Neural Computing and Applications (2023) 35:4993–5013
https://doi.org/10.1007/s00521-021-06093-5

S.I. : MACHINE LEARNING APPLICATIONS FOR SECURITY

An effective NIDS framework based on a comprehensive survey of feature optimization and classification techniques

Pankaj Kumar Keserwani[1] · Mahesh Chandra Govil[1] · Emmanuel S. Pilli[2]

Received: 30 December 2020 / Accepted: 28 April 2021 / Published online: 22 May 2021

Abstract
The technological advancement leads to an increase in the usage of the Internet with many applications and connected devices. This increased network size causes increased complexity and creating rooms for the attackers to explore and exploit vulnerabilities to carry out various attacks. As a result upsurge of network attacks can be realized in recent years and is diversified, which can be affirmed by the admittance of various organizations. Varieties of intrusion detection systems (IDSs) have been designed and proposed to tackle such issues based on the misuse-based, anomaly based, and sometimes hybrid techniques. The high rate of network data generation and its enormous volume makes it challenging for IDSs to maintain their efficacy and reliability. This paper discusses a comprehensive understanding of IDS types, six benchmark network datasets, high distributed dimensionality reduction techniques, and classification approaches based on machine learning and deep learning for intrusion detection with their importance to ascertain the efficacy and reliability of IDSs. Furthermore, based on the literature review, a general framework for NIDS has been proposed. At last model for network IDS (NIDS) is designed by following the proposed framework. Achieved accuracy and detection rate of the proposed NIDS model on the UNSW-NB15 dataset are 98.11% and 97.81%, respectively, and achieving better performance than other approaches comparatively.

Keywords Intrusion detection · Machine learning · Attacks · Feature optimization · Deep learning

1 Introduction

Cyberattacks are increasing as the technologies are expanding to facilitate its users. Some attacks are witnessed recently includes Ransomwareattack in Texas towns in 2019 [1], WannaCry attack in the UK in 2017 [1], etc. These cyberattacks lead to catastrophic financial loss as well as defamation to the organizations and industries. In a new study, it has been discovered that a big company suffers from a loss of $10.3 million on an average every year due to cyberattacks. In contrast, a medium-sized company suffers $11,000 on an average annually [2]. As reported in [3] cybercrime loss is going to reach $6 Trillion by coming 2021. The average cost of tackling an attack is estimated to be $3,000,000. Malicious insider attacks caused the loss of $173,516 in 2017 as reported by Dr. ErdalOzkaya and MiladAslaner in their book titled "Hands-On Cyber Security for Finance" [4]. The modern attackers are highly motivated to conduct the attacks and have enough resources, time, and money to support them in achieving the targeted goals. Attackers conduct the attacks in large numbers and in a well-organized and sophisticated manner. They have easy access to the required tools and techniques, i.e., malware infection frameworks like Zeus, SpyEye, drive-by-download Web toolkit, etc., for performing the attacks easily [5]. The attack causes a threat to the confidentiality, integrity, or availability (CIA) of an

Pankaj Kumar Keserwani and Mahesh Chandra Govil have contributed equally to this work

✉ Pankaj Kumar Keserwani
pankajkeserwani.cse@nitsikkim.ac.in

Mahesh Chandra Govil
director@nitsikkim.ac.in

Emmanuel S. Pilli
espilli.cse@mnit.ac.in

[1] Department of Computer Science and Engineering, National Institute of Technology Sikkim, Burfang Block Ravangla, South Sikkim 737139, India

[2] Department of Computer Science and Engineering, Malaviya National Institute of Technology Jaipur, JLN Marg, Jaipur, Rajasthan 737139, India

information system. The intrusion prevention techniques such as encryption and authentication alone are insufficient to fulfill all the security needs of the current time. Intrusion detection can be an alternative to increase network security. Intrusion is an unauthorized activity or attack which causes damage to the information system. On detecting an intrusion, an alert is generated to the network administrator to minimize the system damage. Different researchers have proposed numbers of IDSs with different machine learning (ML) and deep learning (DL) techniques to address various needs in detecting intrusions or attacks. Earlier network intrusion detection was conducted manually, where all the network activities are monitored, collected, and analyzed with the help of system analysts and system administrators to find out the malicious activities. Later, with the advent of technologies and increased users, the size and complexity of network flow also increased due to which manual intrusion detection became very hard or nearly impossible. To handle the issue misuse-based intrusion detection was introduced. With the rise in complexity of the network, new types of attacks also came into the picture. To handle the new attacks, data mining and machine learning (ML) based methods were introduced. Firstly, supervised approaches were used, which can detect only the predefined attacks with high accuracy and low false-positive rate. The supervised approach failed to detect the new attacks known as zero-day attacks. The unsupervised learning methods can detect the new type of attacks but has a high false-positive rate (FPR). Hence, hybrid approaches of supervised and unsupervised methods sometimes perform better in detecting network intrusions. The changing structure of the network is generating the distributed data with a large number of attributes. As a result, the traditional NIDSs are facing challenges in detecting network intrusions efficiently and effectively. The researchers tried to use feature selection (FS) strategies with ML techniques to design effective IDS approaches. Recently many researchers have proposed ML-based solutions for network attack detection. The solutions include data pre-processing, feature selection (FS), hyper-parameter selection (HS), ML algorithms, and performance metrics.

Siddique et al. [6] have highlighted the shortcomings of many earlier works on KDDCup'99 datasets for NIDS. Rathore et al. [7] developed an IDS architecture to address challenges that occurred due to the large volume of real-time network traffic. The proposed NIDS model was evaluated using well-known ML algorithms on the KDDCup'99 dataset. Additionally, the impact of the feature selection approach on classification has also been evaluated. Sharafaldin et al. [8] introduced a network dataset from the emulated network that consists of up-to-date patterns for attack and normal traffic and evaluated it on seven ML algorithms. Li et al. [9] have developed NIDS for anomaly detection where they utilized long short-term memory (LSTM) neural network and broad learning techniques (BLS) as DL-based techniques and the model was evaluated on NSL-KDD and a Border Gateway Protocol (BGP) based datasets. However, the proposed NIDS anticipated all required mentioned stages except one important step known as feature selection (FS). Le et al. [10] have proposed a FS-based approach to improve the detection rate of the classifier in a networks and they considered NSL-KDD and ICSX12 datasets to evaluate the approach together with many ML and DL algorithms. Divekar et al. [11] highlighted flaws of KDDCup'99 to evaluate the modern NIDS using different ML algorithms. They eliminated the irrelevant features using the Mean Decrease Impurity (MDI) technique and addressed the data imbalance issue through SMOTE. Belouch et al. [12] have applied many classical ML algorithms on all available features of the UNSW-NB15 dataset for proposing the NIDS model. They found that Naïve Bayes was the fastest and Random Forest was providing the most accurate prediction. Hussain and Lalmuanawma [13] evaluated an extensive group of classical ML algorithms on KDDCup'99 and NSL-KDD datasets for anomaly detection in networks.

This paper discusses a comprehensive literature review for understanding the IDS with its types, six benchmarked networking datasets used to validate the designed NIDS, dimensionality reduction techniques, approaches for classification based on ML and DL. Furthermore, based on the literature review, a framework to design NIDS models has been proposed. At last, one NIDS model is presented as a proof of concept. The major contributions of the paper are as follows:

1. The review of different types of IDS using tree structure to understand the baseline difference among them.
2. Brief description of six benchmark networking datasets used to evaluate the performance of designed NIDS.
3. The review of feature optimization techniques with their importance to understand the baseline difference among them.
4. The review of machine learning (ML) and deep learning (DL) based classification techniques for detecting intrusion with their importance to ascertain the efficacy and reliability.
5. Proposal of a framework based on the literature review to design an efficient and effective NIDS.
6. Simulation of a NIDS model based on the proposed framework as a proof of concept.
7. Evaluation of the developed NIDS model on a relevant benchmark networking dataset - UNSW-NB15.

The rest of the paper is organized as follows: Sect. 2 provides a discussion of different types of IDS in a tree structure format. Section 3 describes the six benchmarked publically available networking datasets. Section 4 presents different feature optimization techniques. Section 5 elaborates the classification of ML and DL techniques. Section 6 presents a literature survey for research works in NIDS. Section 7 represents a proposed NIDS framework based on the survey. Section 8 represents the simulation, results, and comparative analysis of a developed NIDS model based on the proposed framework. Finally, Sect. 9 concludes the paper.

2 IDS classification

In this modern era, many advanced technologies are developed like the Internet of Things (IoT) and Cloud Computing, edge computing, etc., to facilitate the user. But these advancements are also creating room for the attacker to explore and exploit different types of attacks. When someone uses the Internet from the electronic device such as a laptop, desktop, mobile, IoT device, he is always at risk because someone within the connected network or Internet can access confidential information or launch an attack. When the device is isolated, someone may attempt to get physical access to misuse it. Intrusion detection is an art, science, technique, or tool to detect host or network attacks. An intrusion detection system (IDS) can be hardware or software for monitoring the malicious activities of a host or network [14]. The IDS detects intrusions by examining and analyzing the network packets. When an intrusion is detected, the IDS generates an alarm, terminates the connection, and drops or blocks the offending packets. Classifications of IDSs are provided in Fig. 1.

1. *Architecture based* [15] On the basis of architecture, the IDS is divided into three categories—centralized, decentralized, and distributed, which are given below as:
   (a) *Centralized IDS* In this, a central server is there to receive the signals from multiple sensors of a specified network for which it has been set up. The central server analyses the received signals to detect the intrusions. This architecture is better for a small network and is prone to a single point of failure (SPoF).
   (b) *Decentralized IDS* In this IDS, multiple servers are there in the specified networks for which they have been set up and receiving signals from their closed sensors. Each of these servers preprocesses the received signals and sends the signal data to the main server periodically after a certain duration. These IDS can handle big networks with higher performance than the centralized IDS and is able to avoid the SPoF.
   (c) *Distributed IDS* In this IDS, multiple autonomous agents act as the sensors. The agents collect the signals and are able to preprocess the collected data at the same time. The agents communicate with each other by following the peer-to-peer (P2P) protocol, where the concept of the main server is not there, and the workload is distributed among the agents. Distributed IDS can be set up for a larger network than centralized IDS and decentralized IDS. In both

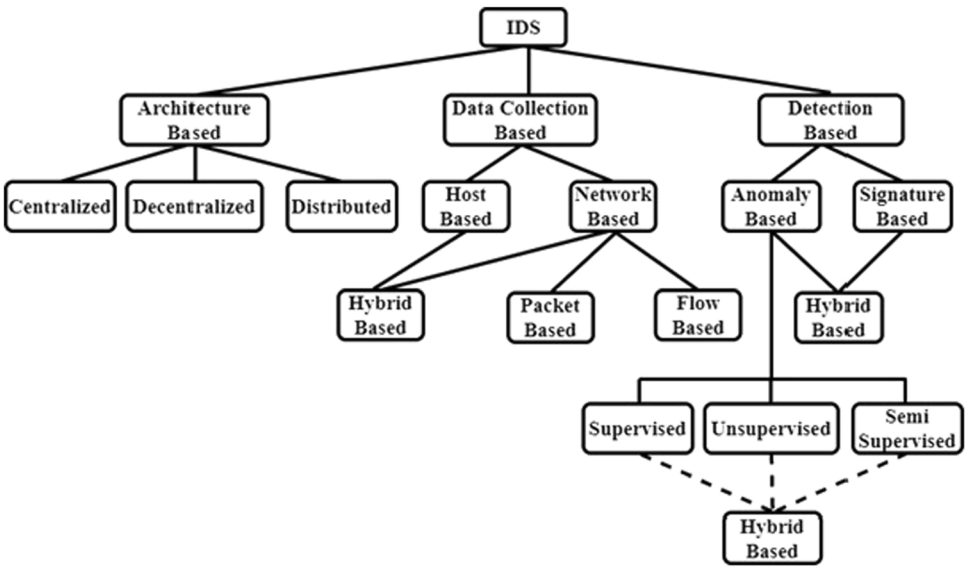

Fig. 1 General classification of intrusion detection systems

decentralized and distributed IDSs, communication between agents is very important and crucial. If communication is lost, it will be very difficult to detect the attacks.

