Artificial Intelligence Review (2022) 55:453–563
https://doi.org/10.1007/s10462-021-10037-9

A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions

Ankit Thakkar[1] · Ritika Lohiya[1]

Published online: 1 July 2021
© The Author(s), under exclusive licence to Springer Nature B.V. 2021

Abstract

With the increase in the usage of the Internet, a large amount of information is exchanged between different communicating devices. The data should be communicated securely between the communicating devices and therefore, network security is one of the dominant research areas for the current network scenario. Intrusion detection systems (IDSs) are therefore widely used along with other security mechanisms such as firewall and access control. Many research ideas have been proposed pertaining to the IDS using machine learning (ML) techniques, deep learning (DL) techniques, and swarm and evolutionary algorithms (SWEVO). These methods have been tested on the datasets such as DARPA, KDD CUP 99, and NSL-KDD using network features to classify attack types. This paper surveys the intrusion detection problem by considering algorithms from areas such as ML, DL, and SWEVO. The survey is a representative research work carried out in the field of IDS from the year 2008 to 2020. The paper focuses on the methods that have incorporated feature selection in their models for performance evaluation. The paper also discusses the different datasets of IDS and a detailed description of recent dataset CIC IDS-2017. The paper presents applications of IDS with challenges and potential future research directions. The study presented, can serve as a pedestal for research communities and novice researchers in the field of network security for understanding and developing efficient IDS models.

Keywords Intrusion detection system · Machine learning · Deep learning · Swarm and evolutionary computation · Feature selection · Performance measures · Challenges · Future research directions and applications

Abbreviations

DNN	Deep Neural Network
DBN	Deep Belief Network
SDN	Software Defined Network

✉ Ritika Lohiya
18ftphde30@nirmauni.ac.in

Ankit Thakkar
ankit.thakkar@nirmauni.ac.in

[1] Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad, Gujarat, India

HAST-IDS	Hierarchical Spatial-Temporal features-based Intrusion Detection System
LSTM	Long Short Term Memory
CAIDA	Center for Applied Internet Data Analysis
IRC	Internet Relay Chat
MAE	Mean Absolute Error
MSE	Mean Squared Error
ARI	Adjusted Rand Index
MI	Mutual Information
AMI	Adjusted Mutual Information
NMI	Normalized Mutual Information
FMI	Fowlkes Mallows scores
SMS	Short Message Service
CSAIL	Computer Science and Artificial Intelligence Laboratory
DDoS	Distributed Denail of Service
R2L	Remote to Local
RFE	Recursive Feature Elimination
U2R	User to Root
DoS	Denial of Service
ELM	Extreme Learning Machine
FP	False Positive
TP	True Positive
FN	False Negative
TN	True Negative
NSA	Negative Selection Algorithm
AIS	Artificial Immune System
SNS	Self-NonSelf
RSKFCM	Robust Spatial Kernel Fuzzy C-Means
EM	Expectation Maximization
SMO	Sequential Minimal Optimization
PAM	Partition Around Mediods
CLARA	Clustering Large Applications
LMDRT	Logarithmic Marginal Density Ration Transformation
DT	Decision Tree
IG	Information Gain
SVM	Support Vector Machine
FAR	False Alram Rate
DR	Detection Rate
ROC	Receiver Operating Characteristics
MLP	Multi-Layer Percepton
ANN	Artificial Neural Network
PCA	Principal Component Analysis
CC	Correlation Coefficient
SA	Simulated Annealing
CFS	Correlation Feature Selection
FVBRM	Feature Vitality Based Reduction Method
KDD	Knowledge Discovery Database
LS-SVM	Least Square-Support Vector Machine
EMFFS	Ensemble Multi-Filter Feature Selection
RBF	Radial Basis Function

TPR	True Positive Rate
FPR	False Positive Rate
MIFS	Mutual Information Feature Selection
FMIFS	Flexible Mutual Information based Feature Selection
FLCFS	Flexible Linear Correlation based Feature Selection
TASVM	Triangle Area Support Vector Machine
k-NN	k Nearest Neighbour
BPN	Back Propogation Network
NB	Naïve Bayes
KMC	K-Means Clustering
CANN	Cluster Centers and Nearest Neighbours
SOM	Self Organizing Maps
MOPF	Modified Optimum Path Forest
DPC	Density Peak Clustering
RNN	Recurrent Neural Network
CPE	Cost Per Example
RBNN	Radial Basis Neural Network
FFNN	Feed Forward Neural Network
GRNN	Generalized Regression Neural Network
PNN	Probabilistic Neural Network
GRU-RNN	Gated Recurrent Unit-Recurrent Neural Network
RBM	Restricted Boltzmann Machine
PE	Portable Executable
CNN	Convolutional Neural Network
ISCX	Information Security Center of Excellence
GA	Genetic Algorithm
PSO	Particle Swarm Optimization
GNP	Genetic Network Programming
LR	Logistic Regression
HG-GA	HyperGraph-Genetic Algorithm
GSA	Genetic Selection Algorithm
ER	Error Rate
CFO	Cuttle Fish Optimization
AR	Accuracy Rate
KPCA	Kernel-Principal Component Analysis
CART	Classification and Regression Tree
BA	Bat Algorithm
ACO	Ant Colony Optimization
LUS	Local Unimodal Sampling
WMA	Weighted Majority Algorithm
WMN	Wireless Mesh Network
IDS	Intrusion Detection System
ML	Machine Learning
DL	Deep Learning
MLDL	Machine Learning and Deep Learning
SWEVO	Swarm and Evolutionary Algorithm
NIST	National Institute of Standards and Technology
OWASP	Open Web Application Security Project
IoT	Internet of Things

BoT-IoT	Botnet-Internet of Things
ELM	Extreme Learning Machine

1 Introduction

Evolution of different types of network attacks and an increase in the data exchange between the computing devices pose the requirement to secure network and computing devices. It is believed that the global IP traffic would increase threefold by the year 2022, which would account for 26% increase in the annual growth (Index 2017). The network traffic collected per capita is 16 GB per month as recorded in the year 2017 which is expected to rise to 50 GB by the year 2022 (Index 2017). The tools used in the network to exchange information, to collect data, or to monitor associated activities increase the attack surface. This put forth requirements to build an efficient IDS system that can protect the network against the security attacks.

The intruders are persistently searching for new targets and rectify the network tools to breach the security of cyber defense systems. As per the technological review conducted by Massachusetts Institute of Technology (MIT) in 2018 (Knight 2018), the big risks pertaining to network security that should be taken into consideration are compromising data where people's personal information is the main target, ransomware in cloud computing businesses (Grzonka et al. 2018), rearming artificial intelligence, attacks on IoT systems, mining cryptocurrencies, and hacking poll booths (Knight 2018). The organization such as NIST has designed the security framework to ensure flexibility in handling the effect on physical, network, and individual's resources due to network intrusions (Barrett 2018). The framework designed by NIST focuses on ensuring security to data, cyber-physical systems, industrial control systems, and Internet of Things (IoT). Moreover, OWASP also lists out the vulnerabilities of web applications that help the startup web companies to be aware of major security flaws (Søhoel et al. 2018).

Over the years, IDSs have been evolved to ensure that network traffic is legitimate and not malicious. The next-generation firewall is developed for addressing the complexity of network attacks. It integrated IDS module to detect the intrusion based on signatures, behavioral analysis, and malicious activities. The term IDS can be coined as the system which generates alarms on any anomalous behaviors in the network environment. The primary task of IDS is to protect a system by applying various combinations of processes to prevent any intruders or malicious activities that compromise system security. More often, a security expert can take necessary actions to mitigate the damage caused by the intrusions.

An intrusion can be of any form. For instance, an intruder can take over unauthorized access to a user's account by stealing the user's password, masquerading, eavesdropping, or injecting malicious code. A system can be harmed by the people who are inside the network by exploiting the vulnerabilities of software application and/or the server that might compromise the system. Many tools and services like firewalls, password encryption, access control, and intrusion prevention systems have been employed to protect the network against any threat. However, the list of threats on the network and computer is endless and is continuously evolving; therefore, intrusion detection remains an active area of research.

The conceptualization of IDS with technological perspective presented in Vidal et al. (2020) and Gupta et al. (2016a) along with the types and properties of IDS are explained.

Statistical techniques to detect anomalies in the network traffic are discussed in Hodge and Austin (2004) and Niu et al. (2011). These methods were implemented to detect outliers which define the isolation in the observed data. In Chandala et al. (2009), different forms of anomalies were studied that include system exploits, digression from normal activity, novel patterns for zero-day attacks, and eccentricity of observations. IDS have been explored for different techniques such as Machine Learning and Deep Learning (MLDL) (Hamed et al. 2018; Hodo et al. 2017; Thakkar and Lohiya 2021b). These methods are used for evaluating the performance of IDS and also described the different aspects and properties of IDS such as identification and mitigation of intrusions, placement of IDS sensors in the network, and techniques used for performance evaluation of IDS (Sabahi and Movaghar 2008).

