.9	87.9	88.2	88.3	90.7	87.4
HN = 30	ACC	96.93%	96.89%	96.66%	96.92%	96.92%	96.88%	96.66%
LR = 0.01	Time	89.6	91.3	88.1	88.6	89.5	90.4	88
HN = 30	ACC	96.92%	96.88%	96.61%	96.92%	96.92%	96.84%	96.60%
LR = 0.5	Time	27.1	29.1	18.9	27.1	26.2	28.1	18.8
HN = 15	ACC	96.91%	96.89%	96.65%	96.91%	96.91%	96.89%	96.65%
LR = 0.1	Time	27.2	29.2	19.1	27.2	27.1	29.1	19
HN = 15	ACC	96.91%	96.88%	96.65%	96.90%	96.90%	96.87%	96.64%
LR = 0.01	Time	28.4	30.3	20.2	28.3	28.3	30.1	20
HN = 15	ACC	96.92%	96.87%	96.55%	96.91%	96.91%	96.85%	96.55%

local connectivity. Besides, the preparation time of profound neural networks is in every case, not precisely other related strategies (such as CNN and RNN).
Moreover, Table 16 confirms the exactness and training time of generative/unaided models in the Bot-IoT dataset with different hidden nodes and learning rate. The profound DA gets a higher precision of 98.39% compared to other states of the art algorithms. Interval and Box plots expressing the overall deviation of accuracy for Bot-IoT and CSE-CIC-2018 data set is presented in Figs. 15 and 16 respectively.

After summarizing review of DL methods, several challenges related to DL methods have been identified which needs to be resolved (Table 17).

7 Challenges

DL is a powerful tool for intrusion detection. But it also has its fair share of challenges that need to be addressed. One of the challenges in DL is to maintain accuracy while compressing large scale DL models. Although DL models are focused on incomplete or noisy data, feature learning, reliable DL models are required by many outdated objects in "Big data" to explore low-quality data on priority.

Another challenge in DL is to implement self-learning. Day by day, new attack scenarios are evolving. Therefore, features identified to detect one category of attacks might soon become outdated/insufficient for the others. Thus, there is a need to develop a framework that can automatically learn features, reduce computational time and increase accuracy.

Generalization is a critical challenge in DL systems. It is not possible to give a labelled sample of every problem to a DL algorithm. Therefore, it will have to be first generalized with its previous samples to classify new data. Currently, DL lacks a mechanism for learning abstractions through verbal definitions. It performs well only if billions of training examples are available.

Another challenge of the DL system is that it is not aware of how a neural network arrives at a solution/conclusion. Even neural network produces good results, but it is hard to predict if a failure occurs due to lack of transparency in their thinking process. It is not suitable for those domains where verification of the process is necessary, like medicine.

Overfitting the model is another challenge of DL. It refers to an algorithm that models the training data too well. It means an algorithm learns training data to the extent that it negatively affects the model's performance. When the accuracy stops improving over a certain number of

Table 16 Training time and accuracy of DL supervised and unsupervised models with various hidden nodes and learning rate using the Bot-IoT dataset (Ferrag et al. 2020)

Parameters	Metric	CNN	RNN	DNN	DA	DBM	DBN	RBM
LR = 0.5	Time	1367.2	1400.6	991.6	2816.2	2800.1	2921.7	2111.9
HN = 100	ACC	98.37%	98.31%	98.22%	98.39%	98.38%	98.31%	98.28%
LR = 0.1	Time	1022.1	1001.8	711.9	2566.9	2531.2	2644.2	1991.6
HN = 100	ACC	98.12%	97.99%	97.50%	98.31%	98.37%	98.12%	98.21%
LR = 0.01	Time	812.2	801.5	600.2	2466.2	2401.1	2521.8	1861.7
HN = 100	ACC	97.99%	97.62%	97.22%	98.32%	98.31%	98.11%	98.20%
LR = 0.5	Time	412.2	451.2	391.1	2101.8	2109.8	2201.9	1771.9
HN = 60	ACC	97.88%	97.29%	97.10%	98.00%	98.00%	97.98%	97.72%
LR = 0.1	Time	366.2	377.1	302.9	1821.1	1811.9	1912.8	1421.1
HN = 60	ACC	97.21%	96.97%	96.92%	98.00%	97.97%	97.96%	97.22%
LR = 0.01	Time	339.6	331.2	250.8	1461.2	1432.6	1461.6	1129.6
HN = 60	ACC	97.10%	96.96%	96.77%	97.93%	97.92%	97.18%	96.87%
LR = 0.5	Time	221.7	222.1	170.3	1266.8	1239.6	1291.6	1022.6
HN = 30	ACC	97.10%	96.90%	96.66%	97.92%	97.93%	96.99%	96.76%
LR = 0.1	Time	144.2	150.4	102.2	791.6	788.1	801.1	701.6
HN = 30	ACC	96.92%	96.88%	96.66%	97.92%	97.91%	96.92%	96.76%
LR = 0.01	Time	101.1	102.5	88.1	524.2	522.1	560.2	400.8
HN = 30	ACC	96.92%	96.88%	96.61%	96.96%	96.94%	96.86%	96.62%
LR = 0.5	Time	101.1	102.5	88.1	210.3	201.9	221.7	150.5
HN = 15	ACC	96.91%	96.88%	96.65%	96.96%	96.91%	96.89%	96.66%
LR = 0.1	Time	91.3	92.6	66.6	133.7	133.1	138.2	100.2
HN = 15	ACC	96.91%	96.88%	96.65%	96.93%	96.92%	96.88%	96.67%
LR = 0.01	Time	65.3	70.7	56.5	60.1	60.2	72.8	50.4
HN = 15	ACC	96.90%	96.77%	96.45%	96.72%	96.41%	96.55%	96.65%