2. *Data Collection Based Classification of IDS* It classifies the IDS into three types

   (a) *Host-based intrusion detection system (HIDS)* It is set up to monitor the traffic of a particular host for detecting the intrusions, which can be a rogue process, unauthorized access, or modification of system files. The agent software monitors and checks for unauthorized access, application software's suitability, DoS attack on the host. In a host-based IDS solution, agent software runs on a host system, such as a server or PC. The agent software stores all actions locally and sends them to the central repository periodically. Many researchers proposed HIDS in their works, such as [16, 17].

   (b) *Network-based intrusion detection system (NIDS)* In this, the network traffic is monitored and analyzed to detect the intrusions as anomalies. Many researchers have proposed NIDS in their works, such as [18, 19]. In the network-based IDS, a sensor is used for a segment to monitor and examines the network traffic packets and their contents to detect attacks. The sensor captures packets and checks them as per the intrusion detection rules. Examples of network-based IDS are Cisco hardware sensors of 4200 series, Catalyst 6500 switch IDS module, etc. The sensor can log the malicious activities, drop or block malicious packets, and terminate the malicious connections.

   (c) *Hybrid IDS* For taking the benefits of high detection rate and low false alarm rate of signature-based IDS and detecting zero-day attack of profile-based IDS hybrid implementation is utilized, which is known as hybrid IDS. Host and network-based IDS should be combined in such a way that the performance of IDS can be improved; such IDS is known as hybrid IDS. Mohan et al. [20] designed a hybrid IDS by combining multivariate correlation analysis (MCA), integrity check using a hash, signature-based updates for features, and detection techniques. The single hybrid IDS uses a complex event processing (CEP) as the backbone, which combines and correlates the input from all the host and network IDS deployments for better accuracy.

   For a big enterprise, hybrid IDS (host-based and network-based) provides a perfect solution in a balanced manner. The network IDS faces problems when it deals with a large volume of traffic, and more host-based IDSs create difficulty in management. The host-based IDS can manage the traffic for its system efficiently. Sensors can be placed anywhere to find the maximum coverage, generally at the backbone. After that, key hosts are identified to install agent software to provide extra security to the host applications. The IDS solution is considered effective when it detects the network and host vulnerabilities in a real-time manner.

3. *Detection Based Classification of IDS* IDS can be classified further into three types: Signature-based IDS, anomaly or behavior-based IDS, and hybrid IDS.

   (a) *Misuse or signature-based IDS* In this, the IDS model is trained based on historical data so that it may learn the predefined signatures and classifies the new attacks as one of them. When implemented in a real environment, it compares the data stream from the signatures to detect the intrusion. This type of IDS has a low false alarm rate (FAR) and hence is preferred IDS by the organization. Some examples of these type IDS are Suricata [21], SNORT [22], etc., and also known as misuse-based intrusion detection. The signature-based IDS challenges are: (1) The signature-based IDS detects only those attacks for which it has trained. Hence, the IDS system should be trained with up-to-date signatures. (2) It does not perform when it deals with the attack, which is repeated over a long period due to its limited buffer size; it cannot store the signature for a long time.

   (b) *Anomaly based IDS* In this, a model is trained on the basic behavior for which the instances follow the normal behavior. If an instance does not confirm the normal behavior, i.e., deviate from normal behavior, it is considered an anomaly or outlier and is also known as profile-based IDS. In this, the traffic is captured, normally known as profiling, and a profile is used for comparing the traffic to find the network attacks. A profile is nothing but information on the network traffic patterns as well as statistics. The anomaly based IDS can detect the zero-day attack. Many applications are used for anomaly detection, such as credit card fraud detection, ad fraud detection, bank forgery detection. [23]. Anomaly based IDS is (1) prone to high false alarm rate (FAR) than the signature-based IDS due to

variation of the network traffic patterns and consume more computing in detecting an attack. (2) Needs more time in profile recreation due to variations of network traffic patterns from different network topologies. (3) Needs more care in profile capturing to avoid a high false alarm rate.

(c) *Hybrid IDS* In both the cases, misuse based and anomaly based are combined with utilizing the attain low FAR and zero-day attack detection capability. Many researchers have proposed hybrid IDS, such as Kim et al. (2014) [24] used the hybridization of the C4.5 decision tree for signature-based IDS and support vector machines (SVMs) for anomaly based IDS. Hatef et al. (2018) [6] presented a hybrid NIDS known as hybrid intrusion detection approach in cloud computing (HIDCC) with the help of Snort for signature-based attack and a combination of learning vector quantization algorithm and C4.5 algorithm for anomaly detection.

3 Datasets used in IDS

The capabilities of any intrusion detection system to detect the intrusive element are established through the datasets upon which it is validated. Despite of unavailability of network packet data analysis in most of the commercial product setup because of privacy issues, numerous standard datasets such as NSL-KDD99 [7], NSL-KDD [8], AWID [9], ISCX 2012 [10], UNSW-NB15 [11], CICIDS 2017 [12] are made publically accessible for research & development purpose. This section discusses the features of six popular benchmark IDS datasets in brief.

1. *KDD Cup 99 dataset* This dataset by Defence Advanced Research Project Agency (DARPA): Created in 1988 by DARPA, this knowledge discovery and data mining (KDD) dataset is one of the earliest but important contributions to the intrusion detection system research. However, the aim was to create a detailed and realistic IDS research benchmarking dataset, but its correctness and competence to inspect a realistic intrusive behavior remained under the question to date. The dataset was collected through simulated intrusions on a modeled network which was believed to resemble the actual US Air Force LAN. KDD was collected for two months' duration and consists of a total of nearly 4,900,000 records of a series of TCP sessions representing a large variety of simulated network attacks. A portion of it with 2,000,000 connection records with 41 attributes was derived and used for the third international KDD tools competition in 1999 and hence named as KDD Cup99 dataset. This data set has around 75% duplicated traffic records in both training and testing data, which can lead to biased behavior in the IDS build around these data and reduce the capability to detect abnormal traffic in the network. This dataset is no longer up-to-date as it does not include most of the modern malicious attacks but still playing an important role in IDS research and is widely being used by researchers.
2. *NSL-KDD dataset* This is derived from the famous KDD Cup99 dataset. The duplicated traffic records present in KDD Cup99 data affect the abnormality detection accuracy of any IDS system builds around these data. And ultimately, the evaluated IDS becomes misleading as the huge number of duplicated records prevents these systems to learn about irregular instances of traffic packets. By removing the duplicated records from the original KDD Cup99 data, the resultant NSL-KDD data consist of only 125,973 and 22544 records in training and testing data, respectively. IDS built around these data sets are more consistent and accurate. This dataset has 22 variants of intrusion attacks and 41 attributes (21 of the TCP connections and 19 of nature of the connection) as earlier.
3. *Aegean WiFi Intrusion Dataset (AWID)* This dataset is a labeled dataset, and it contains 155 features 802.11 Wi-Fi network environment. The last attribute or column is labeled one telling about normal and attack intrusion type. The dataset has been divided into four subgroups as AWID-CLS-F, AWID-ATK-F, AWID-CLS-R, and AWID-ATK-R.

   (a) *AWID-CLS-F* The AWID-CLS-F sub-dataset contains granular labeling for each class of attacks. AWID-CLS-F contains 37,817,835 training samples with 15,77,49,037 normal as well as 46,26,210 attack samples of three types and 4,85,24,866 test samples with 4,73,25,477 normal as well as 11,99,389 attack samples of three types.

   (b) *AWID-ATK-F* The AWID-ATK-F sub-dataset contains granular labeling for each class of attacks. AWID-ATK-F contains 16,23,75,247 training samples with 15,77,49,037 normal as well as 46,26,210 attack samples of nine types and 4,85,24,866 test samples with 4,73,25,477 normal as well as 11,99,389 attack samples of sixteen types.

   (c) *AWID-CLS-R* The AWID-CLS-R sub-dataset contains granular labeling for each class of attacks. AWID-CLS-R contains 17,95,575

training samples with 16,33,190 normal as well as 1,62,385 attack samples of three types and 5,75,643 test samples with 5,30,785 normal as well as 44,858 attack samples of three types.

(d) *AWID-ATK-R* The AWID-ATK-R sub-dataset contains granular labeling for each class of attacks. AWID-ATK-R contains 1,765,000 training samples with 1,633,190 normal and 162,385 attack samples of nine types and 575,467 test samples with 539,785 normal and 44,858 attack samples of fourteen types.

4. *ISCX 2012 dataset* In this dataset real traffic of various protocols like HTTP, SMTP, SSH, IMAP, POP3, and FTP was analyzed to distinguish between normal traffic and intrusive traffic. This dataset contains a large range of real network attacks. The ISCX dataset was created at the Information Security Centre of Excellence at the University of New Brunswick. This dataset contains two million labeled data samples for twenty attributes. The dataset covers network activities of seven days for data related to normal and attacks traffic of four types, namely Brute Force SSH, HTTP DoS, Infiltrating, and DDoS. Only two percent of the sample data of the whole dataset belongs to attack traffic.
5. *UNSW NB15 dataset*The UNSW-NB-15 dataset [11] was developed by Australian Center for Cyber Security (ACCS) in 2015. The experimental set up was created for normal and abnormal network traffics using the IXIA PerfectStorm tool, and nine different kinds of network attacks were simulated via two different simulation schemes. The first simulation was run for 16 hours with one attack per second, while the other simulation was performed for 15 hours, with 10 attacks per second. This dataset consists of a total of 43 three features. This dataset equally encapsulates contemporary and modern synthesized network attacks making it suitable for evaluating modern IDSs. Network traffic packets were captured using the tcpdump tool, and reliable dataset features were created using Argus and BroIDS tools with the help of 12 algorithms written in C# language to analyze the depth-flow of TCP/IP connections. IXIA reporting was used for labeling the dataset. The network configuration was established using IXIA traffic generator with three servers in which two servers were designated for normal traffic generation, and one was to incorporate malicious activities in the network. The intercommunication between servers and hosts was ensured using two routers that were further connected to the firewall, allowing all traffic to pass through it. On one of these routers, the tcpdump tool was deployed to capture traffic packets in Pcap files. Traffic analysis was done of the captured Pcap file to come to the dataset statistics based on different packet features. Tables 1 and 2 demonstrate the simulation schemes and data collection details from them.