Studying different research areas where Machine Learning (ML) and IDS have been used, was the main focus in Chandala et al. (2009). The research work (Javaid et al. 2016; Sangkatsanee et al. 2011; Peddabachigari et al. 2007) have considered the full feature set for classifying attacks using MLDL, and data mining techniques with the datasets such as DARPA (Brown et al. 2009; McHugh 2000) and KDD CUP 99 (Tavallaee et al. 2009). These dataset used for the evaluation of the IDS are quite conventional and consists of redundant and irrelevant information. The solution for the issues of redundancy and irrelevancy of features are provided using feature selection techniques in John et al. (1994). Feature selection has become obligatory in real network scenarios because of the emergence of high dimensional network data. An IDS is capable of handling a large amount of data, but the presence of irrelevant and redundant data might deteriorate the performance of the IDS (Sung and Mukkamala 2004). Feature selection techniques such as filter, wrapper and embedded have been used with different MLDL and data mining methods to increase the classification accuracy.

MLDL techniques based IDS applications are explored in our paper. The paper also discusses feature engineering methods that were used to enhance the performance of the underlying MLDL technique. The performance of the MLDL techniques was measured using the performance metrics that provides basis for deriving the criteria for evaluation and developing heuristics for constructing IDS models. The characteristics of the performance metrics help to find the consistency relationship between the identical performance measures to know which performance measure is better than the other. This empirical and theoretical analysis of the performance measures enhances the robustness of the learning algorithms. The credibility and performance of IDS can be applied to a diversity of applications such as surveillance of activities in military camps to defend against enemies, to detect frauds in credit card scam and network intrusions for cybersecurity.

For instance, network traffic is captured and analyzed for detecting any nonconforming patterns to know whether the system is compromised or not. Therefore, an IDS can be used in a variety of fields such as system security, software security, security against cybercrimes, and securing the Internet of Things (IoT) that are discussed in brief in this paper. The survey focuses on the applicability of ML, Deep Learning (DL), and Swarm and Evolutionary Algorithms (SWEVO) along with feature selection techniques from the IDS perspective for the duration 2008 to 2020. Performance measures for evaluating IDS models and applications of IDS are also discussed.

1.1 Prior survey in intrusion detection system

IDS is an important and dynamic research domain. A study related to intrusions considering various sources such as intrusion detection from the sequence of system calls is examined in Canzanese et al. (2015) while intrusion detection based on the communication channel is presented in Ampah et al. (2011). A survey on the knowledge of anomaly-based IDS is presented in Chandala et al. (2009). A review of techniques for Internet traffic analysis and flow-based characteristics of the network traffic is studied in Callado et al. (2009) and (Sperotto et al. 2010), respectively. Network anomaly detection methods such as statistical techniques based anomaly detection, classifier based anomaly detection, ML-based anomaly detection, and finite-state machine-based anomaly detection are discussed in Zhang et al. (2009).

The techniques for measuring the performance of IDS based on the placement of sensors in the network and its ability to detect and prevent the attacks is studied in Sabahi and Movaghar (2008). A survey on ML and DL techniques implemented on network intrusion datasets such as DARPA (Brown et al. 2009; McHugh 2000) and KDD CUP 99 (Tavallaee et al. 2009) is presented in Xin et al. (2018). The use of ML and data mining techniques such as Decision Tree (DT), Support Vector Machines (SVM), Fuzzy-Association Mining, Genetic Algorithms (GAs), and Bayesian Networks for IDS is presented in Buczak and Guven (2016) for a time span of 6 years (2009–2014) and considered the papers that have implemented for anomaly-based and misuse-based IDS. The paper discussed DARPA and KDD CUP 99 public datasets but feature engineering aspect was not covered in detail.

Network anomaly detection methods, tools, and datasets have been presented in Bhuyan et al. (2014). The paper discussed a variety of methods and datasets for network anomaly detection including selection strategy based feature selection methods. A list of different tools for capturing network traffic is also presented in Bhuyan et al. (2014). An analysis of the methods pertaining to the four domains namely, classification techniques, statistical techniques, information theory, and clustering techniques are presented in Ahmed et al. (2016). A study of ML-based IDS is carried out in Hamed et al. (2018). The paper also discussed the datasets used for evaluation of IDS performance. In Nguyen and Armitage (2008), IP traffic classification has been taken into consideration. The paper categorized and reviewed the work on the basis of the ML techniques implemented as traffic classifiers to classify IP traffic for IDS. The authors have surveyed papers for the time duration of 3 years (2004–2007) with the focus on flow level internet data. A survey of unsupervised techniques for hybrid IDS is presented in Nisioti et al. (2018). The paper discusses different IDS techniques as well as the need for correlation and attribution for detecting attacks in IDS. A comparison of the surveys studied is listed in Table 1. The major contributions of the paper can be summarized as follows.

- The paper presents a combined and prototypical research work performed in the field of intrusion detection from the year 2008–2020.
- In contradiction to the studies conducted, our survey focuses on discussing the importance of feature engineering that could lead to better performance of IDS in detecting various anomalies. The paper provides a taxonomy of ML, DL, and SWEVO algorithms for building IDS models alongwith feature selection methods to improve the performance of techniques.

Table 1 Comparison of the surveys studied

Topics covered		Canzanese et al. (2015)	Ampah et al. (2011)	Chandala et al. (2009)	Callado et al. (2009)	Sperotto et al. (2010)	Zhang et al. (2009)	Sabahi and Movaghar (2008)	Xin et al. (2018)	Buczak and Guven (2016)	Bhuyan et al. (2014)	Ahmed et al. (2016)	Nguyen and Armitage (2008)	Hamed et al. (2018)	Nisioti et al. (2018)	Our survey
Methods	ML-based classification	✓	✓	✓	✓	✓	✓	✓		✓	✓		✓	✓		✓
	ML-based clustering	✓	✓	✓		✓	✓		✓		✓		✓	✓	✓	✓
	Deep learning	✓	✓	✓		✓	✓		✓	✓	✓		✓	✓		✓
	Swarm and evolutionary algorithms															✓
Intrusion detection system	Host and Network IDS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Anomaly and misuse	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 1 (continued)

Topics covered		Canzanese et al. (2015)	Ampah et al. (2011)	Chandala et al. (2009)	Callado et al. (2009)	Sperotto et al. (2010)	Zhang et al. (2009)	Sabahi and Movaghar (2008)	Xin et al. (2018)	Buczak and Guven (2016)	Bhuyan et al. (2014)	Ahmed et al. (2016)	Nguyen and Armitage (2008)	Hamed et al. (2018)	Nisioti et al. (2018)	Our survey
Feature engineering	Data based			✓												✓
	Selection strategy			✓				✓		✓	✓		✓		✓	✓
	Swarm and evolutionary algorithms															✓
Applications	Attacks detection															✓
	IoT system security															✓
	Cyber attacks															✓

Table 1 (continued)

Topics covered		Canzanese et al. (2015)	Ampah et al. (2011)	Chandala et al. (2009)	Callado et al. (2009)	Sperotto et al. (2010)	Zhang et al. (2009)	Sabahi and Movaghar (2008)	Xin et al. (2018)	Buczak and Guven (2016)	Bhuyan et al. (2014)	Ahmed et al. (2016)	Nguyen and Armitage (2008)	Hamed et al. (2018)	Nisioti et al. (2018)	Our survey
Performance measures	Classification based										✓				✓	✓
	Clustering based										✓					✓
	Swarm and evolutionary algorithms															✓

- Stating the importance of feature engineering, the paper discussed feature extraction and feature selection using conventional techniques such as filter, wrapper, and embedded, and feature selection using SWEVO.
- The paper presents different IDS datasets used in the previous studies and discusses recent datasets in detail which has not been explored lately.
- The paper discusses the importance of performance measures for comparing techniques statistically based on its discriminatory power. The list of performance measures that should be considered for empirically evaluating the performance of the classifier.
- Different applications of IDS presented for network and data security.
- Finally, potential challenges and research gaps are discussed and probable solutions to mitigate the research gaps are presented with the aim to address the issue of detecting novel and variety of attacks to meet the goal of network security.

The roadmap of the paper is shown in Fig. 1 and is described as follows: Sect. 2 describes general IDS taxonomy with a brief introduction to IDS, classification of IDS, response mechanism of IDS, architecture,and decision module of IDS. Section 3 discusses the various feature engineering techniques implemented to improve the performance IDS. Section 4 presents the different methods of ML, DL, and SWEVO for the IDS evaluation. Section 5 gives a brief overview of the datasets used for IDS and discussion on the CIC-IDS 2017 dataset. The performance metrics to be considered for evaluating IDS are presented in Sect. 6. Section 7 discusses the applications of IDS; Sect. 8 marks down potential challenges and future research directions. We have concluded our paper with future research scope in the field of IDS in Sect. 9.

1.2 Research methodology and search strategy

The goal of our survey is to perform comprehensive analysis and understanding of IDS in context to techniques implemented, feature engineering, performance measures, applications, challenges, and future research direction. The motive of our research study is to present a fundamental platform to the researchers in the field of IDS. With the advent of increase in usage of networking devices, it is challenging to handle a large amount of data generated from the devices. Furthermore, increase and evolution in data also results in increased attack possibilities and vulnerabilities in data/system resources. Therefore, it is crucial to study and explore role of IDS in securing data/system resources. Therefore, survey performed in our paper mainly prioritize to address research questions formulated in Table 2. Moreover, research questions are articulated to explore the role of IDS built using various techniques along with feature engineering, performance measures, datasets, and applications domains. Table 2 lists out various research questions pertaining to the study conducted.