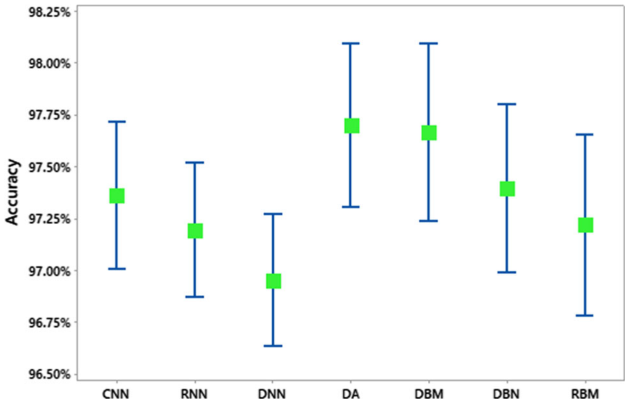


Fig. 15 Interval plot comparison showing over all deviation of accuracy for Bot-IoT dataset

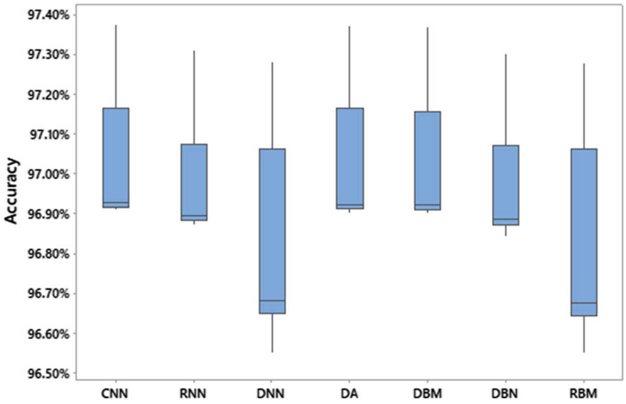


Fig. 16 Box plot comparison showing over all deviation of accuracy for CSE-CIC-2018 dataset

epochs, we can say the model is over trained or overfitted.

To solve the real-world problems, DL models require the machines to be equipped with sufficient processing power like GPUs. These processing units consume a lot of power and are costly. Therefore, it is not feasible for small industries to train the data with GPUs. To reduce the training time of GPU acceleration is another challenge. Although a lot of research is going on DL models, they still fail to handle zero-day attacks.

Table 17 Architecture and application areas of DL methods

Architecture	Application area
CNN	Natural language processing, Image recognition, Face recognition, Document analysis
AE	Natural language processing, Compact representation of data
RNN	Speech and Handwriting recognition
LSTM	Natural language text captioning,Speech and Handwriting recognition, Image captioning
DBN	Image recognition, Natural language understanding
RBN	Feature learning, dimensionality reduction, classification

8 Summary

DL methods have been applied to several fields for solving complex problems, including intrusion detection. These methods addressed many issues of shallow ML methods like improving the accuracy of detecting intrusions. This paper presented a systematic review of ML and DL methods for IDSs. To that end, we introduced IDS and provided its classification. We presented a review of datasets and performance metrics used for evaluating IDS' performance. This paper introduces the main ML methods and their applications for detecting intrusions, followed by pros and cons. The paper also provided DL methods and recent advancements for IDSs. Finally, we listed the challenges of ML and DL methods for IDSs and provided clues for future research in this field.