   A total of 49 features were extracted from dumped Pcap files using Argus and Bro-IDS tools. Argus toll consisting of Argus Server-Client architecture is responsible for generating attributes of the network packets by processing Pcap files. Bro-IDS monitor and label the network traffic as normal or abnormal. Finally, the features created by both the tools are matched and stored in a CSV file. The features in UNSW-NB15 datasets are grouped under (1) flow features, (2) basic features, (3) content features, (4) time features, and (5) additional features, and (6) label features. This dataset has a total number of 1,75,341 records in its training set and 82,332 records in the testing set. This dataset has been used for experiments and performance evaluation of our proposed model.
6. *CICIDS-2017 dataset* The CICIDS2017 dataset available with eight different files containing realistic traffic background. These files are generated at the Canadian Institute of Cybersecurity for different network events as normal and attacks of five days network traffic with the help of 25 users. This dataset had been collected on the basis of real traces. The dataset many different real attack scenarios such as Botnet, DoS Attack, DDoS Attack, Brute Force Attack, HeartBleed Attack, Web Attack, and Infiltration Attack and benign as the different activities of the network traffic. The CICIDS2017 dataset contains 2,830,108 data samples where 2,358,036 data samples belong to benign traffic, and 471,454 data samples belong to malicious. The CICIDS2017 dataset is one of the good datasets which contains up-to-date novel types of attacks.

Table 1 Simulation details of scheme 1 and scheme 2

Network features	Scheme 1	Scheme 2
No. of attacks	1/sec	10/sec
Duration of simulation	16 h	15 h
Data collected	50 GB	50 GB

Table 2 Data collection details of scheme 1 and scheme 2

Traffic features	No. of packets in scheme1	No. of packets in scheme2
Total_Flows	987,627	976,882
Source_Packets	41,168,425	41,129,810
Destination_Packets	53,402,915	52,585,462
Source_Bytes	4,860,168,886	5,940,523,728
Destination_Bytes	44,743,560,943	44,303,195,509
Normal_Packets	1,064,987	1,153,774
Malicious_Packets	22,215	299,068

4 Feature-optimization techniques

4.1 Statistical-based methods

Many feature selection methods are there, but the most common methods are filter method, wrapper method, embedded method, and hybrid method [13]. The filter method uses constitutive properties such as divergence or correlation from the dataset for selecting features [25]. A threshold is defined on the basis of some relevant criteria such as information gain (IG), gain ratio (GR), correlation-based feature selection (CFS); the features which do not satisfy the threshold criteria are ignored. The wrapper method is the same as the filter method. Only the difference is that the classifier is independent of the filter method, while the classifier is part of the evaluation phase for selecting the features. It selects or removes features on the basic objective function. Hence, it takes more time than the filter method but is more accurate than the filter method [26] and causes overfitting if the used classification algorithm learns perfectly [27]. In the embedding method, an ML model such as decision tree is trained to decide the weights of each feature which helps in selecting features [28]. Hence, if the same classifier performs feature selection as well as the classification process, it is known as an embedded method. Here, the computation cost is less than the wrapper method [29]. The hybrid method uses two-level of filtering: first with wrapper methods and the output of filter class is applied to the input of the wrapper method for better feature selection [30]. The feature selection process must be taken care of since some features contribute more and confuse the classification.

4.2 Metaheuristic algorithms

Metaheuristic algorithms an optimization technique. When applied to find out the solution to a certain problem, the obtained solution may be either optimal or suboptimal. Metaheuristic algorithms may be exact or approximate algorithms. If the obtained solution is optimal, the metaheuristic algorithm is exact, and if the solution is near-optimal, the metaheuristic algorithm is the approximate.

The approximate algorithms can be classified as heuristic algorithms and approximation algorithms. The heuristic algorithm is able to provide a good solution within specified and acceptable time. The approximation algorithm is able to provide better-quality solutions within a specified duration. Heuristic algorithms are of two types: specific heuristic and metaheuristics. A specific algorithm is designed to solve the specific case only, while the metaheuristics algorithm is designed to solve any general optimization problem. Metaheuristic algorithms are utilized in numerous different fields. Metaheuristics algorithms are not complete methods since it does not provide the best global solution instead they provide approximate solutions [31].

A metaheuristic algorithm is designed in such a way that it works on the basis of exploration and exploitation to find the solution of a problem [32]. In the exploration phase, the object is moved to multiple sites to collects all information in order to find a suitable new site, know as a new learning procedure. All population-based metaheuristic algorithms follow this learning procedure. In the exploitation phase, all collected information in the exploration phase is utilized to find the best site satisfying the available resources.

Metaheuristic algorithms have been used in many different fields. One of the fields is feature selection, where many researchers have been shown the efficacy of these algorithms with high efficiency. Broadly metaheuristic algorithms are categorized into two subcategories: single-based metaheuristic algorithms, population-based metaheuristic algorithms. Single-based metaheuristic algorithms, also termed as trajectory algorithms, look to formulate an underlying solution to start and move away from exploring neighboring areas. In each execution, if the output is better than the current, the process continues; otherwise current one is considered as local minima. In this way, it follows a specific path [31]. Population-based metaheuristic algorithms start with a set of initialized solutions, and a population (another solution set) is created. Search is terminated if certain predefined criteria are fulfilled. The optimal solution obtained in such algorithms is totally dependent on the fact that how the population is manipulated. These algorithms are mostly categorized into

evolutionary computation (EC) and swarm intelligence (SI). For feature selection in IDS, population-based metaheuristic algorithms are utilized mostly [31].

For determining the most relevant features, we generally use the feature selection methods, which may belong to the filter method, wrapper method, embedded method, or hybrid method. One may choose a metaheuristic algorithm also such as GA, PSO. To do this, first, each solution in the swarm or population should be encoded in a binary form where 1 indicates a selected feature, and 0 indicates a non-selected feature. The size of each solution vector is the same as the original number of features. One may use another mechanism to represent solutions such as combinatorial problems. In the second step, the selected classifier which evaluates the selected features works as the fitness/objective function of the selected algorithm. Recently, many researchers have applied and are applying hybrid filter/wrapper feature selection approaches.

In the recent trends of feature engineering, feature selection works as a key tool for addressing the curse of dimensionality to increase the detection rate of IDS by using the classifiers of different types. Since it is very difficult to preprocess the large volume of data by mining and transformation. In general, strategy based or search-based feature selection techniques are popular for selecting the relevant features from a dataset [33]. Hajisalem et al. [34] utilized the hybridization artificial bee colony (ABC) and artificial fish swarm (AFS) algorithm for feature selection in their proposed classifier. The classifications of feature selection are presented in Fig. 2.

The filter methods are independent of data mining (DM) techniques, while the wrapper method is dependent on DM techniques. The search strategy plays a key role in search-based methods, which should be computationally economical. The feature selection process should be guided in such a way that the output must be information lossless [35].

The feature selection works conducted by different researchers are presented in Table 3.

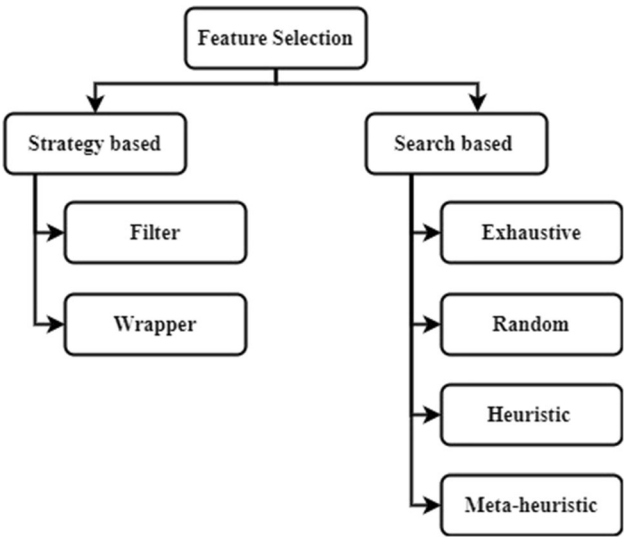

Fig. 2 Classification of feature selection techniques

5 Classification techniques

When an algorithm is designed for classification, it is known as a classifier. The learning by the classifier is known as classification techniques. Machine learning is the major representative classification technique, which performs artificial intelligence (AI) related tasks in the system such as recognition, prediction. [60]. Basically, four types of ML are there: supervised, unsupervised, semi-supervised, and re-enforcement [61].

1. *Supervised learning* In this, the target variable is provided already with the independent variables to the learning algorithm to learn the signature of each sample and the learning model is tested with testing samples. If satisfied each of the parameters, i.e., greater than or equal to the specified threshold for each of the parameters, then the model is implemented in the real environment. This technique is also termed as predictive or directed classification. The learning algorithm is known as a supervised learning algorithm. The state of art popular supervised algorithms is decision tree (DT), random forest (RF), stacking, bagging boosting, ensemble method, support vector machine, k-nearest neighbor (k-NN), artificial neural network (ANN), etc. [62].
2. *Unsupervised learning* In this, the target values are missing from the learning algorithms, making it impossible to train the learning algorithm. Hence, this learning cannot be applied to regression or prediction directly. In this learning, when the input data (X) without the target variable to the learning algorithm then models the insight structure or distribution among the data samples by itself and presents the interesting structure. This learning is also termed as descriptive or undirected classification. Some of the applications of unsupervised learning are clustering, anomaly detection, association rule mining, and latent variable models. The popular unsupervised algorithms are k-means, Apriori, self-organizing map (SOM), deep learning, etc.
3. *Semi-Supervised Machine Learning* In this learning, a large amount of data (X) is provided to the learning algorithm where a very small amount of labeled data. Hence, it can be seen as a hybrid version of supervised learning and unsupervised, i.e., this learning lies

Table 3 Feature selection works conducted by researchers

Reference	Method type	Method	Dataset	Year
[36–38]	Filter-based	CFS	AWID	2017, 2017, 2018
[36, 37]	Filter-based	Correlation	AWID	2017
[36, 37]	Wrapper-based	ANN	AWID	2017
[36, 37]	Wrapper-based	SVM	AWID	2017
[36, 37]	Wrapper-based	C4.5	AWID	2017
[39]	Wrapper-based	RBFC	AWID	2019
[40]	Filter-based	IG	AWID	2016
[41]	Feature transformation	PCA	NSL-KDD	2015
[42]	Filter	Ensembled filtering, consistency, correlation-based FS	NSL-KDD	2015
[43]	Filter	Combination of correlation coefficient and IG	NSL-KDD	2015
[44]	Hybrid (filter and wrapper)	Combination of different filters and stepwise RWA	NSL-KDD	2015
[45]	Filter	Gain raito, IG, Chi-Squared	NSL-KDD	2019
[46]	Metaheuristic	PSO, ACO, ABC	NSL-KDD	2018
[47]	Filter	LCFS	KDDcup99	2011
[48]	Metaheuristic	hypergraph-enetic algorithm		2017
[49]	Metaheuristic	AM-FOA	Benchmark dataset	2015
[50]	Metaheuristic	TVCPSO	NSL-KDD	2016
[51]	Metaheuristic	Pigeon inspired optimizer	UNSW-NB15	2020
[52]	Metaheuristic	Cuttlefish	KDDcup99	2019
[53]	Metaheuristic	PSO	NSL-KDD	2019
[54]	Filter	Information gain (IG)	NSL-KDD	2018
[55]	Filter	Rule based	UNSW-NB15	2019
[56]	Hybrid	Central point (CP) and Association Rule Mining (ARM).	UNSW-NB15	2017
[57]	Metaheuristic	Double PSO	CICIDS2017	2020
[58]	Hybrid	K-means + GA	NSL-KDD	2015
[59]	heuristic	CFS-BA	CICIDS2017	2020

between both supervised and unsupervised learning. This learning is more near to a real-world problem. The popular semi-supervised algorithms are deep belief networks, auto-encoder, etc.