1.2.1 Search strategy for article inclusion/exclusion

For our survey, we have selected research articles by performing manual as well as automatic search for assorting relevant articles that can suffice our research goals. In automatic selection process relevant articles from various electronic databases namely, IEEE, Springer, Science Direct, Wiley, and various national and international conference

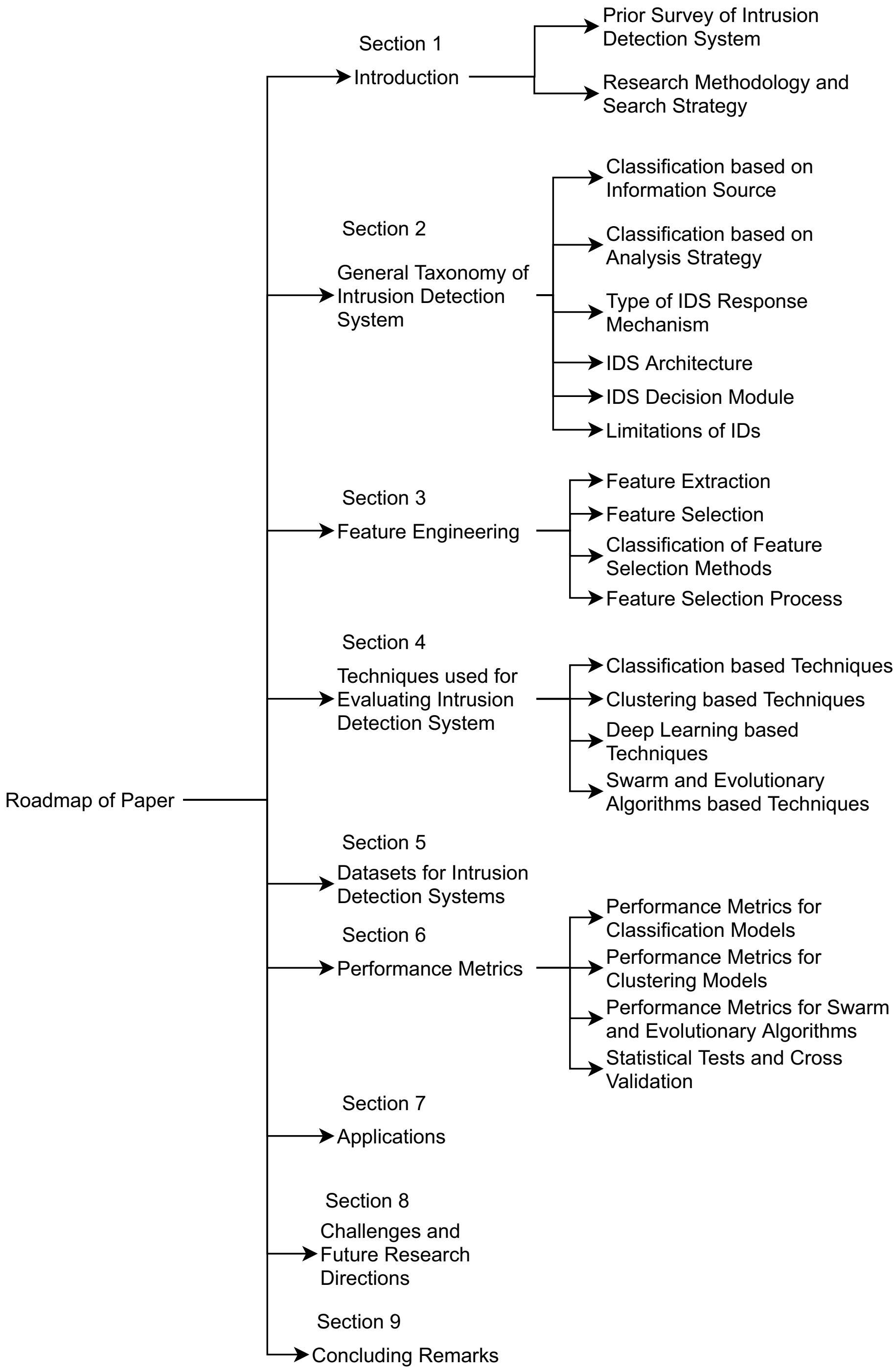
Roadmap of Paper
Section 1
Introduction
Prior Survey of Intrusion Detection System
Research Methodology and Search Strategy
Section 2
General Taxonomy of Intrusion Detection System
Classification based on Information Source
Classification based on Analysis Strategy
Type of IDS Response Mechanism
IDS Architecture
IDS Decision Module
Limitations of IDs
Section 3
Feature Engineering
Feature Extraction
Feature Selection
Classification of Feature Selection Methods
Feature Selection Process
Section 4
Techniques used for Evaluating Intrusion Detection System
Classification based Techniques
Clustering based Techniques
Deep Learning based Techniques
Swarm and Evolutionary Algorithms based Techniques
Section 5
Datasets for Intrusion Detection Systems
Section 6
Performance Metrics
Performance Metrics for Classification Models
Performance Metrics for Clustering Models
Performance Metrics for Swarm and Evolutionary Algorithms
Statistical Tests and Cross Validation
Section 7
Applications
Section 8
Challenges and Future Research Directions
Section 9
Concluding Remarks

Fig. 1 Roadmap of paper

Table 2 Research question, description, and section

Sr no.	Research question	Description	Section
1	What are the techniques used in existing research work in the field of IDS?	The question aims at analyzing the existing work performed in the field of IDS	Section 1.1 highlights the prior survey performed in the field of IDS. Moreover, Sect. 4 discusses various research work with classification, clustering, deep learning, and swarm and evolutionary algorithms
2	What is feature engineering and categories of feature engineering applied for IDS?	The question describes the feature engineering process and types of feature engineering techniques applied for IDS	Section 3 discusses various categories of feature engineering, classification of feature selection techniques, and feature selection process that might be used for selecting features for intrusion detection and classification
3	How IDS developed using various techniques are evaluated?	The question aims at exploring various datasets used for evaluating performance of IDS	Section 5 Datasets for intrusion detection systems focuses to explore various datasets developed for intrusion detection and classification
4	What are the various performance measures used for representing the efficiency of IDS model?	The question focuses on investigating various performance measures that can be used for demonstrating the performance of developed IDS model	Section 6 discusses various performance metrics for classification, clustering, and swarm and evolutionary based IDS models. Moreover, statistical tests and cross validation is also discussed
5	What are the different application areas where IDS can be used?	The question focuses on investigating various application areas wherein IDS can be used	Section 7 discusses various application areas where IDS can be used
6	What are the research gaps based on the literature review?	The question highlights various research challenges that needs to be addressed for designing efficient IDS	Section 8 Challenges and future research directions focuses to highlight various research challenges and future reseach directions in the field of IDS

proceedings and journals have been considered. The basis for considering different electronic databases was to have inclusion of comparative and systematic research work performed in the field of intrusion detection and classification. The research articles that have been considered for our survey are from the time frame of 2008 to 2020.

In manual selection process, we employed various keywords and variables for electing the articles for our study and analysis. The variables and keywords were based on following criteria and are listed in Table 3.

- The variables and keywords were in context with the defined research questions.
- The research articles were chosen by inspecting the search with alternative words or synonyms.
- Keywords and variables used for the search process were extracted from books and research articles related to IDS.
- Articles were also searched and linked using boolean operators namely "or" and "and" in the search string.

The search queries were formulated using the keywords/variables derived from various research articles and relevant books related to IDS. The keywords/variables used for forming search queries are "Intrusion Detection System", "Machine Learning", "Deep Learning", "Performance Measures", "Applications of IDS", "IDS datasets", to name a few. Thus, considering various keywords/variables search queries can be formulated as follows.

- ("Intrusion Detection System" OR "IDS") AND "Machine Learning".
- ("Intrusion Detection System" OR "IDS") AND "Deep Learning".
- ("Intrusion Detection System" OR "IDS") AND "Swarm and Evolutionary Algorithms".
- ("Intrusion Detection System" OR "IDS") AND "Datasets".
- ("Applicability of Intrusion Detection System" OR "Applicability of IDS" OR "Applications of IDS".
- "Performance Metrics" AND "Machine Learning".
- "Performance Metrics" AND "Evolutionary Algorithms"

Table 3 Types of variables/keywords considered for article selection for review

Variable/keyword	Explanation
Year	Year Duration (2008–2020)
Type of Study	Survey for IDS, Novel technique for IDS, Comparative Analysis of techniques implemented for IDS
Applicability of Study	Techniques implemented for which type of IDS and how the problem has been addressed
Technique	Type of technique implemented such as ML, DL, or SWEVO
Type of Classifier	Specific name of the technique implemented such as Decision Tree
Feature Engineering	Techniques implemented along with feature selection or feature extraction
Dataset	Techniques implemented with specific datasets designed for IDS such as NSL-KDD

- (“Intrusion Detection System” OR “IDS”) AND “Recent Trends”.

The process of selecting research articles from such extensive databases is a difficult task and therefore, relevant research articles were chosen based on certain inclusion and exclusion criteria. The inclusion criteria for research articles are summarized as follows.