Declaration

Conflict of Interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

Abdullah B, Abd-Alghafar I, Salama GI, Abd-Alhafez A (2009) Performance evaluation of a genetic algorithm based approach to network intrusion detection system. In: International conference on aerospace sciences and aviation technology. The Military Technical College, pp 1–17

Abubakar AI, Chiroma H, Muaz SA, Ila LB (2015) A review of the advances in cyber security benchmark datasets for evaluating data-driven based intrusion detection systems. In: SCSE, pp 221–227

Aburomman AA, Reaz MBI (2016) Ensemble of binary SVM classifiers based on pca and lda feature extraction for intrusion detection. In: 2016 IEEE advanced information management, communicates, electronic and automation control conference (IMCEC). IEEE, pp 636–640

Aghdam MH, Kabiri P (2016) Feature selection for intrusion detection system using ant colony optimization. IJ Netw Secur 18(3):420–432

Aissa NB, Guerroumi M (2016) Semi-supervised statistical approach for network anomaly detection. Procedia Comput Sci 83:1090–1095

Akilandeswari V, Shalinie SM (2012) Probabilistic neural network based attack traffic classification. In: 2012 Fourth international conference on advanced computing (ICoAC). IEEE, pp 1–8

Al-Dhafian B, Ahmad I, Al-Ghamid A (2015) An overview of the current classification techniques in intrusion detection. In: Proceedings of the international conference on security and management (SAM). The Steering Committee of The World Congress in Computer Science, Computer, p 82

Al-Yaseen WL, Othman ZA, Nazri MZA (2017) Multi-level hybrid support vector machine and extreme learning machine based on modified k-means for intrusion detection system. Expert Syst Appl 67:296–303

Ali MH, Al Mohammed BAD, Ismail A, Zolkipli MF (2018) A new intrusion detection system based on fast learning network and particle swarm optimization. IEEE Access 6:20255–20261

Aljumah A (2017) Detection of distributed denial of service attacks using artificial neural networks. In: IJACSA international journal of advanced computer science and applications, vol 8(8)

Almomani A, Alauthman M, Albalas F, Dorgham O, Obeidat A (2020) An online intrusion detection system to cloud computing based on neucube algorithms. In: Cognitive analytics: concepts, methodologies, tools, and applications. IGI Global, pp 1042–1059

Alsubhi K, Aib I, Boutaba R (2012) Fuzmet: a fuzzy-logic based alert prioritization engine for intrusion detection systems. Int J Netw Manag 22(4):263–284

Alzaylaee MK, Yerima SY, Sezer S (2020) Dl-droid: deep learning based android malware detection using real devices. Comput Secur 89:101663

Aminanto ME, Kim K (2016a) Deep learning-based feature selection for intrusion detection system in transport layer. In: Proceedings of the Korea Institutes of information security and cryptology conference, pp 740–743

Aminanto ME, Kim K (2016b) Detecting impersonation attack in wifi networks using deep learning approach. In: International workshop on information security applications. Springer, pp 136–147

Aminanto ME, Kim K (2017) Improving detection of wi-fi impersonation by fully unsupervised deep learning. In: International workshop on information security applications. Springer, pp 212–223

Aminanto ME, Choi R, Tanuwidjaja HC, Yoo PD, Kim K (2017) Deep abstraction and weighted feature selection for wi-fi impersonation detection. IEEE Trans Inf Forensics Secur 13(3):621–636

Arabo A (2019) Distributed ids using agents: an agent-based detection system to detect passive and active threats to a network. In: ICCWS 2019 14th international conference on cyber warfare and security: ICCWS 2019. Academic Conferences and Publishing Limited, p 11

Arrieta AB, Díaz-Rodríguez N, Del Ser J, Bennetot A, Tabik S, Barbado A, García S, Gil-López S, Molina D, Benjamins R et al (2020) Explainable artificial intelligence (xai): concepts, taxonomies, opportunities and challenges toward responsible ai. Inf Fusion 58:82–115

Ashfaq RAR, Wang XZ, Huang JZ, Abbas H, He YL (2017) Fuzziness based semi-supervised learning approach for intrusion detection system. Inf Sci 378:484–497

Bamakan SMH, Amiri B, Mirzabagheri M, Shi Y (2015) A new intrusion detection approach using pso based multiple criteria linear programming. Procedia Comput Sci 55:231–237

Bamakan SMH, Wang H, Yingjie T, Shi Y (2016) An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization. Neurocomputing 199:90–102

Belavagi MC, Muniyal B (2016) Performance evaluation of supervised machine learning algorithms for intrusion detection. Procedia Comput Sci 89:117–123

Bengio Y et al (2009) Learning deep architectures for ai. Found Trends Mach Learn 2(1):1–127

Beni G, Wang J (1993) Swarm intelligence in cellular robotic systems. In: Robots and biological systems: towards a new bionics? Springer, pp 703–712

Brugger ST, Chow J (2007) An assessment of the Darpa ids evaluation dataset using snort. UCDAVIS Dept Comput Sci 1(2007):22