4. *Reinforcement Learning (RL)* In type learning, an agent is used to learn the environment through its experiences by receiving the rewards or punishment through feedback on its actions. The rewards and punishment are nothing but the signals as positive for rewards and negative for punishment. The main goal of RL is to devise a perfect action-based model that can maximize its rewards. RL applications are video games, robotics, self-driving cars, etc., with the help of artificial intelligence (AI).

5.1 Deep learning (DL)

Deep learning is a subset of ML and refers to a broad class of ML techniques. Deep learning moves around many-core keywords such as supervised learning, semi-supervised learning, unsupervised learning, multiple layers, back-propagation to learn, AI, pattern recognition. [63]. DL algorithms are suitable for huge datasets. Based on the application of DL, it can further sub-classified as supervised such as Image classification, object detection, face recognition. Unsupervised such as word embedding, image encoding, and semi-supervised, the most appropriate for the real-world applications, where very few labeled data samples are available for a huge dataset. The categorization of deep learning is presented in Fig. 3.

Guo et al. [64] discuss the DL algorithm in detail in a book titled "Deep Learning." Pouyanfar et al. [65] discussed some of the DL applications in their paper. Deng and Lu [63] also discussed the applications of DL. Weston et al. [66] discussed the DL via semi-supervised or hybrid embedding.

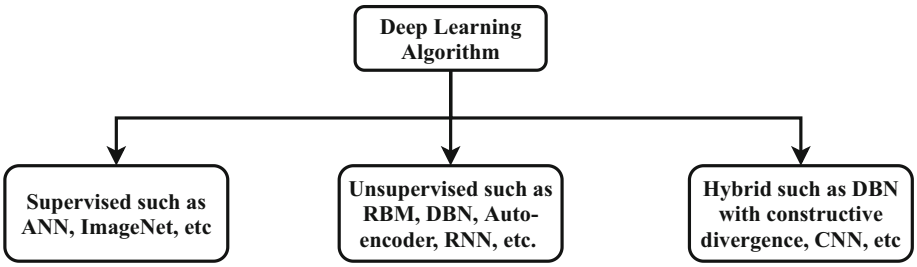

Fig. 3 DL algorithm classification

5.2 Performance metrics

A confusion matrix shows the number of correct and incorrect predictions made by the model against the actual outcomes (target value) in the data. Following Table 4 displays a 2X2 matrix for two classes (Positive and Negative).

Different parameters based on the confusion matrix are depicted.

- *True Positives (TP)* Correct positive prediction.
- *True Negatives (TN)* Correct negative prediction.
- *False Positives (FP)* Incorrect positive prediction.
- *False Negatives (FN)* Incorrect negative prediction.
- *Accuracy* It is equal to the correct predictions divided by total predictions.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

- *Error Rate* It is equal to incorrect predictions divided by total predictions.

$$\text{ErrorRate} = \frac{\text{FP} + \text{FN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{2}$$

- *Precision* It is the ratio of true positive prediction and total positive prediction. It is inversely proportional to false-positive rate (FPR).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3}$$

- *Recall (Sensitivity):* It is the ratio of true positive prediction and actual positives. It is also known as detection rate (DR).

Table 4 Confusion matrix

Confusion matrix	Model		
		Positive	Negative
Actual	Positive	TP	FP
	Negative	FN	TN

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4}$$

- *F1 score* It is equal to the weighted average of recall and precision. It is more informative than accuracy since it considers both false positives and false negatives and more helpful if the class distribution is uneven.

$$\text{F1score} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{5}$$

- *False-Positive Rate (FPR) or False Alarm Rate (FAR)* It is equal to the false positive predictions divided by actual negative values in the dataset.

$$\text{FPR or FAR} = \frac{\text{FP}}{\text{TN} + \text{FN}} \tag{6}$$

The false negative rate (FNR) is given below as:

$$\text{FNR} = 1 - \text{FPR} \tag{7}$$

6 Network intrusion detection survey

Wang et al. [67] transformed features before inputting them to SVM-based IDS. They applied the SVM-based IDS to the NSL-KDD dataset and achieved improved performance. They achieved the accuracy, DR, and FAR of 99.28 %, 99.16 %, and 0.61 %, respectively. George et al. [68] applied PCA for dimensionality reduction and SVM for anomaly detection in the network data. They tested their model on the KDD99 dataset with the precision and recall of 93.75% and 87.5 %, respectively. Raman et al.[48] used the combination of hypergraph and GA for feature selection and SVM for classification in their proposed IDS. They tested their HG-GA-SVM model on the NSL-KDD dataset with the DR and FAR of 97.14 % and 0.83 %, respectively. Hamamoto AH et al. [69] utilized GA for feature selection and fuzzy logic for detecting anomalous events in the network through their proposed IDS with the accuracy of 96.53 % on KDD99 dataset. Vijayanand R et al.[70] proposed an IDS model where they applied GA for feature selection and multiple SVMs for detecting

intrusion in wireless mesh networks environment. They tested their GA-SVM model on the CICIDS2017 dataset with an accuracy of 99.85 %. Kuang and Siyang [71] utilized kernel PCA (KPCA) and reduced the features. After that, they applied SVM with RBF function for classification of the intrusions on KDD99 and achieved DR of 94.22 % and FAR of 1.025 % with their N-KPCA-GA-SVM model.

Bamakan et al. [72] used ramp loss in SVM classifier so that class imbalance problems, as well as outliers, may be addressed and achieved higher accuracy of 98.6% on the NSL-KDD dataset. Viegas et al. [73] used multi-objective function for feature selection and applied state of art ML algorithms to detect intrusions in the network and achieve the accuracy of 99.99 % on the DARPA1998 dataset.

NIDS ensures the good security level of a network from various attacks. Various tools, approaches, and methods based on machine learning are there for detecting intrusion in a network. Gao et al. [74] applied DBN, SVM, ANN on the KDD CUP 1999 for Big Data prediction in IDS and found the DBN performance was better with the accuracy 93.49 %. Nguyen et al. [75] proposed a cyberattacks detection approach for mobile cloud environment with the accuracy of 97.11 % on the KDD CUP 1999 dataset.

Li et al. [76] applied autoencoder for feature selection and DBN as a classifier to detect intrusion as malicious code. In their experiment, they found DBN has successfully improved the accuracy in less time 92.1 % on KDDCUP'99 dataset, and Alom and Taha [77] combined autoencoders and RBMs for feature selection and dimensionality reduction after that Unsupervised Extreme Learning Machine (UELM) were used detection and achieve the accuracy 92.12 % on the KDD-99 dataset. Yogesh Sharma and Monika Rokade [78] designed a system to collect the network flow data with the help of Packet X LIB and WINCAP Driver for anomaly based intrusion detection. The Genetic Algorithm generates rules for the fuzzy logic controller and the output this controller provides to neural network algorithm to detect the anomaly.

Wang et al. [79] designed an intrusion detection algorithm and tested them in the CTU-13 dataset. In the preprocessing of the designed algorithm, the raw network data were converted into images, and then, the images are fed to a convolutional neural network (CNN). In the classification step, two approaches were followed: First, by using a 20-class classifier to identify traffic types as normal or malicious and achieve the accuracy of 99.17 %. In the second approach, a binary classifier was inputted to two CNNs for identifying traffic types as malicious or binary with an accuracy of 100 %.

Javaid et al. [80] designed a NIDS using a sparse autoencoder with two hidden layers and three softmax output layers as a binary class classifier as well as a multiclass classifier. In binary classifier classifies the network traffic as normal or malicious. The multiclass classifier classifies the network traffic in five classes, either normal or one of four attacks defined in the dataset. The accuracy of binary classifiers and a multiclass classifier were 88.4 % and 79.1 %, respectively.

Diro and Chilamkurti [81] utilized the autoencoder with three hidden layers and achieved an accuracy of 99.2 % for binary classification and 98.27 % for multiclass classification. They proposed a DL classification algorithm for intrusion detection in the IoT network of fog computing and achieved. They tested the DL algorithm on the NSL-KDD dataset and achieved very good accuracy, DR, FAR 99.2 %, 99.27 %, 0.85 %, respectively. Keserwani et al. [82] have proposed NIDS approaches based on nature-inspired algorithms for feature selection and machine learning algorithm for IoT environment. The approaches have been evaluated on the KDDCup'99, NSL-KDD, and CICIDS-2017 datasets for multiclass classification and achieved accuracies of more than 99 %.

Yu et al. [83] designed a NIDS in which they utilized a dilated convolutional autoencoders (DCAEs) as a classifier to classify network traffic as normal and malicious. In the preprocessing phase, the network traffic data were converted into two-dimensional vectors of numerical form. These data were fed to DCAE so that it may be trained in an unsupervised manner since it used unlabeled data in its training phase. They utilized the two publically available datasets; CTU-UNB and UNB-ISCX 2012.

Kang and Kang [84] proposed a DL-based approach to detect intrusions in the in-vehicle network environment, where they used DBN with eleven layers DL method for classifier and achieved the accuracy and FAR of 97.8 % and 1.6 %, respectively. Features of in-vehicle network communication were collected from the controller area network (CAN) packets.

Aminanto and Kim [85] used an Aegean WiFi Intrusion Dataset (AWID) for testing their proposed approach for detecting the impersonation attacks. In their approach, they used the unsupervised k-means clustering algorithm and stacked autoencoder to train the model with two hidden layers. They received DR and FAR 92.18 %, and 4.40 %, respectively.

Maimó et al. [86] proposed a method for anomaly detection and tested the method on the CTU dataset of botnet attacks [87]. They used DBN and RNN DL techniques as classifiers in sequence in their method. If the packet by the DBN was malicious, then the packet was inputted to RNN. The DBN with two hidden layers achieved precision, recall of 81.26 %, and 99.34 %, respectively, on the training dataset.

Lotfollahi et al. [88] proposed an approach for encrypted traffic classification with the help of an autoencoder

combined with a CNN and a classification layer. Their approach differentiated between the virtual private network (VPN) and non-VPN traffic. The approach achieved an F1 score of 98 % on the ISCX VPN-nonVPN dataset, which consists of traffic in pcap format files [89]. Wang et al. [90] perform a similar task by using 1D CNN with more than 99.5 % of precision and recall.