- Research articles published in the time frame of 2008–2020 in the field of intrusion detection and classification were considered.
- Research articles published in the time frame of 2008–2020 and have applied ML, DL, or SWEVO techniques for intrusion detection and classification were selected.
- Research articles that have applied any feature selection technique along with ML, DL, or SWEVO technique were selected.
- Peer reviewed research articles from various databases namely, IEEE, Science Direct, Springer, Wiley, and various national and international conference proceedings of these databases were considered.
- Research articles discussing various intrusion detection datasets were considered.
- Research articles based on applicability and application of IDS were selected.
- Articles complying to defined research questions were selected.

IDS is a broad research area with variety of techniques that have been used for building an efficient IDS. Therefore, to justify our study with reasonable number of research articles, we have considered following exclusion criteria.

- Non-peer reviewed and editorial articles were not considered for our review.
- Articles based on IDS for specific technologies such as cloud computing, wireless sensor network, to name a few were excluded. This is because datasets used for these technologies were often simulated dataset and had different characteristic features.
- To ensure the uniqueness in the research work performed in the field of IDS, articles with similarity of already included articles were not considered.

1.2.2 Current trend of research in intrusion detection system

In recent years, ML and DL techniques are being applied widely for detecting attacks and vulnerabilities. However, there have been efforts to design and implement ensemble and hybrid techniques in order to achieve better results. Considering the current trend in research, we have included research work wherein.

- Feature engineering in terms of feature selection and feature extraction have been applied along with ML, DL, or SWEVO techniques.
- Recent and frequently used datasets have been used.
- Hybridized or ensemble architecture has been designed for intrusion detection and classification.

Thus, considering the current research trend as well as inclusion and exclusion criteria we have presented a survey on IDS with 170 research articles that constitutes the general taxonomy of IDS, techniques for I

DS, and intrusion detection datasets. Figure 2 shows number of articles included in last 5 years for our study based on inclusion and exclusion criteria.

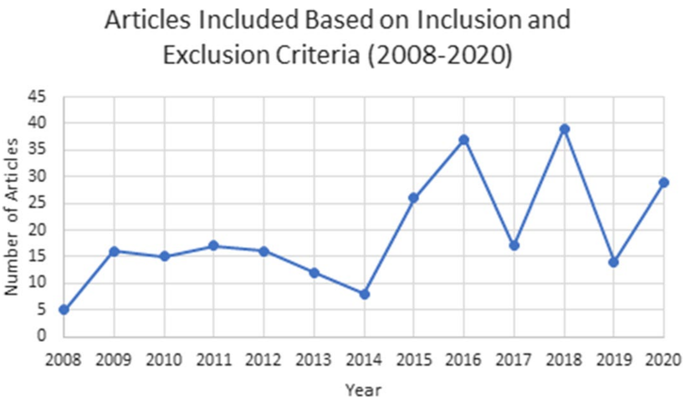

Fig. 2 Number of articles included based on inclusion and exclusion criteria (2008–2020)

2 General taxonomy of intrusion detection system

An anomaly can be stated as "an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism" (Gupta et al. 2016a). It is the action performed in order to trade-off security of the network and systems as contemplated by the computer security model which is confidentiality, integrity, and authentication (Vidal et al. 2020). This is achieved by breaching the security mechanism, gaining unauthorized access, and performing attacks within or outside the network. IDS is the system that provides security towards such thwarts by monitoring data coming from varied sources. The common types of attacks detected by the IDS (Daş et al. 2015), their description along with examples are listed in Table 4.

The basic functionalities of IDS are as follows (Vasilomanolakis et al. 2015): (i) keeping track of network activities, (ii) analysing the gathered data, (iii) checking the system configurations for exploiting vulnerabilities if exists, (iv) identifying patterns or signatures of attacks, (v) storing the recognized pattern or signature in the database and (vi) generating alert if any pattern or signature matches. On the basis of the functionalities provided by the IDS, the components of IDS are shown in Fig. 3. By monitoring the network, information is collected from the network packets. The attackers perform network attacks by injecting malicious code or analyzing the network packets for gaining information. Attacks can happen either on the server that handles all the network transactions or the system host which is actually performing the network activities. Actions can also be performed to exploit vulnerabilities present in the system. In fact, techniques such as MLDL leads to have smarter IDS to detect network threats.

Components of IDS are as follows.

(i) Monitoring Network: A network needs to be monitored to gather necessary packets containing network-related information. A network packet is a combination of packet header and packet payload. Both, header and payload, can be useful to extract the necessary information to perform an attack. Even the network flow is analyzed to find patterns of data to be exploited to execute an attack. Hence, datasets built for intrusion detection have packet level and flow level features to classify attacks.

(ii) Data Collection: It refers to gathering the details about the target system on which the attack is to be conducted. This can be achieved by performing queries using

Table 4 Types of attacks (Daş et al. 2015)

Attacks	Examples	Description
Routing Attacks	Black Hole Attack, Grey Hole Attack, Worm Hole Attack, Sybil Attack, Man in the Middle, Byzantine Attack	These are the network layer attacks which spoof, alter or replay the routing information
Sniffer Attacks	Phishing, Spam Detection, Online Scam, Illegal transaction, Account Hijack, Defacement	These types of attacks are performed by stealing or intercepting the network traffic data using network sniffing tools
Passive Attacks	Eavesdropping, Traffic Analysis	These are the attacks performed on the cryptographic system in which the attacker does not communicate with the systems involved but attempts to break the system by observing the data flowing between them
Insider Attacks	User to Root, Flooding Attacks, Port Scanning	These are the malicious attacks performed on the system network where the attacker is someone who belongs to the system and is assigned the authorization to access the network. He may also have information regarding the architecture of the system
Malicious Attacks	Botnet, Malware	It is defined as the way to misuse the system by injecting malicious software such as computer viruses or by performing social engineering to take advantages of the user's system
DoS/DDoS	Buffer overflow, Ping of Death, ICMP, Smurf, UDP Flood, SYN Flood	These are the attacks which make the network busy by flooding the network with packets as a result preventing the legitimate users to access the network resources
Cyber Attacks	Cyberbullying, Cyber Stalking	These are the targetted attacks performed on a specific organization and individuals to gain access to confidential, intellectual, institutional assets in order to damage or have financial gain
Vulnerabilities	Misconfiguration, System Exploits	These are the loopholes in the code or system design that creates a possibility for the system to be compromised. These are the flaws that generate possible attack points, through which an attacker can execute their code or ingress a system's memory
Others	Password Attacks, Brute force, Dictionary Attacks	These attacks are performed to steal the passwords or other intellectual and confidential information of the users

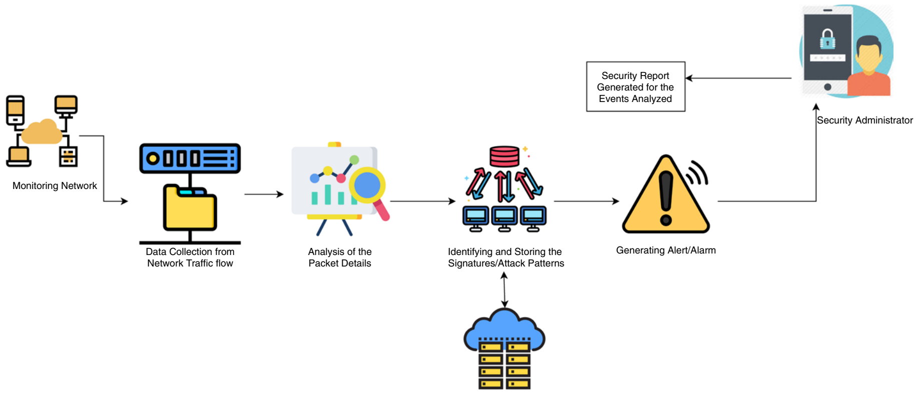
Monitoring Network
Data Collection from Network Traffic flow
Analysis of the Packet Details
Identifying and Storing the Signatures/Attack Patterns
Generating Alert/Alarm
Security Administrator
Security Report Generated for the Events Analyzed

Fig. 3 Components of IDS

network command or tools. For instance, packet-level details can be obtained by sniffing the packets flowing through the network using "Wireshark" or obtaining server and host-related details such as domain name using network commands like "nslookup" (Mandal and Jadhav 2016).

(iii) Analysis of Packet Details: This can be referred as scanning the network packet for stealing confidential information. For instance, an R2L (Remote to Local) attack can be performed by compromising the system and gaining unauthorized access to the system. Some of the attacks which can be carried out for gaining access are sniffing the packet and stealing the credentials or injecting the malware such as a trojan horse to gain remote access of the system. More often, these types of vulnerabilities can be exploited only if the target system has few open ports.

(iv) Identifying and Storing the Signature/Attack Patterns: The next step after analysis of packet details is to identify the attack patterns of already known attacks and novel attacks or signature of some known exploits which can be used to launch insider attacks. These signatures and patterns are stored in the database for the future reference; and hence, the security administrator can easily report intrusive behavior, if found anomalous.

(v) Generating Alert: After recognizing the attack pattern, an alert/alarm is generated and reported to the security administrator. Alert is triggered based on the matching of the signature/pattern.

The classification of IDS is presented in Fig. 4. It is classified based on the information source used by the IDS for analysis of the network or analysis strategy adopted by the IDS for classifying the intrusions. Moreover, the type of the response given by the IDS, the architecture adopted, and the type of the decision delivered by the IDS based on the architecture can also be considered as the attributes for classification.