Caminero G, Lopez-Martin M, Carro B (2019) Adversarial environment reinforcement learning algorithm for intrusion detection. Comput Netw 159:96–109

Çavuşoğlu Ü (2019) A new hybrid approach for intrusion detection using machine learning methods. Appl Intell 49(7):2735–2761

Chaabouni N, Mosbah M, Zemmari A, Sauvignac C, Faruki P (2019) Network intrusion detection for iot security based on learning techniques. IEEE Commun Surv Tutor 21(3):2671–2701

Chen H, Schuffels C, Orwig R (1996) Internet categorization and search: a self-organizing approach. J Vis Commun Image Represent 7(1):88–102

Chen CP, Zhang CY, Chen L, Gan M (2015) Fuzzy restricted Boltzmann machine for the enhancement of deep learning. IEEE Trans Fuzzy Syst 23(6):2163–2173

Chevalier R, Plaquin D, Villatel M, Hiet G (2020) Intrusion detection systems. US Patent App. 16/486,331

Chitrakar R, Huang C (2012) Anomaly based intrusion detection using hybrid learning approach of combining k-medoids clustering and Naive Bayes classification. In: 2012 8th International conference on wireless communications, networking and mobile computing. IEEE, pp 1–5

Cloudstor (2019) Cloudstor. https://cloudstor.aarnet.edu.au/plus/index.php/s/2DhnLGDdEECo4ys. Accessed 15 Apr 15 2020

Creech G, Hu J (2013) Generation of a new ids test dataset: time to retire the kdd collection. In: 2013 IEEE wireless communications and networking conference (WCNC). IEEE, pp 4487–4492

Crosbie M, Spafford G, et al. (1995) Applying genetic programming to intrusion detection. In: Working notes for the AAAI symposium on genetic programming. MIT Press, Cambridge, pp 1–8

Dong B, Wang X (2016) Comparison deep learning method to traditional methods using for network intrusion detection. In: 2016 8th IEEE international conference on communication software and networks (ICCSN). IEEE, pp 581–585

Eesa AS, Orman Z, Brifcani AMA (2015) A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems. Expert Syst Appl 42(5):2670–2679

Effendy DA, Kusrini K, Sudarmawan S (2017) Classification of intrusion detection system (ids) based on computer network. In: 2017 2nd International conferences on information technology, information systems and electrical engineering (ICITISEE). IEEE, pp 90–94

Farnaaz N, Jabbar M (2016) Random forest modeling for network intrusion detection system. Procedia Comput Sci 89:213–217

Farzaneh B, Montazeri MA, Jamali S (2019) An anomaly-based ids for detecting attacks in rpl-based internet of things. In: 2019 5th International conference on web research (ICWR). IEEE, pp 61–66

Feng F, Liu X, Yong B, Zhou R, Zhou Q (2019) Anomaly detection in ad-hoc networks based on deep learning model: a plug and play device. Ad Hoc Netw 84:82–89

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

Gamage S, Samarabandu J (2020) Deep learning methods in network intrusion detection: a survey and an objective comparison. J Netw Comput Appl 169:102767

Gautam RKS, Doegar EA (2018) An ensemble approach for intrusion detection system using machine learning algorithms. In: 2018 8th International conference on cloud computing, data science & engineering (confluence). IEEE, pp 14–15

geopolitical-attacks (2019) geopolitical-attacks. https://www.privacyaffairs.com/geopolitical-attacks/3. Accessed 15 Apr 2020

Gharib A, Sharafaldin I, Lashkari AH, Ghorbani AA (2016) An evaluation framework for intrusion detection dataset. In: 2016 International conference on information science and security (ICISS). IEEE, pp 1–6

Ghose N, Lazos L, Rozenblit J, Breiger R (2019) Multimodal graph analysis of cyber attacks. In: 2019 Spring simulation conference (SpringSim). IEEE, pp 1–12

Gomez J, Dasgupta D (2002) Evolving fuzzy classifiers for intrusion detection. In: Proceedings of the 2002 IEEE workshop on information assurance, pp 321–323

Gu J, Lu S (2021) An effective intrusion detection approach using SVM with Naïve Bayes feature embedding. Comput Secur 103:102158

Gu J, Wang L, Wang H, Wang S (2019) A novel approach to intrusion detection using SVM ensemble with feature augmentation. Comput Secur 86:53–62

Guo C, Ping Y, Liu N, Luo SS (2016) A two-level hybrid approach for intrusion detection. Neurocomputing 214:391–400

Gupta BB, Joshi RC, Misra M (2012) Ann based scheme to predict number of zombies in a ddos attack. IJ Netw Secur 14(2):61–70