NIDS ensures the good security level of a network from various attacks. Various tools, approaches, and methods based on machine learning are there for detecting intrusion in a network. In the present era, various public and private industries, organizations, and companies are in a big and complex network environment to support their users, which are generating a large amount of data. To handle these big data, powerful automated ML, DL techniques are required to perform analysis on those data, thereby useful information could be interpreted from them. This level of information is very helpful, valuable, and supporting the network administrator to initiate the proper actions to tackle the different kinds of attacks or intrusions in a cost-effective manner. Since the classification techniques through ML or DL models can effectively predict the intrusion or anomaly, only proper training is required for ML, or DL classification techniques, i.e., relevant preprocessed attributes should be provided as input to ML or DL classification models. Many experimental results proved that the performance analysis of various classification algorithms is better, more efficient, and effective in real-life environments. Since they provided the correct detection rate or classification rate, producing less error and high accuracy. The researchers have utilized various classification techniques such as decision tree (DT), random forest (RF), Adaboost, ensemble method, support vector machine (SVM), artificial neural networks (ANNs), autoencoder (AE), recurrent neural network (RNN), deep neural network (DNN), restricted Boltzmann machine (RBM) are analyzed on various IDS or NIDS datasets. The different NIDS models based on feature selection and classification techniques are presented in Table 5.

Table 6 represents the abbreviation for Tables 3 and 5.

7 Proposed framework

Most of the IDS frameworks come in bundled with predefined signatures with specific methods for intrusion detection. The research community has designed and developed many method(s) specific intrusion detection frameworks. Some of them even not addressing the data imbalance issue. Wang et al. [67] have proposed an intrusion detection framework which modules are data transformation, model building, and testing. But the framework did not discuss the data imbalance issue and dependent on support vector machine (SVM) for feature augmentation. Feature augmentation is a method to expand the features artificially from the existing features. Chiba et al. [111] proposed an intrusion detection framework that uses a genetic algorithm and simulated annealing algorithm for searching the optimal values required parameters of the backpropagation neural network (BPNN), such as learning rate (LR) and momentum. BPNN is presented for the classification. The framework is using specific methods. Zeng et al. [112] have demonstrated an intrusion detection framework for designing a IDS model. The framework consists of preprocessing, DL methods such as long short-term memory (LSTM), stacked autoencoder (SAE), convolutional neural network (CNN) as the classifier. The proposed framework is not generic and not discussed the testing phase. Md Reazul et al. [113] presented methods (bayesian and wrapper) specific intrusion detection framework. Yazan et al. [114] have proposed a DL-IDS framework for IoT environment. The framework is presented for the attacks presented in the NSL-KDD dataset, but the data imbalance issue is neglected. The framework needs to be customized and adoptive to manage the organizational needs for security [115]. In this paper, based on the comprehensive survey, a generalized framework for intrusion detection, a framework depicted in Fig. 4 is designed and proposed.

A IDS model can be created using any combination of one feature selection method and one classification method that needs to be evaluated on the network dataset by following the processes presented in the framework. The developed model can be proposed in case it satisfies the threshold level of the considered metrics or parameters such as accuracy, detection rate, false alarm rate. The feature selection method may be of any type wrapper, filter, or meta-heuristic. The classification method may be any ML (DT, SVM,k-NN, RF) or DL (DBN, autoencoder, or ANN) method. Different modules presented in the framework are discussed below as:

1. *Dataset Creation* The dataset creation contains three methods - Monitoring network, data collection, and feature extraction. The specified network can be designed using many types of attack scenarios using attack simulation tools and can monitor using any monitoring tool like tcp dump. The data collection process collects the data from the monitoring tool in pcap format, and the required features can be extracted from the collected packets in the feature extraction process. As discussed earlier, despite many challenges and privacy issues, many benchmarked networking is available on the Internet to evaluate the developed NIDS model. Example of such datasets is KDD-99, NSL-KDD, UNSW-NB15, CICIDS-2017, etc., that

Table 5 Summary of the recent existing NIDS models

Ref.	Detection model	Dataset	NoF	Acc.	DR	FAR	NoC	YoP
[91]	En-ABC	NSL-KDD	–	97.62	1.05	97.59	23	2020
[92]	AE	NSL-KDD	–	87	–	–	29	2020
[93]	NSGA2-BLR+RF	NSL-KDD	19	99.65	91.92	0.18	6	2020
[94]	NSGAII-ANN and RF	KDD99	19	94.8	–	6	4	2020
[59]	CFS-BA + Voting (C4.5,RF,ForestPA)	AWID	8	99.52	99.5	0.15	17	2020
[59]	CFS-BA + Voting (C4.5,RF,ForestPA)	CICIDS-2017	13	99.89	99.9	0.12	17	2020
[95]	Firefly algorithm and C4.5	NSL-KDD	10	–	0.03	99.98	59	2019
[96]	Multivariate correlation	UNSW-NB15	3 to 20	98.65	1.26	99.74	15	2019
[97]	SDAE-IDS	KDD Cup 99	42	95	–	–	22	2019
[98]	Packet payload based on LSTM and (CNN).	CICIDS 2017	–	99.92	99.78	0.0165	0	2019
[98]	Packet payload based on LSTM and (CNN).	ISCX 2012	–	99.64	99.17	0.332	0	2019
[99]	BBA+FSFF	UNSW-NB15	18	–	99.09	0.63	23	2019
[100]	IG-PCA-Ensemble	ISCX 2012	7	99.011	99.1	0.011	68	2019
[101]	DeepWindow (MI + MIC + LSTM)	CIC-IDS2017	–	99.5	99.4	–	7	2019
[39]	Autoencoder + MI + RBFC	AWID	7	98	99.04	3	5	2019
[70]	GA-SVM	CICIDS-2017 Partial	20	–	99.85	0.0009	68	2018
[102]	Random forest	UNSW-NB15	11	75.66	–	–	3	2018
[103]	XGBoost	CIC-IDS2017	80	91.36	98.38	12	19	2018
[48]	HG-GA, SVM	NSL-KDD	35	96.72	97.14	0.83	100	2017
[104]	SVM	ISCX 2012	11	–	98.5	1.1	8	2017
[105]	RFAODE	Kyoto 2006+	15	90.51	92.38	0.14	27	2017
[56]	LR	UNSW-NB15	all	83	–	14.2	48	2017
[67]	LMDRT-SVM	NSL-KDD	–	99.31	99.2	0.6	90	2017
[106]	LSA, K-mean + MLP	NSL-KDD	25	99.37	99.42	0.66	47	2016
[107]	EMFFS	NSL-KDD	13	99.67	99.76	0.42	183	2016
[50]	TVCPSO-SVM	NSL-KDD	–	98.3	97.05	0.87	121	2016
[108]	LSSVM-FMIFS	Kyoto 2006+	4	–	97.8	0.43	308	2016
[109]	Greedy Stepwise + FONN	NSL-KDD	11	99.64	99.62	0.309	2	2016
[42]	OS-ELM	NSL-KDD	28	98.66	98.26	0.99	161	2015
[110]	NSGA2- GHSOM	NSL-KDD	25	99.12	99.6	2.24	132	2014

have been created by using the mentioned three methods.

2. *Preprocessing* The created or collected dataset is fed to the data preprocessing module to transform by using cleansing, encoding, and normalizing techniques in a format to parse in further ML processes smoothly to produce qualitative results. Details of the techniques are provided in the sections below.

   - *Data Cleansing* In this, technique of data preprocessing, incorrect, incomplete, or noisy data problems are handled by removing or updating them. It is the groundwork for accurate and effective data analysis.
   - *Encoding* In this, the string type or categorical variables are converted into the numeric form using a methods such as one hot, label, ordinal, helmert, binary, frequency, mean, weight of evidence, probability ratio, hashing, backward difference, leave one out, m-estimator, thermometer, and james-stein.
   - *Normalization* In this, the variable with different numeric domain ranges is scaled to the same domain range, usually from 0 to 1, without affecting range differences of actual values, i.e., general distribution, or without loosing the information.

In preprocessing, we deal with missing values, noise in data, strings to numeric so that further processes (feature selection and classification) became easier.

Table 6 Abbreviation for Tables 3 and 5

Short form	Full form
NoF	Number of features
YoP	Year of publication
AE	Autoencoder
NoC	Number of citations
En-ABC	An ensemble artificial Bee
RF	Random forest
NSGA2	Non-dominated sorting genetic algorithm
ANN	Artificial nural network
NA	Not available
LSTM	Long short-term memory
CNN	Convolution neural networks
BBA	Binary bat algorithm
FSFF	Feature similarity-based fitness function
EMFFS	Ensemble-based multi-filter feature selection
RFAODE	Random forest one-dependence estimator
TVCPSO	Time-varying chaos particle swarm optimization
LMDRT	Logarithm marginal density ratios transformation
CFS-BA	Correlation-based feature selection Bat algorithm
FONN	Fuzzy ownership neural network
MI-MIC	Mutual information-maximal information coefficient
SDAE	Stacked de-noising autoencoders
FSA	Feature selection algorithm
GHSOM	Growing Hierarchical self-organizing map
OS-ELM	Online sequential extreme learning machine
RBFC	Radial basis function classifier
PCA	Principal component analysis
PSO	Particle swarm optimization
ACO	Ant colony optimization
ABC	Artificial Bee Colony
AMFOA	Adaptive mutation fruit fly optimization algorithm
CFS-BA	Correlation-based feature selection-bat algorithm

3. *Feature Selection* This module is very helpful in addressing the curse of dimensionality. It reduces the time for prediction and computing costs. This module uses the knowledge base for putting up the rules to be utilized to make the system smart in the future.
4. *Classification Module* The dataset is prepared after optimization of the feature set, which selects relevant feature set. The data of selected features are divided into two subsets - training and testing.
5. *Training Phase* Here, in the training phase, the training data are resampled if the dataset belongs to multiclass problem so that the minority class data samples could get importance. In this way, the data imbalance problem is addressed in the training phase so that the attacks of minority classes can be identified during the testing phase. The training data are fed to the considered ML-based supervised classifier to built a NIDS model in the training phase.
6. *Testing Phase* The trained model is used on the testing dataset to classify the network traffic into various classes and confusion matrics are determined. The performance metrics such as accuracy, precision, recall, and f1-score are calculated from the confusion matrics to evaluate the model. The ML or DL method is used for classification so that intrusion can be identified. It returns the classification selected feature works as input for the classification method.

8 Development of a NIDS model based on proposed framework

8.1 Dataset collection

UNSW-NB15 dataset is considered to evaluate the performance of the developed NIDS model that has been designed after monitoring, data collection, and feature extraction processes in the .csv file. The entries in this dataset were created by a cybersecurity research group by using the tools tcpdump and Ixia PerfectStorm at the Australian Centre for Cyber Security (ACCS) [116]. The generated dataset contains approximately 2.5 million records representing normal and modern network traffic attacks. Bro-IDS, Argus tools, and developed algorithms were used to generate 49 features. Attack_cat and label are two labeled features. The Attack_cat contains nine types of attack labels as Worms, Shellcode, Reconnaissance, Analysis, Generic, Backdoor, DoS, Exploits, and Fuzzers and one normal. In contrast, label contains 1 to represent abnormal traffic and 0 to represent normal traffic [117].