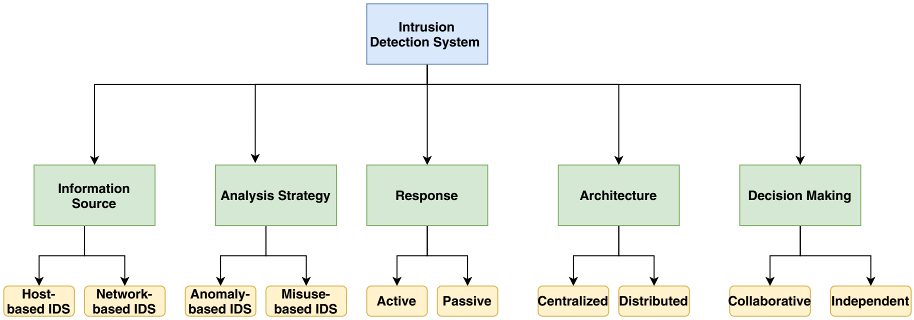

Fig. 4 Classification of IDS

2.1 Classification based on information source

The data collected for analysis of the intrusion can be gathered from varied sources, and hence, depending on information sources, IDS are classified into host-based IDS and network-based IDS (Vidal et al. 2020; Gupta et al. 2016a).

2.1.1 Host-based IDS

Here, an IDS is installed on the local system or host. The audit trails of the configured host are examined to gather information regarding the status of the system's behavior, signatures of any malicious activities and can take any preventive measures to protect the local system. The audit trails can be located from different sources such as system logs, application logs, and host monitoring. These logs can be gathered from operating system's network entity logs such as Unix, NT/2000/XP (Peng et al. 2016), security mechanisms such as firewall, network devices such as router and web server, and networking protocol such as FTP. The malicious activities can be recorded such as tampering the file data, segmentation fault error, system software crash, unauthorized access to the system, or rigorous use of system resources (Deshpande et al. 2018).

2.1.2 Network-based IDS

Here, IDS considers the entire network environment for monitoring activities in and out of the network. All the packets present inside and outside the network environment are examined. The network traffic data considered for examination increases the possibility of tracking potential loopholes that may compromise the network. Here, the network traffic being monitored is very massive and large. Therefore, network sensors can be deployed to tackle such a huge amount of data which may result in better efficiency and effectiveness of IDS. Network-based IDS analyses and inspects the audit trails of multiple hosts available in the network. In a network, there can be multiple events that can lead to intrusions and hence, each network event needs to be meticulously examined for intrusion detection (Vidal et al. 2020; Gupta et al. 2016a).

2.2 Classification based on analysis strategy

The detection of malicious activities and intrusive behavior of the system can be carried out by the analysis strategy adopted by the IDS based on the infrastructure of the system. Based on the analysis strategy, IDS are classified into anomaly-based IDS and misuse-based IDS (Vidal et al. 2020; Gupta et al. 2016a).

2.2.1 Anomaly-based IDS

In anomaly-based intrusion detection, significant patterns are examined for reflecting any deviations from the normal patterns. Network patterns can be analyzed statically and dynamically. If the state of the system does not change for a prolonged time then it is considered to be static. The network patterns can be analyzed using the software and hardware portion of the system. The configuration of the hardware portions of any system remains static, and therefore, it diverts the task of analysis towards the software portion. The main task of the system relies on the stagnant part of network data i.e. the code. For instance, in operating systems, data never changes from critical software to bootstrap. Static anomaly detection focuses on maintaining the integrity of the system. If an error has occurred or the part of the system has been tampered by an intruder then a static portion of system deviates from the previous state (Ranshous et al. 2015).

In dynamic anomaly detection, audit trails and monitored network traffic are taken into consideration. Audit trails in a system's operating system capture event system logs in a sequential manner (Ranshous et al. 2015). In the case of a distributed environment, partial sequencing of system log events is sufficient for detection. On the other hand, scenarios such as time interval of usage of the particular resource is considered. For such cases, normal consumption of resource is distinguished from abnormal consumptions by defining thresholds. Here, the detection of anomaly is achieved by tracking and monitoring the behavior of computer users. An alert is generated if the pattern of data or behavior deviates from the actual network traffic patterns.

The major advantage of using anomaly-based IDS is, zero-day attacks can be easily identified by analyzing patterns as slight variation from the normal traffic pattern is considered anamolous (Agrawal and Agrawal 2015). Moreover, it does not depend on the target operating environments. The drawback of such type of IDS is that it might generate a huge number of false positives. It is not necessary that every abnormal pattern in the network traffic is anomalous, the security expert might ignore some of these false positives which may lead to ignoring the real anomalous activities. While building the profile and constructing the training phase, there is a high chance that some user actions might be skipped if the network is not monitored properly. The log containing all the patterns of normal profile needs to be updated to reduce the false alarm rates (Agrawal and Agrawal 2015).

2.2.2 Misuse-based IDS

Misuse-based IDS is also referred as signature-based IDS. Here, an IDS is constructed based on system vulnerabilities and attack signatures which are already known. It deals with recognizing intruders who are trying to hamper the system by exploiting the vulnerabilities. For maintaining the security of the system, all the loopholes should be eliminated. Intrusion detection is a series of steps that result in an alert generation to take a preventive measure for any anomalous activities. The misuse-based methods differ in terms of how they differentiate or shape the behavior of any intrusion activity (Vidal et al. 2020; Gupta et al. 2016a). Ideally, misuse-based detection system use rules to explain the events which best describes the unusual actions inside the system. Many rules can be formulated and combined for estimating different intrusion scenarios. Misuse-based IDS looks for events that matched the rules. The events can be used for later investigation by audit records and can be monitored by examining the system calls.

The challenging task for misuse-based IDS is to keep the database containing attack signatures updated. Misuse-based IDS is not good with recognizing novel attacks as it fails to establish a correlation with the already available attack signature and new attacks. The maintenance of misuse-based IDS is a time-consuming process which involves continuous patching as well as analysis of vulnerabilities and exploits (Agrawal and Agrawal 2015). Acquiring the knowledge of any attack in an operating environment depends on the operating system version, platform, and applications. The detection of insider attacks is even more difficult. For instance, misuse of legitimate user privileges cannot be tracked or sensed by the system as malicious activity.

In regard to correctly identifying intrusion, user applications and network environment play an integral role. There are performance measures which help in concluding that which techniques should be implemented for detecting intrusion. These performance measures are derived from the ability to correctly predict an intrusion. The performance measures of predicting intrusion are classified as:

- Intrusion, Non-Malicious: This can be defined as the activity which is malicious but the system fails to detect the presence of intrusions. This can also be referred as False Negative (FN).
- Non-Intrusion, Malicious: This can be defined as the activity regarded as malicious even though it does not contain any intrusions. This can also be referred as False Positive (FP).
- Non-Intrusion, Non-Malicious: This can be defined as the activity identified as a non-malicious and non-intrusion. This can also be referred as True Negative (TN).
- Intrusion, Malicious: This can be defined as the activity which is intrusive and is correctly identified as malicious. This can be referred as True Positive (TP).

2.3 Type of IDS response mechanism

Response mechanism is the way an IDS responds when an intrusion has occurred; it can be an active or a passive response (Anwar et al. 2017). Active IDS response mechanism can be stated as the system built to block the intrusions or attacks instantly at the time they are detected without even concerning the security expert (Inayat et al. 2016). It has an advantage of detecting and handling attacks that occur in real-time. Some of the responses recorded by the active response mechanisms are:

- To generate an intrusion detection report
- To trigger alert/alarm
- To have an extra logging facility for the events occurring in the network
- To have a remote logging facility for the events occurring remotely
- To build an Intrusion Prevention System for preventing the suspected attacks instantly
- To have a backup of the activities logged

Passive IDS response mechanism can be stated as the system built to monitor the network traffic by tyrannizing the network operations having any unusual pattern or network activity. It cannot proactively handle the intrusions which have taken place in the network. Some of the responses recorded as passive response mechanism are:

- To lock the user accounts abruptly
- To suspend the running processes on the system
- To terminate the user login and shutdown the system
- To blocking the IP addresses of the users and dissolve the port services
- To create and employ temporary shadow files
- To enforce unauthorized access by remote login
- To intimidate the intruder

2.4 IDS architecture

Various infrastructure schemes have been suggested to meet the requirements of having an effective combination of resources and data for IDS. These infrastructures can be divided as centralized and distributed (Snapp et al. 2017). In centralized IDS, a central node analyzes the network traffic and triggers an alert if any unusual behavior is found. The information is collected from other network nodes, wherein each node monitors the network traffic and sends the information to the central node. Thereafter, the central node generates alerts on the basis of the information received from the corresponding nodes implanted in the network. These type of systems have shortcomings such as they have a single point of contact, so if the central node is compromised it may expose the entire system to be vulnerable. It leads to processing overhead because the amount of data and/or request handled by the central node is limited.

On the other hand in distributed IDS, each unit is capable of detecting and responding to the intrusion generated. A distributed IDS exhibits a tree-like structure. This is because the nodes used for the analysis of the network traffic are placed in a hierarchical manner wherein each unit communicate with each other from bottom to top. However, as the units are distributed they pose the challenge of fault tolerance, load balancing, and insider threat detection (Snapp et al. 2017).