Hajimirzaei B, Navimipour NJ (2019) Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm. ICT Express 5(1):56–59

Hasan M, Islam MM, Zarif MII, Hashem M (2019) Attack and anomaly detection in iot sensors in iot sites using machine learning approaches. Internet Things 7:100059

He D, Chen X, Zou D, Pei L, Jiang L (2018) An improved kernel clustering algorithm used in computer network intrusion detection. In: 2018 IEEE international symposium on circuits and systems (ISCAS). IEEE, pp 1–5

Hodo E, Bellekens X, Iorkyase E, Hamilton A, Tachtatzis C, Atkinson R (2017) Machine learning approach for detection of nontor traffic. In: Proceedings of the 12th international conference on availability, reliability and security, pp 1–6

Hussain MS, Khan KUR (2020) A survey of ids techniques in manets using machine-learning. In: Proceedings of the third international conference on computational intelligence and informatics. Springer, pp 743–751

Ibrahim LM, Basheer DT, Mahmod MS (2013) A comparison study for intrusion database (kdd99, nsl-kdd) based on self organization map (som) artificial neural network. J Eng Sci Technol 8(1):107–119

Jacob NM, Wanjala MY (2018) A review of intrusion detection systems. Glob J Comput Sci Technol 5:66

Javaid A, Niyaz Q, Sun W, Alam M (2016) 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), pp 21–26

Jebur SA, Nasereddin H (2015) Enhanced solutions for misuse network intrusion detection system using sga and ssga. Int J Comput Sci Netw Secur 15(5):66

Kabir E, Hu J, Wang H, Zhuo G (2018) A novel statistical technique for intrusion detection systems. Fut Gener Comput Syst 79:303–318

Kalteh AM, Hjorth P, Berndtsson R (2008) Review of the self-organizing map (som) approach in water resources: analysis, modelling and application. Environ Model Softw 23(7):835–845

Kandan AM, Kathrine GJ, Melvin AR (2019) Network attacks and prevention techniques—a study. In: 2019 IEEE international conference on electrical, computer and communication technologies (ICECCT). IEEE, pp 1–6

Kang MJ, Kang JW (2016) Intrusion detection system using deep neural network for in-vehicle network security. PLoS ONE 11(6):6

Kevric J, Jukic S, Subasi A (2017) An effective combining classifier approach using tree algorithms for network intrusion detection. Neural Comput Appl 28(1):1051–1058

Khan FA, Gumaei A (2019) A comparative study of machine learning classifiers for network intrusion detection. In: International conference on artificial intelligence and security. Springer, pp 75–86

Kim G, Lee S, Kim S (2014) A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. Expert Syst Appl 41(4):1690–1700

Kim J, Kim J, Thu HLT, Kim H (2016) Long short term memory recurrent neural network classifier for intrusion detection. In: 2016 International conference on platform technology and service (PlatCon). IEEE, pp 1–5

Kudłacik P, Porwik P, Wesołowski T (2016) Fuzzy approach for intrusion detection based on users commands. Soft Comput 20(7):2705–2719

Kumar G, Kumar K, Sachdeva M (2010) The use of artificial intelligence based techniques for intrusion detection: a review. Artif Intell Rev 34(4):369–387

Kumar V, Chauhan H, Panwar D (2013) K-means clustering approach to analyze nsl-kdd intrusion detection dataset. Int J Soft Comput Eng 6:66

Kurniabudi K, Purnama B, Sharipuddin S, Darmawijoyo D, Stiawan D, Samsuryadi S, Heryanto A, Budiarto R (2019) Network anomaly detection research: a survey. Indones J Electr Eng Inform 7(1):37–50

Kyoto (2019) Kyoto. http://www.takakura.com/Kyoto_data/. Accessed 15 Apr 2020

Kyoto2006+ (2015) Kyoto2006+ dataset. http://www.takakura.com/Kyoto_data/. Accessed Feb 2015

Le Goues C, Nguyen T, Forrest S, Weimer W (2011) Genprog: a generic method for automatic software repair. IEEE Trans Softw Eng 38(1):54–72

Li W (2004) Using genetic algorithm for network intrusion detection. Proc US Dept Energy Cyber Secur Group 1:1–8

Li Y, Ma R, Jiao R (2015) A hybrid malicious code detection method based on deep learning. Int J Secur Appl 9(5):205–216

Liao HJ, Lin CHR, Lin YC, Tung KY (2013) Intrusion detection system: a comprehensive review. J Netw Comput Appl 36(1):16–24

Lin WC, Ke SW, Tsai CF (2015) Cann: an intrusion detection system based on combining cluster centers and nearest neighbors. Knowl Based Syst 78:13–21

Liu H, Lang B, Liu M, Yan H (2019) Cnn and rnn based payload classification methods for attack detection. Knowl Based Syst 163:332–341