8.2 Preprocessing

Data cleaning for null values, data conversion from string to numeric for string types of features, and standard scalar normalization have been used in the preprocessing module. The StandardScaler transforms data in such a way that the distribution contains a mean value 0 and a standard deviation of 1.

8.3 Feature selection

There are many feature selection techniques. In our proposed model for the selection of relevant feature selection, a hybrid genetic algorithm (GA) has been coded with the following properties:

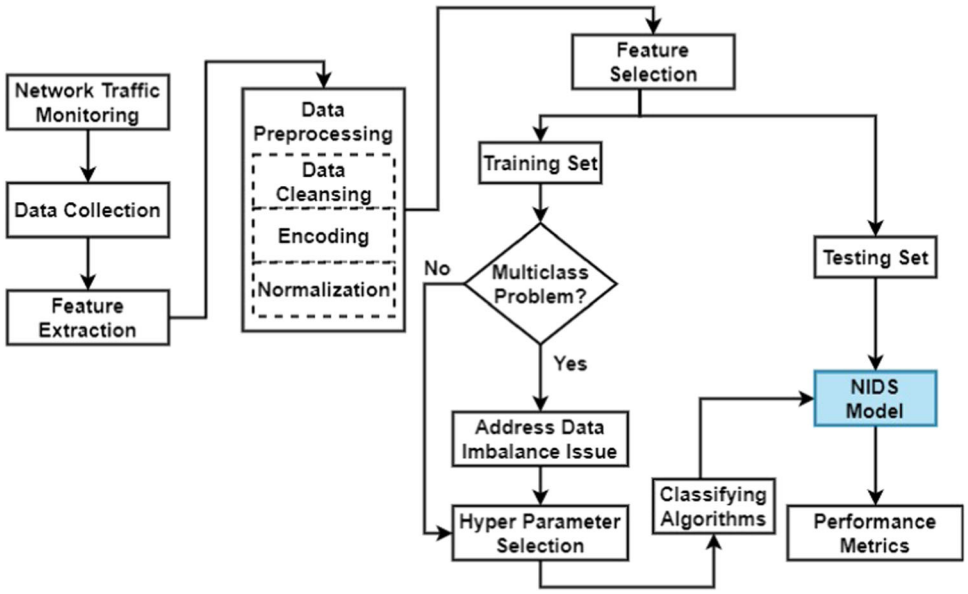

Fig. 4 Proposed generalized framework for IDS

1. It combines the searchability of GA with local search heuristics. First, it generates the candidate subsets of features randomly. With a local search, the operation is called stepwise elimination to eliminate the un-relevant features for the logistic regression algorithm from a randomly selected subset of features to overcome trapping into local minima.
2. To achieve high accuracy property of wrapper and high-efficiency property of filter techniques, it uses a wrapper in criterion to rank all candidate features in local searching which help to prune worthless candidate feature subsets of features lead to saving computation effort in the training phase of the classifier to achieve high accuracy.
3. A formula is used to calculate the mutual information (MI) between features and classes from a subset of selected candidate features to know the relevancy and redundancy among them. In local search, MI helps to rank every candidate feature.
4. MI between the predicted labels with logistic regression and the actual classes has also been used by the global search objective function of the wrapper to avoid the inconsistency among wrapper and logistic regression training. As a result, the selected features with the highest accuracy in the logistic regression classifier become the output.
5. At last the selected feature from UNSW-NB-15 is :'state', 'dur', 'dpkts', 'sbytes', 'dinpkt', 'sinpkt', 'sttl', 'djit', 'proto', 'service', 'sload', 'swin', 'sjit', 'stcpb', 'dbytes', 'spkts', 'dload', 'dloss', 'dttl', 'sloss', 'is_sm_ips_ports'.

8.4 Classification

A deep neural network (DNN) has been used for classification. The DNN learns the parameters in each hidden layer where the rectified linear unit (ReLU) activation function has been used to add non-linearity in the model. The sigmoid activation function has been used in the output layer to predict packets as normal or attack. The selected feature vector is provided as the input nodes in the DNN structure. Each node computes an output by the ReLU function. Linear combinations of the outputs are attached to the next hidden layers one after another. Four hidden layers have been used in the DNN. The combination of models in DNN has been applied to improves the performance of the classification method. Figure 5 exhibits that 21 features are provided at the input layer. Hidden layer1, hidden layer2, hidden layer3, and hidden layer4 contain 128 nodes, 64 nodes, 32 nodes, and 10 nodes. The output layer contains one node.

Deep neural networks (DNNs) consisting of a large amount of parameters are treated as very powerful classification techniques. Overfitting in such a large network reduces the performance of the classification techniques. Hence, it is a serious problem in such networks. Due to the large size of the networks, it needs many computations, which creates difficulty in addressing overfitting by combining the predictions of many large neural networks during testing. Dropout has been used in the DNN to address the mentioned problems. The dropout technique randomly drops units and their connections from the designed DNN during training. The units can be prevented from too much co-adapting from exponentially different thinned networks. It becomes easy to approximate the effect of all these

Fig. 5 Designed deep neural network (DNN)