2.5 IDS decision module

As discussed earlier, an IDS structure can be either centralized or distributed. On the basis of this, the decision-making scheme of an IDS can be grouped as collaborative or independent (Inayat et al. 2016). In a distributed IDS, multiple nodes are scattered in the network at different levels. Hence, the decision of analyzed activity is intrusive or not is decided in a collaborative manner. The decision is made using statistical techniques whereas, in centralized IDS, a single node independently derives the decision using the information assembled by the node.

Moreover, in distributed IDS, the units can either be distributed at different levels or the units are scattered at different places such as nodes in a cluster, but each node contributes collectively to different capabilities. While in the centralized IDS, the central node processes the data collected from the entire network. Table 5 summarizes the attributes of IDS along with its advantages and disadvantages.

Table 5 Summary of IDS attributes

Attributes of IDS	Categories	Advantages	Disadvantages
Information Source	Host-based IDS	• Provides system level protection • Works good with insider attacks	• Not good with zero day attacks • Dependant on system generated logs
	Network-based IDS	• Strong deterrence towards outside attacks. • Provides network level security	• Prone to false positives • Does not have any system related information
Analysis Strategy	Anomaly-based IDS	• Does not depend on the pre-defined signatures • Good with detection of unknown attacks	• Requires in-depth knowledge of the design • Not good with encryption
	Misuse-Based IDS	• Authentic and exhibits very low false alarm rate • Provides a very precise steps to mitigate the intrusion	• Cannot detect zero-days and variants of already known attacks • Needs regular update in signature database
Response	Active	• Real time detection • Proactive	• Lacks in depth analysis of packet • Based on logging activity
	Passive	• Good in analysing and monitoring packets • Indicates operator about the vulnerabilities	• Cannot do real time alert generation • Cannot take actions on its own
Architecture	Centralized	• Data is stored at a single point. • No computation overhead	• Single point of failure • Slow response time
	Distributed	• Peer-to-Peer communication • Data is handled in distributed manner	• Low detection accuracy • Fault tolerance • Complexity in load balancing
Decision Making	Collaborative	• Joint decision of all the units	• Computation overhead
	Independent	• Less computation overhead	• Decision is dependent on individual node

2.6 Limitations of IDS

Even though an IDS can be tuned to examine the contents of the network packets for inspecting the data to qualify and quantify the attacks, it still has some shortcomings as given below.

- IDS cannot prevent or block the attack just detected, by identifying the patterns or matching the signature of attack from the database. In order to prevent or block the intrusion, IDS must be integrated with other security mechanisms such as Intrusion Prevention Systems.
- An IDS performs a detailed analysis of the network and monitors the network activity, but it is not capable of executing necessary action at the time of detection of an attack. Therefore, it constantly needs a security officer or administrator to take actions against the identified threats in the network.
- An IDS is inefficient with the processing of encrypted network packets. It requires networking tools to examine encrypted network packets. This may leave the system resources in a vulnerable state until the intrusion is detected.
- The number of false positives generated by IDS is very high that affects the efficiency of the system.
- The attack signature database is required to be updated regularly to incorporate new attacks signatures.
- The IDS are susceptible to protocol-based attacks (Barbhuiya et al. 2013).

3 Feature engineering

A massive amount of data is generated in various domains such as social media, medical care, network security, and education. The ubiquitous nature of data results in a critical issue such as the curse of dimensionality, i.e, the problem of data sparsity when transformed into high dimensional space. Similarly, techniques that deal with datasets having a large number of features do not perform well as they incline to over-fit the unknown data. The large datasets require more memory and computational cost for analyzing the data (Carrasquilla 2010). In this regard, feature engineering turned out to be boon for handling high dimensional data. Feature engineering is a very prolific area of research in vivid fields of application like pattern recognition (Mitra et al. 2002), machine learning (Khan et al. 2018), and data mining (Talavera 2005), it has been used for applications such as text categorization (Nigam et al. 2000), image retrieval (Zhang et al. 2008), and intrusion detection (Wang 2010; OpenDNS 2016).

Feature engineering has turned out to be a potent and coherent strategy in handling low as well as high dimensional data for addressing classification problems. The significant empirical analysis of feature engineering has incorporated simpler and comprehensive models that have enhanced the performance of techniques in constructing more refined and comprehensive data. The current escalation of data pose considerable shortcomings in handling data and has increased the possibilities of using feature engineering for handling the data. In this section, we provide a substantial knowledge on feature engineering research inspired by various data-related problems such as redundant features and irrelevant features. We consider the feature engineering from data processing perspective and study

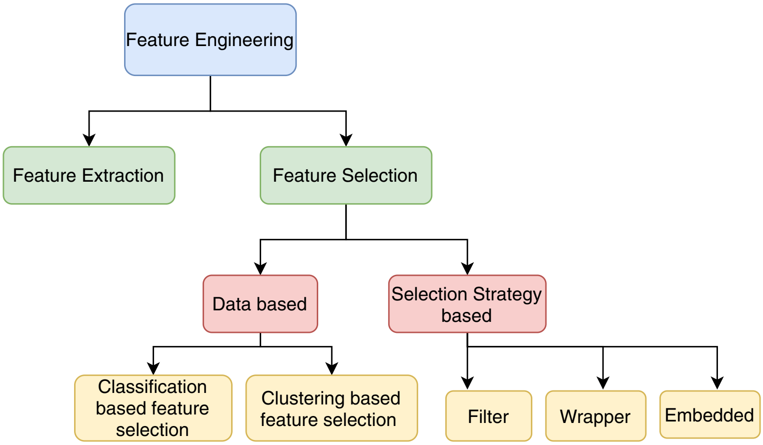

Fig. 5 Feature engineering

various aspects of feature engineering for transforming the data into a more refined form. To accentuate the need for feature engineering, it is divided into two categories namely, feature extraction and feature selection as shown in Fig. 5.

3.1 Feature extraction

Feature extraction deals with the reduction in the attributes of data. It projects the phenomenon of mapping high dimensional features to a feature space having a lower dimension. The projected feature space exhibits the properties of original features and can be demonstrated as a fusion of linear or non-linear features (Potluri et al. 2017). A feature selection method is described as the method of selecting relevant features from the underlying dataset. Both the above paradigms contribute to improve the performance of the learning model and increase its computational efficiency. Hence, both can be generalized as effective methods for feature engineering. Feature extraction can be useful for extracting features that can contribute to enhancing the performance of the learning algorithm. Feature extraction results in the creation of a new feature which changes the physical meaning of the features and as a result, it intricates any further analysis of these features (Carrasquilla 2010). Contradicting to feature extraction, feature selection sustains the physical meaning of the features by selecting a set of most relevant features from the original features (Mitra et al. 2002). This increases the efficiency and interpretability of the learning models. Thus, feature extraction and feature selection dominate feature engineering process by possibly improving the learning efficiency of the application model, decreasing computational cost, or avoiding over-fitting of data.

3.2 Feature selection

The feature selection method is described as the technique of obtaining the subset of features from the available features. The feature selection process can be illustrated by the framework proposed in Novaković (2016) which is based on selection criteria, evaluation criteria, and the techniques used for learning. The feasibility of the features obtained is evaluated based on evaluation criteria such as distance, information, dependence, and consistency (Liu and Motoda 2012; Jović et al. 2015; Ambusaidi et al. 2016). The dimension of the problem domain is directly proportional to the increase in the number of features and the problem of feature selection is believed to be NP-hard (Novaković 2016). A feature selection process can be as follows: generate an optimal subset of features, assess the generated subset of features, termination criterion, and validate the results obtained from the generated set of features (Liu and Motoda 2012).

3.3 Classification of feature selection methods

Feature selection algorithms can broadly be classified based on the data availability and selection strategy used for the feature selection.

3.3.1 Data based feature selection

According to the data availability, feature selection methods can be classified as classification and clustering-based methods on the grounds that data might have class labels for addressing different classification problems. For instance, the classification based feature selection method is designated for problems that exhibit the properties of selecting an optimal set of features that can easily classify the samples in different classes or estimate the targets for regression problem by establishing a correlation between different labels of classes or the attributes of regression targets. The selected feature plays a major role to train the classification and regression models. The feature selection process is independent of the learning algorithm. The learning algorithm embeds the selected features into the training model for assessing performance. These selected features can be used to classify new input into available classes using the underlying model. In general, classification based feature selection methods are majorly applicable to classification based problems (Liu and Motoda 2012).

In clustering-based feature selection methods, predicting the label of clustered data is a tedious task in terms of effort and time and hence, clustering-based feature selection methods gain attention for depicting the labels of data. To derive the importance of features, clustering-based feature selection methods define feature relevance. Unlike classification based methods, clustering-based feature selection techniques consider all the features that are present in the dataset. Even such feature selection methods do not depend on the learning algorithms. The clustering-based learning algorithm can be used to enhance the performance step-by-step or the features with most relevance can be embedded in the learning model. In the end, the feature selection produces the cluster organization of the dataset using the selected features. Generally, classification based feature selection methods perform better when sufficient labeled data is present, while clustering-based feature selection techniques do not need any form of labeled data (Liu and Motoda 2012). Therefore, it is advisable to have a feature selection method which can deal with both, labeled and unlabeled data and can select features considering correlation and relevance (Song et al. 2013).