Luo J (1999) Integrating fuzzy logic with data mining methods for intrusion detection. Master's thesis, Mississippi State University. Department of Computer Science

Ma T, Wang F, Cheng J, Yu Y, Chen X (2016) A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks. Sensors 16(10):1701

Mahoney MV, Chan PK (2003) An analysis of the 1999 darpa/lincoln laboratory evaluation data for network anomaly detection. In: International workshop on recent advances in intrusion detection. Springer, pp 220–237

Mäkelä A (2019) Network anomaly detection based on wavenet. In: Internet of things, smart spaces, and next generation networks and systems: 19th international conference, NEW2AN 2019, and 12th conference, ruSMART 2019, St. Petersburg, Russia, August 26–28, 2019, proceedings, vol 11660. Springer, p 424

Maniyar PS, Musande V (2016) Rules based intrusion detection system using genetic algorithm. Int J Comput Sci Netw 5(3):554–558

Manzoor I, Kumar N et al (2017) A feature reduced intrusion detection system using ANN classifier. Expert Syst Appl 88:249–257

Mehmood T, Rais HBM (2016) Machine learning algorithms in context of intrusion detection. In: 2016 3rd International conference on computer and information sciences (ICCOINS). IEEE, pp 369–373

Meng T, Jing X, Yan Z, Pedrycz W (2020) A survey on machine learning for data fusion. Inf Fusion 57:115–129

Mohamad Tahir H, Hasan W, Md Said A, Zakaria NH, Katuk N, Kabir NF, Omar MH, Ghazali O, Yahya NI (2015) Hybrid machine learning technique for intrusion detection system

Mousavi SM, Majidnezhad V, Naghipour A (2019) A new intelligent intrusion detector based on ensemble of decision trees. J Ambient Intell Hum Comput 66:1–13

Muna AH, Moustafa N, Sitnikova E (2018) Identification of malicious activities in industrial internet of things based on deep learning models. J Inf Secur Appl 41:1–11

Napiah MN, Idris MYIB, Ramli R, Ahmedy I (2018) Compression header analyzer intrusion detection system (cha-ids) for 6lowpan communication protocol. IEEE Access 6:16623–16638

Nehinbe JO (2011) A critical evaluation of datasets for investigating idss and ipss researches. In: 2011 IEEE 10th international conference on cybernetic intelligent systems (CIS). IEEE, pp 92–97

Ngiam J, Khosla A, Kim M, Nam J, Lee H, Ng AY (2011) Multimodal deep learning. OpenReview

Nguyen MT, Kim K (2020) Genetic convolutional neural network for intrusion detection systems. Fut Gener Comput Syst 113:418–427

Nguyen TT, Reddi VJ (2019) Deep reinforcement learning for cyber security. arXiv preprint arXiv:1906.05799

Nskh P, Varma MN, Naik RR (2016) Principle component analysis based intrusion detection system using support vector machine. In: 2016 IEEE international conference on recent trends in electronics, information & communication technology (RTEICT). IEEE, pp 1344–1350

Nyathi T, Pillay N (2018) Comparison of a genetic algorithm to grammatical evolution for automated design of genetic programming classification algorithms. Expert Syst Appl 104:213–234

Ojugo A, Eboka A, Okonta O, Yoro R, Aghware F (2012) Genetic algorithm rule-based intrusion detection system (gaids). J Emerg Trends Comput Inf Sci 3(8):1182–1194

Othman SM, Ba-Alwi FM, Alsohybe NT, Al-Hashida AY (2018) Intrusion detection model using machine learning algorithm on big data environment. J Big Data 5(1):1–12

Palmieri F (2019) Network anomaly detection based on logistic regression of nonlinear chaotic invariants. J Netw Comput Appl 148:102460

Pandey A, Sinha A, PS A (2019) Intrusion detection using sequential hybrid model. arXiv preprint arXiv:1910.12074

Porras PA, Fong MW, Valdes A (2002) A mission-impact-based approach to infosec alarm correlation. In: International workshop on recent advances in intrusion detection. Springer, pp 95–114

Praanna K, Sruthi S, Kalyani K, Tejaswi AS (2020) A CNN-LSTM model for intrusion detection system from high dimensional data

Pradhan M, Nayak CK, Pradhan SK (2020) Intrusion detection system (ids) and their types. In: Securing the internet of things: concepts, methodologies, tools, and applications. IGI Global, pp 481–497

Prusty S, Levine BN, Liberatore M (2011) Forensic investigation of the oneswarm anonymous filesharing system. In: Proceedings of the 18th ACM conference on Computer and communications security, pp 201–214