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 128)               2688 = (21*128)+128
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 64)                8256 = (128*64)+64
_________________________________________________________________
dropout_2 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_3 (Dense)              (None, 32)                2080 = (64*32)+32
_________________________________________________________________
dropout_3 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_4 (Dense)              (None, 10)                330 = (32*10)+10
_________________________________________________________________
dropout_4 (Dropout)          (None, 10)                0
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 11 = (10*1)+1
_________________________________________________________________
activation_1 (Activation)    (None, 1)                 0
=================================================================
Total params: 13,365
Trainable params: 13,365
```

thinned network predictions with a single unthinned network containing smaller weights. Hence, the dropout technique significantly reduces overfitting for increasing the performance of the classification techniques.

8.5 Results and comparative analysis

The empirical experiments of the proposed model have been implemented in Python programming language on a laptop with an Intel Core i7-5500U CPU @ 2.40 GHz 2401 Mhz, 2 Cores 4 Logical processors with 12 GB RAM running Microsoft Windows 10 Professional. The only binary class classification has been considered to show the proof concept of the proposed framework with the help of the proposed GA-DNN model. Moreover, in order to evaluate the performance of the proposed GA-DNN model accuracy, FAR, DR, precision, f1-score, the recall has been considered.

Confusion matrix and classification report of the proposed NIDS model are demonstrated in Table 7. Some performance metrics values—accuracy, precision, recall, f1-score of the proposed NIDS model are presented in Table 8

Comparison of proposed GA-DNN model with recent existing works is presented in Table 9 and Fig. 6.

Table 7 Confusion matrix and classification report for the proposed model on UNSW-NB15 dataset

Confusion matrix		Class	Normal (0)		Attack (1)
		Normal (0)	16197		495
		Attack (1)	501		34410
	Class	Precision (%)	Recall (%)	f1-score (%)	Support
Classification report	0	97	97	97	16692
	1	99	99	99	35911
	Micro avg	98	98	98	52603
	weighted avg	98	98	98	52603

Table 8 Performance metrics values of the proposed NIDS model

Performance metrics	Value in (%)
Accuracy	98.10657
Recall	98.10657
Precision	98.10675
f1-score	98.10666

Table 9 Comparative analysis of the proposed GA-DNN NIDS model with existing approaches

Model	Accuracy (%)	FAR (%)	DR (%)
Proposed GA-DNN	98.11	1.89	97.81
MSCNN [118]	85.6	55.3	97.2
MSCNN-LSTM [118]	89.8	47.4	99.1
Integrated model [55]	84.83	2.01	90.32
DO IDS [119]	92.8	3.3	NA

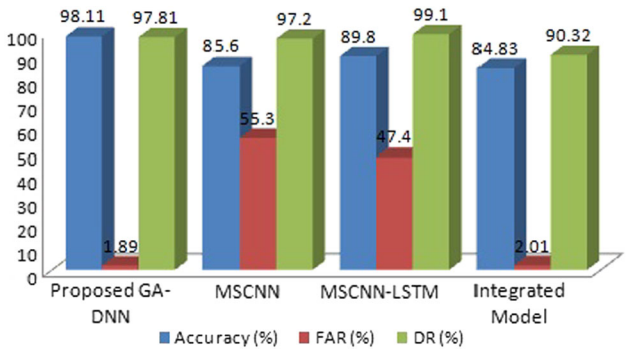

Fig. 6 Comparison of proposed GA-DNN model with recent existing works

8.6 Discussion

Table 10 demonstrates the accuracy comparison of 11 recent existing approaches on UNSW-NB15 dataset.

The proposed GA-DNN model is depicting the highest accuracy than the existing recent approaches. The proposed approach or model uses genetic algorithms (GA) for feature selection. The GA follows the natural selection principle to filter a subset from a population of possible solutions. The subset is called parents to produce the "next generation." The next generations are used to produce their successors and are continued until an optimal solution is produced. Various GA's differ in the way that how the "parents" are shortlisted and how they are used to produce the "next generation." Here, searchability of GA and local search heuristics is combined to eliminate the un-relevant features. After that, the wrapper technique is used to rank the candidate features to prune worthless candidates. The calculated mutual information (MI) between selected candidate features and classes helps to prune the redundant candidates and to get ranked features as selected features. Hence, the used GA provides a better selection of relevant feasible features, and the irrelevant features are ignored. The selected features are fed to the deep neural network (DNN), which has been designed by selecting appropriate parameters such as number of neurons in each layer, dropout rate, batch size, number of epochs to produce high accuracy in prediction. The combination of GA-based feature selection and tuned DNN for classification has reduced the computational complexity and improved the efficiency. Because of this, the proposed GA-DNN NIDS model is producing high accuracy.

9 Conclusion

This paper represents the literature review of different types of IDS and IDS techniques and a description of six popular IDS datasets. To address the curse of dimensionality feature selection methods are used for intrusion detection. Hence, the importance of feature selection techniques and their general classification has been discussed. Finding optimal features for each class label has not been discussed since very few IDS are to follow this concept. This paper also discusses the different classification approaches based on machine learning (ML) and deep learning (DL) for intrusion detection. The power of classification techniques has also been discussed. Furthermore, based on the literature review, a general framework for IDS has been proposed. At last model for NIDS has been proposed based on the proposed framework as a proof of concept. Achieved accuracy and detection rate of the proposed model on the UNSW-NB15 dataset are 98.11% and

Table 10 Accuracy comparison of proposed ISSA-MDBN model in case of binary classification on test samples of UNSW-NB15 dataset

Model name	Accuracy (%)	Model name	Accuracy (%)
Proposed GA-DNN	98.11	AE-SVM-ABC [120]	90.00
DO IDS [119]	92.80	Dendron [121]	84.33
PSI-NetVisor [122]	94.54	ENADS [117]	85.56
C5 [55]	90.74	CNN-WDLSTM [123]	97.17
TSDL [124]	89.13	Neural network [125]	86.70
NB [119]	82.10	UIDS [126]	88.92

97.81%, respectively, and achieving better performance than other approaches comparatively.

Declarations

Conflict of interest The authors declare that they have no competing or conflicting interests.

References

1. What is a cyber attack? Recent examples show disturbing trends | CSO Online. (2020) https://www.csoonline.com/article/3237324/what-is-a-cyber-attack-recent-examples-show-disturbing-trends.html. Accessed 18 May, 2020
2. India Faces \$10.3M Annual Loss From Cyberattack | PYMNTS.com. (2019) https://www.pymnts.com/news/security-and-risk/2018/microsoft-india-financial-loss-cyberattack/. Accessed on 18 May, 2020
3. Cybercrime Damages \$6 Trillion by 2021. (2019) https://cybersecurityventures.com/hackerpocalypse-cybercrime-report-2016/. Accessed on 18 May, 2020
4. Understanding the cost of a cybersecurity attack: The losses organizations face | Packt Hub. https://hub.packtpub.com/understanding-the-cost-of-a-cybersecurity-attack-the-losses-organizations-face/. Accessed on 18 May, 2020
5. Caballero J, Grier C, Kreibich C, Paxson V (2011) Measuring pay-per-install: the commoditization of malware distribution. In: Usenix Security Symposium. 13
6. Hatef MA, Shaker V, Jabbarpour MR, Jung J, Zarrabi H (2018) Hidcc: a hybrid intrusion detection approach in cloud computing. Concurr Comput Pract Exp 30(3):4171
7. KDD Cup 1999 Data. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. Accessed on 24 May, 2020
8. NSL-KDD | Datasets | Research | Canadian Institute for Cybersecurity | UNB. https://www.unb.ca/cic/datasets/nsl.html. Accessed on 24 May, 2020
9. Description: wireless security datasets project. http://icsdweb.aegean.gr/awid/features.html. Accessed on 24 May, 2020
10. IDS 2012 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. https://www.unb.ca/cic/datasets/ids.html. Accessed on 24 May, 2020
11. The UNSW-NB15 data set description. https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/. Accessed on 24 May, 2020
12. Panigrahi R, Borah S (2018) A detailed analysis of cicids2017 dataset for designing intrusion detection systems. Int J Eng Technol 7(3.24):479–482
13. Hoque N, Bhattacharyya DK, Kalita JK (2014) Mifs-nd: a mutual information-based feature selection method. Exp Syst Appl 41(14):6371–6385
14. Liao H-J, Lin C-HR, Lin Y-C, Tung K-Y (2013) Intrusion detection system: a comprehensive review. J Netw Comput Appl 36(1):16–24
15. Vasilomanolakis E, Karuppayah S, Mühlhäuser M, Fischer M (2015) Taxonomy and survey of collaborative intrusion detection. ACM Comput Surv 47(4):1–33
16. Hu J, Yu X, Qiu D, Chen H-H (2009) A simple and efficient hidden markov model scheme for host-based anomaly intrusion detection. IEEE Netw 23(1):42–47
17. Creech G, Hu J (2013) A semantic approach to host-based intrusion detection systems using contiguousand discontiguous system call patterns. IEEE Trans Comput 63(4):807–819
18. Yeung D-Y, Ding Y (2003) Host-based intrusion detection using dynamic and static behavioral models. Pattern Recognit 36(1):229–243
19. Sperotto A, Schaffrath G, Sadre R, Morariu C, Pras A, Stiller B (2010) An overview of ip flow-based intrusion detection. IEEE Commun Surv. Tutor 12(3):343–356
20. Mohan R, Vaidehi V, Mahalakshmi M, Chakkaravarthy SS et al (2015) Complex event processing based hybrid intrusion detection system. In: 2015 3rd international conference on signal processing, communication and networking (ICSCN), pp. 1–6
21. Suricata | Open Source IDS / IPS / NSM engine. https://suricata-ids.org/. Accessed on 18 May, 2020
22. Roesch M et al (1999) Snort: lightweight intrusion detection for networks. Lisa 99:229–238
23. Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Comput Surv 41(3):1–58
24. Kim G, Lee S, Kim S (2014) A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. Exp Syst Appl 41(4):1690–1700
25. Cateni S, Colla V, Vannucci M (2017) A fuzzy system for combining filter features selection methods. Int J Fuzzy Syst 19(4):1168–1180
26. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3(Mar):1157–1182
27. Bermejo P, Gámez JA, Puerta JM (2011) A grasp algorithm for fast hybrid (filter-wrapper) feature subset selection in high-dimensional datasets. Pattern Recognit Lett 32(5):701–711
28. Esposito F, Malerba D, Semeraro G, Kay J (1997) A comparative analysis of methods for pruning decision trees. IEEE Trans Pattern Anal Mach Intell 19(5):476–491
29. Visalakshi S, Radha V (2017) A hybrid filter and wrapper feature selection approach for detecting contamination in drinking water management system. J Eng Sci Technol 12(7):1819–1832
30. Chandrashekar G, Sahin F (2014) A survey on feature selection methods. Comput Electr Eng 40(1):16–28
31. BoussaïD I, Lepagnot J, Siarry P (2013) A survey on optimization metaheuristics. Inf Sci 237:82–117
32. Alweshah M, Abdullah S (2015) Hybridizing firefly algorithms with a probabilistic neural network for solving classification problems. Appl Soft Comput 35:513–524
33. Balasaraswathi VR, Sugumaran M, Hamid Y (2017) Feature selection techniques for intrusion detection using non-bio-inspired and bio-inspired optimization algorithms. J Commun Inf Netw 2(4):107–119
34. Hajisalem V, Babaie S (2018) A hybrid intrusion detection system based on abc-afs algorithm for misuse and anomaly detection. Comput Netw 136:37–50
35. Srivastava MS, Joshi MN, Gaur M (2014) A review paper on feature selection methodologies and their applications. IJCSNS 14(5):78
36. Aminanto ME, Tanuwidjaja H, Yoo PD, Kim K (2017) Weighted feature selection techniques for detecting impersonation attack in wi-fi networks. In: Proc. Symp. Cryptogr. Inf. Secur.(SCIS), pp. 1–8
37. 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 Forens Secur 13(3):621–636
38. Abdulhammed R, Faezipour M, Abuzneid A, Alessa A (2018) Effective features selection and machine learning classifiers for improved wireless intrusion detection. In: 2018 International symposium on networks, computers and communications (ISNCC), pp. 1–6

39. Parker LR, Yoo PD, Asyhari TA, Chermak L, Jhi Y, Taha K (2019) Demise: interpretable deep extraction and mutual information selection techniques for iot intrusion detection. In: Proceedings of the 14th international conference on availability, reliability and security, pp. 1–10
40. Thanthrige USKPM, Samarabandu J, Wang X (2016) Machine learning techniques for intrusion detection on public dataset. In: 2016 IEEE Canadian conference on electrical and computer engineering (CCECE), pp. 1–4
41. De la Hoz E, De La Hoz E, Ortiz A, Ortega J, Prieto B (2015) Pca filtering and probabilistic som for network intrusion detection. Neurocomputing 164:71–81
42. Singh R, Kumar H, Singla R (2015) An intrusion detection system using network traffic profiling and online sequential extreme learning machine. Exp Syst Appl 42(22):8609–8624
43. Wahba Y, ElSalamouny E, ElTaweel G (2015) Improving the performance of multi-class intrusion detection systems using feature reduction. arXiv preprint arXiv:1507.06692
44. Iglesias F, Zseby T (2015) Analysis of network traffic features for anomaly detection. Mach Learn 101(1–3):59–84
45. Hakim L, Fatma R et al (2019) Influence analysis of feature selection to network intrusion detection system performance using nsl-kdd dataset. In: 2019 International conference on computer science, information technology, and electrical engineering (ICOMITEE), pp. 217–220
46. Khorram T, Baykan NA (2018) Feature selection in network intrusion detection using metaheuristic algorithms. Int J Adv Res Ideas Innov Technol 4(4):704