3.3.2 Selection strategy based feature selection

On the basis of selection strategy, feature selection methods are classified into a filter-based selection approach, wrapper-based selection approach, and embedded feature selection approach (Saeys et al. 2007).

Wrapper-based feature selection methods depend on the predictive analysis of the learning technique employed to measure the qualitative characteristics of the features which are selected. For a particular learning model, a wrapper-based feature selection can be carried out in two major steps: i) identification of an optimal set of features from the given dataset, ii) evaluation of the selected features. This process repeated until some termination criteria is met. The subset of features is generated by the feature set search component and afterward, the machine learning technique is applied to evaluate the quality of the feature set selected based on the performance (Saeys et al. 2007). Thus, the entire process of feature selection and evaluation is carried out until the best learning performance is achieved. The wrapper-based method has a setback when it comes to search space; implementing a wrapper-based method for large search space becomes impractical (Dewa and Maglaras 2016). There are various methods which can be deployed to search features from the given search space to yield optimum learning performance such as sequential search, hill-climbing search, genetic algorithms, to name a few.

Filter-based feature selection methods do not depend on any learning methods. This method considers, attributes of data to measure the importance of the features. Filter-based methods exhibit better efficiency in terms of computational overhead than wrapper methods as they are independent of the classifier algorithm (Sánchez-Maroño et al. 2007). Filter-based feature selection can be carried out in two steps: i) based on some evaluation criteria each feature is ranked on the basis of its importance. ii) thereafter, features having low rank compared to others are eliminated. Feature ranking may be univariate or multivariate. In univariate analysis, each of the features is ranked independently regardless of any other feature and in multivariate analysis, multiple features are ranked in batch. Filter-based methods have the ability to select features based on the representative criteria such as feature correlation, mutual information, ability to preserve data, ability to reconstruct the original data.

The embedded method is the fusion of both wrapped-based and filter-based methods. As a result, this method comprises of merits of both the feature selection methods. Embedded methods communicate with the model and have the capability of processing features efficiently. The most commonly used methods are the regularization models that decrease the errors while fitting into the model (Duch et al. 2003). This method emphasizes that the coefficients of the learning model are very small.

3.4 Feature selection process

The feature selection process (Liu and Motoda 2012) is carried out in four steps as shown in Fig. 6 and it is as follows.

3.4.1 Subset generation

It is a procedure of optimally searching for instances in the search space to be evaluated. There are few methods such as forward search process that initializes an empty set of

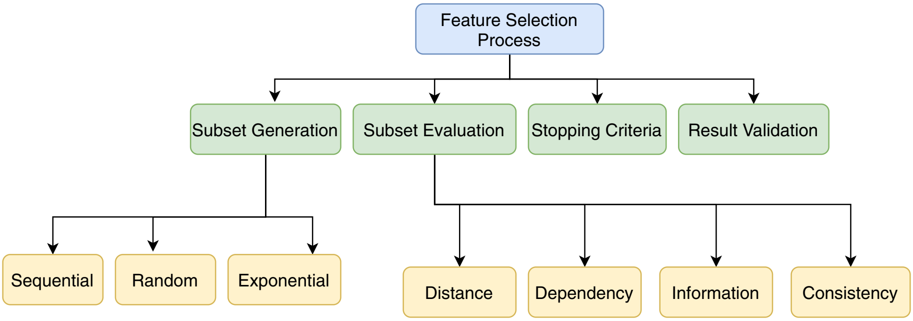

Fig. 6 Feature selection process

features in the beginning and iteratively adds features according to the search criteria defined or a backward search process that initializes with all the features available in the set and removes features iteratively according to the defined search criteria. Generating a subset can also be a random process. Selecting a subset of features randomly prevents the selected features to be confined to a local optimum (John et al. 1994).

For executing the search process, a search strategy must be adopted. For instance, for a dataset containing N features, there exist $2N$ possibilities to create a subset (Novaković 2016). To carry out an extensive search for the given dataset different search strategies can be explored such as exponential, sequential, or random. The exponential search is used for finding infinite or sorted lists. It performs the binary search in the search space defined with the complexity of $\mathcal{O}(\log n)$ (Liu and Motoda 2012). Optimal functions can be used to optimize the search with the high possibility of finding the results.

Another way of executing the search process is the sequential search which exhibits the properties of searching for subsets sequentially. Due to this, the sequential search process might ignore optimal subsets during the course of searching. Variants of the sequential search are greedy algorithm, sequential forward and backward elimination, and bi-directional search (Liu and Motoda 2012). The features in these methods are added and removed iteratively. These methods are simple to implement and search with the complexity of $\mathcal{O}(n^2)$ (Liu and Motoda 2012). The search process can also be carried out by randomly selecting the subset of features. The process of random search can be carried in two different ways: a sequential with randomness instilled can be carried out like in simulated annealing (Doak 1992), or a deterministic rule can be followed to generate a random subset from the given set like in LasVegas algorithm (Fausett et al. 1994). The feasibility of the selected features depends upon the availability of the resources and the randomness of the selection approaches helps in avoiding the local optima.

3.4.2 Subset evaluation

The next step in a feature selection process is to evaluate the selected features while generating the subset. Therefore, to check the optimality of the features, evaluation criteria such as distance, information, dependency, and consistency measure (Liu and Motoda 2012; Jović et al. 2015; Ambusaidi et al. 2016) are used.

Distance measure has discriminative properties. For instance, for a problem of two classes, consider two features P and Q. The feature P is chosen over feature Q if P exhibits higher distance between the conditional probabilities than Q as we aim to opt for the feature which can distinguish the two classes as far as possible. If the distance measure between the two features is zero then they cannot be differentiated.

Information measure refers to the significance of every feature by obtaining the gain ratio of each feature. Prior probability and posterior probability are calculated for every feature to obtain the gain ratio. For instance, for a feature P the gain ratio would be the difference between the prior probability and posterior probability of feature P. For the given features P and Q, if the information gain of P is greater than Q, then P is selected over Q.

Dependency measure refers to similarity or correlation between the features. It evaluates the dependency of each attribute from every attribute present in the dataset to predict the outcome. For instance, given a class C and features P and Q, an association between the class and features is formed, and if the association of P and C is greater than the association between Q and C then P is selected over Q.

Consistency measure uses the bias information and class information for selecting the set of features (Bennasar et al. 2015). The consistency of the features can be defined as the ability of the selected features to classify the given problem, as the entire set of features can classify the problem. For a given set of features, if the features have the same values and they classify in different classes then such features are said to be inconsistent.

Moreover, while using the wrapper-based feature selection method, there is a core dependency on the learning algorithm being implemented. The technique used for addressing the given problem can be improved by the process of feature selection as the selected features can be more suitable for giving out the best performance (Bennasar et al. 2015). For example, the detection rate of attacks can be considered as the dependent criteria for feature selection with network traffic data. The accuracy of each subset of features can be calculated and the best feature subset can be chosen that can computationally result in high accuracy (Bennasar et al. 2015). While dealing with clustering algorithms, the goodness factor of each feature subset is calculated based on the quality of clusters and the quality of the clusters can be defined by the cluster compactness, scatter separability, and maximum likelihood (Ni et al. 2016; Dash and Koot 2009; Alelyani et al. 2018).

3.4.3 Stopping criteria

This can be referred to as an indication to stop the selection process. Some majorly used stopping criteria are: when the search completes, when the defined criteria of iterations is exhausted, or when the desired subset is found.

3.4.4 Result validation

For the given dataset, the accuracy with the total features in the dataset and the selected features can be compared directly to validate the results like in the synthetic data traffic. If the information about the redundant and irrelevant features is available then this can also contribute to validate the result. With the real-world scenarios, it is quite difficult to get prior knowledge of the data and therefore, the algorithm to be implemented should learn the data for the evaluation. For example, simulations can be compared to check the false positive rate of the algorithm used with feature selection and with the whole feature set. The comparison can be performed with the full dataset and selected feature set (Bennasar et al. 2015). In many cases when dealing with real-world scenarios, no prior information regarding the application is known. Therefore, an alternative method should be implemented to measure the performance of the algorithm with feature selection. For instance, the error rate for classifying the data can be used as one of the performance metrics for a given subset of features of the model build. The results can be represented as a comparison showing the error rate of the model before and after feature selection (Liu and Motoda 2012).

4 Techniques used for evaluating intrusion detection systems

IDS has been a diversified field of research; methods from ML, DL, and SWEVO have been implemented to address the intrusion detection as shown in Fig. 7. In this section, we have discussed ML methods, Neural Networks (NN), and SWEVO which have been used

for evaluation of IDS models. These methods addressed the problem domain of IDS by classifying the problem based on the type of data explored. Apart from IDS, these techniques have been used for variety of application domains such as computer vision (Pareek and Thakkar 2021; Thakkar et al. 2013), recommender systems (Chaudhari and Thakkar 2019a; Patel et al. 2012b), stock market prediction (Thakkar and Chaudhari 2021, 2020a, b), handwriting recognition (Chaudhari and Thakkar 2019b), object detection (Patel et al. 2012a), sentiment analysis (Mungra et al. 2020), emotion recognition (Sharma et al. 2019) network security, to name a few. Data exploration is a way of encapsulating, conceptualizing, and analyzing the important characteristics of the data within the dataset. Exploring the data is an integral part of handling any classification problem. This is because it allows to get familiar with the future result, as well as interpreting them correctly. Such level of certainty can only be achieved by validating the raw data and fortifying the data collected without any fallacy. Data exploration also helps in refining the process of feature engineering that will be used in constructing efficient learning models.