Qin X, Lee W (2003) Statistical causality analysis of infosec alert data. In: International workshop on recent advances in intrusion detection. Springer, pp 73–93

Rao CS, Raju KB (2019) Mapreduce accelerated signature-based intrusion detection mechanism (idm) with pattern matching mechanism. In: Soft computing in data analytics. Springer, pp 157–164

Revathi S, Malathi A (2013) A detailed analysis on nsl-kdd dataset using various machine learning techniques for intrusion detection. Int J Eng Res Technol 2(12):1848–1853

Sakurada M, Yairi T (2014) Anomaly detection using autoencoders with nonlinear dimensionality reduction. In: Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis, pp 4–11

Saleh AI, Talaat FM, Labib LM (2019) A hybrid intrusion detection system (hids) based on prioritized k-nearest neighbors and optimized svm classifiers. Artif Intell Rev 51(3):403–443

Sangster B, O'Connor T, Cook T, Fanelli R, Dean E, Morrell C, Conti GJ (2009) Toward instrumenting network warfare competitions to generate labeled datasets. In: CSET

Sanjaya SKSSS, Jena K (2014) A detail analysis on intrusion detection datasets. In: 2014 IEEE international advance computing conference (IACC)

Saranya T, Sridevi S, Deisy C, Chung TD, Khan MA (2020a) Performance analysis of machine learning algorithms in intrusion detection system: a review. Procedia Comput Sci 171:1251–1260

Saranya T, Sridevi S, Deisy C, Chung TD, Khan MA (2020b) Performance analysis of machine learning algorithms in intrusion detection system: a review. Procedia Comput Sci 171:1251–1260

Sato M, Yamaki H, Takakura H (2012) Unknown attacks detection using feature extraction from anomaly-based ids alerts. In: 2012 IEEE/IPSJ 12th international symposium on applications and the internet. IEEE, pp 273–277

Şen S, Clark JA (2009) A grammatical evolution approach to intrusion detection on mobile ad hoc networks. In: Proceedings of the second ACM conference on Wireless network security, pp 95–102

Seo S, Park S, Kim J (2016) Improvement of network intrusion detection accuracy by using restricted Boltzmann machine. In: 2016 8th International conference on computational intelligence and communication networks (CICN). IEEE, pp 413–417

Serrano W (2019) The blockchain random neural network in cybersecurity and the internet of things. In: IFIP international conference on artificial intelligence applications and innovations. Springer, pp 50–63

Sharafaldin I, Gharib A, Lashkari AH, Ghorbani AA (2018) Towards a reliable intrusion detection benchmark dataset. Softw Networking 2018(1):177–200

Shen Y, Han T, Yang Q, Yang X, Wang Y, Li F, Wen H (2018) Cs-cnn: enabling robust and efficient convolutional neural networks inference for internet-of-things applications. IEEE Access 6:13439–13448

Shiravi A, Shiravi H, Tavallaee M, Ghorbani AA (2012) Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Comput Secur 31(3):357–374

Shone N, Ngoc TN, Phai VD, Shi Q (2018) A deep learning approach to network intrusion detection. IEEE Trans Emerg Top Comput Intell 2(1):41–50

Signal detection theory (2019) Signal detection theory. http://gim.unmc.edu/dxtests/roc2.htm/. Accessed 2019

Sohi SM, Seifert JP, Ganji F (2021) Rnnids: enhancing network intrusion detection systems through deep learning. Comput Secur 102:102151

Sommer R, Paxson V (2010) Outside the closed world: on using machine learning for network intrusion detection. In: 2010 IEEE symposium on security and privacy. IEEE, pp 305–316

Song J, Takakura H, Okabe Y, Eto M, Inoue D, Nakao K (2011) Statistical analysis of honeypot data and building of kyoto 2006+ dataset for nids evaluation. In: Proceedings of the first workshop on building analysis datasets and gathering experience returns for security, pp 29–36

Sperotto A, Sadre R, Van Vliet F, Pras A (2009) A labeled data set for flow-based intrusion detection. In: International workshop on IP operations and management. Springer, pp 39–50

Srimuang W, Intarasothonchun S (2015) Classification model of network intrusion using weighted extreme learning machine. In: 2015 12th International joint conference on computer science and software engineering (JCSSE). IEEE, pp 190–194

Srinivas J, Das AK, Kumar N (2019) Government regulations in cyber security: framework, standards and recommendations. Fut Gener Comput Syst 92:178–188

Srivastava N, Salakhutdinov RR (2012) Multimodal learning with deep Boltzmann machines. In: Advances in neural information processing systems, pp 2222–2230

Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958

Tabakhi S, Moradi P, Akhlaghian F (2014) An unsupervised feature selection algorithm based on ant colony optimization. Eng Appl Artif Intell 32:112–123