47. Amiri F, Yousefi MR, Lucas C, Shakery A, Yazdani N (2011) Mutual information-based feature selection for intrusion detection systems. J Netw Comput Appl 34(4):1184–1199
48. Raman MG, Somu N, Kirthivasan K, Liscano R, Sriram VS (2017) An efficient intrusion detection system based on hypergraph-genetic algorithm for parameter optimization and feature selection in support vector machine. Knowledge-Based Syst 134:1–12
49. Wang W, Liu X (2015) Melt index prediction by least squares support vector machines with an adaptive mutation fruit fly optimization algorithm. Chemom Intell Lab Syst 141:79–87
50. 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
51. Alazzam H, Sharieh A, Sabri KE (2020) A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer. Exp Syst Appl 148:113249
52. Mohammadi S, Mirvaziri H, Ghazizadeh-Ahsaee M, Karimipour H (2019) Cyber intrusion detection by combined feature selection algorithm. J Inf Secur Appl 44:80–88
53. Tama BA, Comuzzi M, Rhee K-H (2019) Tse-ids: a two-stage classifier ensemble for intelligent anomaly-based intrusion detection system. IEEE Access 7:94497–94507
54. Aljawarneh S, Aldwairi M, Yassein MB (2018) Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. J Comput Sci 25:152–160
55. Kumar V, Sinha D, Das AK, Pandey SC, Goswami RT (2019) An integrated rule based intrusion detection system: analysis on unsw-nb15 data set and the real time online dataset. Clust Comput 23:1397
56. Moustafa N, Slay J (2017) A hybrid feature selection for network intrusion detection systems: central points. arXiv preprint arXiv:1707.05505
57. Elmasry W, Akbulut A, Zaim AH (2020) Evolving deep learning architectures for network intrusion detection using a double pso metaheuristic. Comput Netw 168:107042
58. Naidoo T, McDonald A, Tapamo J-R (2015) Feature selection for anomaly-based network intrusion detection using cluster validity indices (2015)
59. Zhou Y, Cheng G, Jiang S, Dai M (2020) Building an efficient intrusion detection system based on feature selection and ensemble classifier. Comput Netw 174:107247
60. Namratha M, Prajwala T (2012) A comprehensive overview of clustering algorithms in pattern recognition. IOR J Comput Eng 4(6):23–30
61. Koturwar P, Girase S, Mukhopadhyay D (2015) A survey of classification techniques in the area of big data. arXiv preprint arXiv:1503.07477
62. Caruana R, Niculescu-Mizil A (2006) An empirical comparison of supervised learning algorithms. In: Proceedings of the 23rd international conference on machine learning, pp. 161–168
63. Deng L (2014) A tutorial survey of architectures, algorithms, and applications for deep learning. APSIPA Trans Sig Inf Process 3:e2
64. Guo Y, Liu Y, Oerlemans A, Lao S, Wu S, Lew MS (2016) Deep learning for visual understanding: a review. Neurocomputing 187:27–48
65. Pouyanfar S, Sadiq S, Yan Y, Tian H, Tao Y, Reyes MP, Shyu M-L, Chen S-C, Iyengar S (2018) A survey on deep learning: algorithms, techniques, and applications. ACM Comput Surv 51(5):1–36
66. Weston J, Ratle F, Mobahi H, Collobert R (2012) Deep learning via semi-supervised embedding. Neural Netw Tricks Trade 7700:639–655
67. Wang H, Gu J, Wang S (2017) An effective intrusion detection framework based on svm with feature augmentation. Knowledge-Based Syst 136:130–139
68. George A, Vidyapeetham A (2012) Anomaly detection based on machine learning: dimensionality reduction using pca and classification using svm. Int J Comput Appl 47(21):5–8
69. Hamamoto AH, Carvalho LF, Sampaio LDH, Abrão T, Proença ML Jr (2018) Network anomaly detection system using genetic algorithm and fuzzy logic. Exp Syst Appl 92:390–402
70. Vijayanand R, Devaraj D, Kannapiran B (2018) Intrusion detection system for wireless mesh network using multiple support vector machine classifiers with genetic-algorithm-based feature selection. Comput Secur 77:304–314
71. Kuang F, Xu W, Zhang S (2014) A novel hybrid kpca and svm with ga model for intrusion detection. Appl Soft Comput 18:178–184
72. Bamakan SMH, Wang H, Shi Y (2017) Ramp loss k-support vector classification-regression; a robust and sparse multi-class approach to the intrusion detection problem. Knowledge-Based Syst 126:113–126
73. Viegas EK, Santin AO, Oliveira LS (2017) Toward a reliable anomaly-based intrusion detection in real-world environments. Comput Netw 127:200–216
74. Gao N, Gao L, Gao Q, Wang H (2014) An intrusion detection model based on deep belief networks. In: 2014 Second international conference on advanced cloud and big data. pp. 247–252
75. Nguyen KK, Hoang DT, Niyato D, Wang P, Nguyen D, Dutkiewicz E (2018) Cyberattack detection in mobile cloud computing: a deep learning approach. In: 2018 IEEE wireless communications and networking conference (WCNC), pp. 1–6
76. 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
77. Alom MZ, Taha TM (2017) Network intrusion detection for cyber security using unsupervised deep learning approaches. In: 2017 IEEE national aerospace and electronics conference (NAECON), pp. 63–69

78. Sharma YK, Rokade Monika D (2019) Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic. IOSR J Eng (IOSR JEN) 63–67
79. Wang W, Zhu M, Zeng X, Ye X, Sheng Y (2017) Malware traffic classification using convolutional neural network for representation learning. In: 2017 international conference on information networking (ICOIN), pp. 712–717
80. 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
81. Abeshu A, Chilamkurti N (2018) Deep learning: the frontier for distributed attack detection in fog-to-things computing. IEEE Commun Mag 56(2):169–175
82. Keserwani PK, Govil MC, Pilli ES, Govil P (2021) A smart anomaly-based intrusion detection system for the internet of things (iot) network using gwo-pso-rf model. J Reliab Intell Environ 7(1):3–21
83. Yu Y, Long J, Cai Z (2017) Network intrusion detection through stacking dilated convolutional autoencoders. Secur Commu Netw. https://doi.org/10.1155/2017/4184196
84. Kang M-J, Kang J-W (2016) Intrusion detection system using deep neural network for in-vehicle network security. PloS one 11(6):e0155781
85. Aminanto ME, Kim K (2017) Improving detection of wi-fi impersonation by fully unsupervised deep learning. Int Workshop Inf Secur Appl 10763:212–223
86. Maimó LF, Gómez ÁLP, Clemente FJG, Pérez MG, Pérez GM (2018) A self-adaptive deep learning-based system for anomaly detection in 5g networks. IEEE Access 6:7700–7712
87. Garcia S, Grill M, Stiborek J, Zunino A (2014) An empirical comparison of botnet detection methods. Comput Secur 45:100–123
88. Lotfollahi M, Siavoshani MJ, Zade RSH, Saberian M (2020) Deep packet: a novel approach for encrypted traffic classification using deep learning. Soft Comput 24(3):1999–2012
89. Draper-Gil G, Lashkari AH, Mamun MSI, Ghorbani AA (2016) Characterization of encrypted and vpn traffic using time-related. In: Proceedings of the 2nd international conference on information systems security and privacy (ICISSP), pp. 407–414
90. Wang W, Zhu M, Wang J, Zeng X, Yang Z (2017) End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In: 2017 IEEE international conference on intelligence and security informatics (ISI), pp. 43–48
91. Garg S, Kaur K, Batra S, Aujla GS, Morgan G, Kumar N, Zomaya AY, Ranjan R (2020) En-abc: an ensemble artificial bee colony based anomaly detection scheme for cloud environment. J Parallel Distrib Comput 135:219–233
92. Ieracitano C, Adeel A, Morabito FC, Hussain A (2020) A novel statistical analysis and autoencoder driven intelligent intrusion detection approach. Neurocomputing 387:51–62
93. Khammassi C, Krichen S (2020) A nsga2-lr wrapper approach for feature selection in network intrusion detection. Comput Net 172:107183
94. Golrang A, Golrang AM, Yayilgan SY, Elezaj O (2020) A novel hybrid ids based on modified nsgaii-ann and random forest. Electronics 9(4):577
95. Selvakumar B, Muneeswaran K (2019) Firefly algorithm based feature selection for network intrusion detection. Comput Secur 81:148–155
96. Gottwalt F, Chang E, Dillon T (2019) Corrcorr: a feature selection method for multivariate correlation network anomaly detection techniques. Comput Secur 83:234–245
97. Abusitta A, Bellaiche M, Dagenais M, Halabi T (2019) A deep learning approach for proactive multi-cloud cooperative intrusion detection system. Futur Gener Comput Syst 98:308–318
98. Liu J, Song X, Zhou Y, Peng X, Zhang Y, Liu P, Wu D (2019) Deep anomaly detection in packet payload. arXiv preprint arXiv:1912.02549
99. Patil R, Dudeja H, Modi C (2019) Designing an efficient security framework for detecting intrusions in virtual network of cloud computing. Comput Secur 85:402–422
100. Salo F, Nassif AB, Essex A (2019) Dimensionality reduction with ig-pca and ensemble classifier for network intrusion detection. Comput Netw 148:164–175
101. Shi Z, Li J, Wu C, Li J (2019) Deepwindow: an efficient method for online network traffic anomaly detection. In: 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), pp. 2403–2408 (2019). IEEE
102. Khan NM, Negi A, Thaseen IS et al (2018) Analysis on improving the performance of machine learning models using feature selection technique. In: international conference on intelligent systems design and applications, pp. 69–77
103. Bansal A, Kaur S (2018) Extreme gradient boosting based tuning for classification in intrusion detection systems. In: International conference on advances in computing and data sciences, pp. 372–380
104. Huang H, Khalid RS, Yu H (2017) Distributed machine learning on smart-gateway network towards real-time indoor data analytics. Data Sci Big Data Environ Comput Intell 24:231–263
105. Jabbar M, Aluvalu R et al (2017) Rfaode: a novel ensemble intrusion detection system. Proc Comput Sci 115:226–234
106. Kang S-H, Kim KJ (2016) A feature selection approach to find optimal feature subsets for the network intrusion detection system. Clus Comput 19(1):325–333
107. Osanaiye O, Cai H, Choo KKR, Dehghantanha A, Xu Z, Dlodlo M (2016) Ensemble-based multi-filter feature selection method for ddos detection in cloud computing. EURASIP J Wireless Commun Netw 1:130
108. Ambusaidi MA, He X, Nanda P, Tan Z (2016) Building an intrusion detection system using a filter-based feature selection algorithm. IEEE Trans Comput 65(10):2986–2998
109. Panigrah A, Patra MR (2016) Fuzzy rough classification models for network intrusion detection. Trans Mach Learn Artif Intell 4(2):07
110. De la Hoz E, De La Hoz E, Ortiz A, Ortega J, Martínez-Álvarez A (2014) Feature selection by multi-objective optimisation: application to network anomaly detection by hierarchical self-organising maps. Knowledge-Based Syst 71:322–338
111. Chiba Z, Abghour N, Moussaid K, El Omri A, Rida M (2018) Novel framework based on genetic algorithm and simulated annealing algorithm for optimization of bp neural network applied to network ids. In: proceedings of the 3rd international conference on smart city applications, pp. 1–9
112. Ahmad I, Basheri M, Iqbal MJ, Rahim A (2018) Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. IEEE Access 6:33789–33795
113. Kabir MR, Onik AR, Samad T (2017) A network intrusion detection framework based on bayesian network using wrapper approach. Int J Comput Appl 166(4):13–17
114. Otoum Y, Liu D, Nayak A (2019) Dl-ids: a deep learning-based intrusion detection framework for securing iot. Trans Emerg Telecommun Technol 29:e3803
115. Bhattacharya S, Maddikunta PKR, Kaluri R, Singh S, Gadekallu TR, Alazab M, Tariq U et al (2020) A novel pca-firefly based xgboost classification model for intrusion detection in networks using gpu. Electronics 9(2):219
116. Moustafa N, Slay J (2015) Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network

data set). Mil Commun Inf Syst Conf. https://doi.org/10.1109/MilCIS.2015.7348942
117. Moustafa N, Slay J (2016) The evaluation of network anomaly detection systems: statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set. Inf Secur J Glob Persp 25(1–3):18–31
118. Zhang J, Ling Y, Fu X, Yang X, Xiong G, Zhang R (2020) Model of the intrusion detection system based on the integration of spatial-temporal features. Comput Secur 89:101681
119. Ren J, Guo J, Qian W, Yuan H, Hao X, Jingjing H (2019) Building an effective intrusion detection system by using hybrid data optimization based on machine learning algorithms. Secur Commun Netw. https://doi.org/10.1155/2019/7130868
120. Tian Q, Li J, Liu H (2019) A method for guaranteeing wireless communication based on a combination of deep and shallow learning. IEEE Access 7:38688–38695
121. Papamartzivanos D, Mármol FG, Kambourakis G (2018) Dendron: genetic trees driven rule induction for network intrusion detection systems. Futur Gener Comput Syst 79:558–574
122. Mishra P, Pilli ES, Varadharajan V, Tupakula U (2017) Psinetvisor: program semantic aware intrusion detection at network and hypervisor layer in cloud. J Intell Fuzzy Syst 32(4):2909–2921
123. Hassan MM, Gumaei A, Alsanad A, Alrubaian M, Fortino G (2020) A hybrid deep learning model for efficient intrusion detection in big data environment. Inf Sci 513:386–396
124. Khan FA, Gumaei A, Derhab A, Hussain A (2019) A novel two-stage deep learning model for efficient network intrusion detection. IEEE Access 7:30373–30385
125. Hodo E, Bellekens X, Hamilton A, Dubouilh P.-L, Iorkyase E, Tachtatzis C, Atkinson R (2016) Threat analysis of iot networks using artificial neural network intrusion detection system. In: 2016 International Symposium on Networks, Computers and Communications (ISNCC), pp. 1–6
126. Kumar V, Das AK, Sinha D (2019) Uids: a unified intrusion detection system for iot environment. Evolution Intell 1–13