Taddeo M, McCutcheon T, Floridi L (2019) Trusting artificial intelligence in cybersecurity is a double-edged sword. Nat Mach Intell 66:1–4

Tang TA, Mhamdi L, McLernon D, Zaidi SAR, Ghogho M (2016) Deep learning approach for network intrusion detection in software defined networking. In: 2016 International conference on wireless networks and mobile communications (WINCOM). IEEE, pp 258–263

Tavallaee M, Bagheri E, Lu W, Ghorbani AA (2009) A detailed analysis of the kdd cup 99 data set. In: 2009 IEEE symposium on computational intelligence for security and defense applications. IEEE, pp 1–6

Tidjon LN, Frappier M, Mammar A (2019) Intrusion detection systems: a cross-domain overview. IEEE Commun Surv Tutor 21(4):3639–3681

Tirumala SS (2014) Implementation of evolutionary algorithms for deep architectures. In: CEUR workshop proceedings

Tong X, Wang Z, Yu H (2009) A research using hybrid rbf/elman neural networks for intrusion detection system secure model. Comput Phys Commun 180(10):1795–1801

Tsoukalas LH, Uhrig RE (1997) Fuzzy and neural approaches in engineering. 18216097198

U. of massachusetts amherst (2019) U. of massachusetts amherst, optimistic tcp acking. http://traces.cs.umass.edu/. Accessed 12 Feb 2019

Uci (2019) Uci. http://archive.ics.uci.edu/ml/datasets/kdd+cup+1999+data. Accessed 15 Apr 2020

Unb (2019a) Unb. http://www.unb.ca/cic/datasets/nsl.html. Accessed 15 Apr 2020

Unb (2019b) https://www.unb.ca/cic/datasets/ids.html Accessed 15 Apr 2020

Unb (2019c) https://www.unb.ca/cic/datasets/dos-dataset.html. Accessed 15 Apr 2020

Viegas F, Rocha L, Gonçalves M, Mourão F, Sá G, Salles T, Andrade G, Sandin I (2018) A genetic programming approach for feature selection in highly dimensional skewed data. Neurocomputing 273:554–569

Vinayakumar R, Alazab M, Soman K, Poornachandran P, Al-Nemrat A, Venkatraman S (2019) Deep learning approach for intelligent intrusion detection system. IEEE Access 7:41525–41550

Wang H, Gu J, Wang S (2017) An effective intrusion detection framework based on SVM with feature augmentation. Knowl Based Syst 136:130–139

Wisesty UN, et al. (2017) Comparative study of conjugate gradient to optimize learning process of neural network for intrusion detection system (ids). In: 2017 3rd International conference on science in information technology (ICSITech). IEEE, pp 459–464

Worku A (2019) Minimizing black hole attack in mobile ad hoc network with anomaly based ids approach. PhD thesis, ASTU

Yang S, Li M, Liu X, Zheng J (2013) A grid-based evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 17(5):721–736

Yang Y, Zheng K, Wu C, Niu X, Yang Y (2019) Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks. Appl Sci 9(2):238

Yin C, Zhu Y, Fei J, He X (2017) A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access 5:21954–21961

Yu J, Reddy YR, Selliah S, Kankanahalli S, Reddy S, Bharadwaj V (2004) Trinetr: an intrusion detection alert management systems. In: 13th IEEE international workshops on enabling technologies: infrastructure for collaborative enterprises. IEEE, pp 235–240

Zhang Q, Yang LT, Chen Z (2015a) Deep computation model for unsupervised feature learning on big data. IEEE Trans Serv Comput 9(1):161–171

Zhang Q, Yang LT, Chen Z (2015b) Privacy preserving deep computation model on cloud for big data feature learning. IEEE Trans Comput 65(5):1351–1362

Zhang Y, Zhang Y, Zhang N, Xiao M (2020) A network intrusion detection method based on deep learning with higher accuracy. Procedia Comput Sci 174:50–54

Zhao L, Hu Q, Wang W (2015) Heterogeneous feature selection with multi-modal deep neural networks and sparse group lasso. IEEE Trans Multimed 17(11):1936–1948

Zhao S, Li W, Zia T, Zomaya AY (2017) A dimension reduction model and classifier for anomaly-based intrusion detection in internet of things. In: 2017 IEEE 15th international conference on dependable, autonomic and secure computing, 15th international conference on pervasive intelligence and computing, 3rd international conference on big data intelligence and computing and cyber science and technology congress (DASC/PiCom/DataCom/CyberSciTech). IEEE, pp 836–843

Ziegler S (2019) Internet of things cybersecurity paradigm shift, threat matrix and practical taxonomy. In: Internet of things security and data protection. Springer, pp 1–7

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