There are multiple exploratory techniques to study dataset. Data exploration is majorly performed using the following methods.

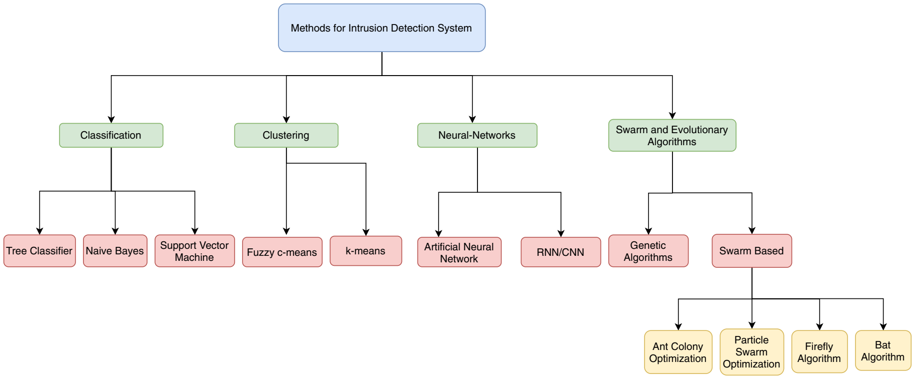

Fig. 7 ML, DL, and SWEVO techniques for IDS

- Uni-variate Analysis: It gives the statistical summary of each attribute associated with the raw dataset.
- Bi-variate Analysis: It derives the statistical relationship between predicted instances and the target instances of interest.
- Multivariate Analysis: It is performed to analyze the interactions between different attributes in the dataset.

The data exploration leads to generating accurate models and making efficient use of resources. The important facet of exploring the data is data profiling that deals with building dataset through descriptive statistics. The basic task for any classification problem is to analyze the dataset for better understanding. The analysis of data helps in visualizing the data in better way. For instance, formalizing IDS as ML-based systems, the raw data are fed during the training phase with labeled samples of network traffic or system calls that help the learning algorithm to know about the potential threat patterns. This can result in a very fast and efficient way to build a model which can predict threats. However, this can pose challenges in the form of a large number of false positives, zero-day attacks, and difficulty in compiling the initial data for training the system (Gharib et al. 2016).

False positives are the result of normal network traffic predicted as a threat. For instance, a user may enter an incorrect password for consecutive three times or use service which is a violation from the standard profile. The zero-day attack can be stated as the attacks whose signature or pattern is not known to the system. The latter drawback can be overcome by building a public dataset like DARPA (Brown et al. 2009) (McHugh 2000) or KDD CUP 99 (Tavallaee et al. 2009). Though these datasets have been considered for research, network attacks and traffic patterns have evolved over the course of time. Therefore, it is very tedious and computationally expensive to identify labels of network data fields that are normal or anomalous. Thus, exploring the data can help in overcoming the drawbacks of network dataset and help in building a statistical relationship between the attributes of data (Kabir et al. 2018). Further, the section discusses the various classification, clustering, neural network, and SWEVO algorithms implemented for evaluating IDS.

4.1 Classification based techniques

The classification based methods build the model by training the set of labels available in the dataset and classifies a data sample of the test set by using the classes identified during the training of the dataset. Classification based techniques work well with the labeled dataset and classify the data into known classes. For the tree classifiers, data is classified by constructing a tree with nodes and edges. The complete knowledge of dataset is not required for construction of the trees; also the high dimensional data can be easily handled irrespective of the type of data (Farnaaz and Jabbar 2016). The classification process becomes tedious if we have only numerical datasets.

The NB classifier is built on the prior probability and conditional probability of each attribute of the given class (Mukherjee and Sharma 2012). The classifier learns about the system using these probabilities and gives high accuracy. Therefore, if the prior knowledge is not correct, it might not perform better; it does not work well if the features are continuous. SVMs are insensitive to input data and its size. It has the capability of converting the non-separable problem into a separable problem if the dimension of the input space is sparse (Tsoumakas et al. 2010). It has the ability to deal with the outliers in the dataset. It is a binary classifier and its computation time is quite large. The neural network-based methods work significantly with high dimensional data, as more data is fed into the models they train better (Naseer et al. 2018).

The computational power of the algorithms increases exponentially and thus, they can perform better than machine learning algorithms (Javaid et al. 2016). Neural network methods such as ANNs have the ability to learn better from the initial inputs and derive relationships based on the input data (Thakkar et al. 2020). This characteristic can help to predict the unseen and unknown data accurately. The development process of neural networks for training is very long and complex. These techniques require large computational time and hence, are computationally very expensive. The performance of the classifiers can be improved by applying feature selection techniques along with the learning algorithm. A summary of classification-based methods for IDS derived from reviewed articles is presented in Table 6.

4.1.1 Tree classifiers

It is a popular classification technique for estimating the outcome by interacting with variables of the underlying dataset. Tree classifiers built with divide and conquer strategy which is the basis of a greedy algorithm. It iteratively constructs a tree-like structure with a root node, leaf nodes, and branches (Pandya and Pandya 2015). Every leaf node in the tree represents an outcome of the decision and the edges represent the decision rule applied on the node for splitting it into different attributes. For deciding the root node, Information Gain (IG) of each attribute is measured and the attribute with the highest gain value is selected as the root node (Sánchez-Maroño et al. 2007). Gradually, for the next split, the IG of the attributes is calculated and the attribute with the highest IG value is nominated as the next node. This process continues until no further attributes are left for splitting (Sánchez-Maroño et al. 2007). There are many variants of tree classifiers such as decision tree, C5.0, and C4.5 (Bujlow et al. 2012), a new version of C4.5, J48 (Sahu and Mehtre 2015), and random forest classifier (Farnaaz and Jabbar 2016).

For building an ideal model for classification, two key aspects are taken into consideration: choosing the dataset for evaluation and selecting a model for evaluating the dataset. Therefore, as the data might contain partially noisy data or irrelevant data, selecting appropriate attributes play a significant role. Therefore, in Sheen and Rajesh (2008) three different approaches for selecting features from the dataset namely, chi-square, IG, and Relief-F which are filter-based feature selection methods implemented on KDD CUP 99 (Tavallaee et al. 2009) dataset. 5000 records are randomly chosen from the dataset and classified into two classes namely attack or normal.

Feature selection algorithms are used with these records and the most significant and highest rank features are listed. The selected features fed to the decision tree classifier model and 10-fold cross-validation used to validate the results (Sheen and Rajesh 2008). The results compared with respect to the classification accuracy with most significant 5, 10, 15, and 20 features; the comparison depicts that IG and chi-square provide similar performance compared to Relief-F. There is a considerable increase in the classification accuracy when compared with classification accuracy obtained using all features.

Apart from these techniques, correlation-based feature selection technique and attribute ratio are also used to find an optimal set of features. For instance, in Chae et al. (2013) attribute ratio is used for feature selection which can be calculated by mean and frequency of the attributes in a given class, random mutation hill-climbing algorithm is used for

Table 6 Classification based methods for IDS

Ref	IDS	Dataset	Feature selection method	Number of features	Attacks detected	Detection method	Result analysis
Zhang et al. (2008)	Misuse, Anomaly, Hybrid	KDD CUP 99	IG	34	DoS, R2L, U2R, Probe	Random Forest, Outlier Detection	• DR: 94.7% • FAR: 2%
Sheen and Rajesh (2008)	Network IDS	KDD CUP 99	IG, Chi-Square, Relief-F	Top 5,Top 10,Top 15,Top 20	DoS, R2L, U2R, Probe	DT	• Accuracy with Chi-Square: 95.5% • Accuracy with IG: 95.5% • Accuracy with Relief-F: 93.4%
Hu et al. (2008)	Network IDS	KDD CUP 99	Weak & Strong Classifers	Not mentioned	Normal, Attack	Adaboost	• DR: 90.04% • FAR: 0.307
Li et al. (2009)	Network IDS	KDD CUP 99	Random Mutation Hill climibing	18	DoS, R2L, U2R, Probe	SVM	• ROC Analysis has been presented for all the five classes of the dataset
Ahmad and Alghamdi (2009)	Network IDS	KDD CUP 99	Basic and Traffic based features	23	Probe	MLP	• DR: 98% • FAR: 0.02
Li et al. (2010)	Anomaly	Stimulated Dataset	Packet Filtering	Not Mentioned	DDoS	ANN	• Classification Accuracy: 99.72%

Table 6 (continued)

Ref	IDS	Dataset	Feature selection method	Number of features	Attacks detected	Detection method	Result analysis
Farid et al. (2010)	Network-based and Host-based IDS	KDD CUP 99	IG	Top 19, Top 17, Top 12	DoS, R2L, U2R, Probe	NB, DT	• For 41 features DR:99.7% FAR:0.06 • For 19 features DR:99.8% FAR:0.05 • For 17 features DR:99.95% FAR:Not mentioned • For 12 features DR:99.98% FAR:Not mentioned
Panda et al. (2010)	Network-based	NSL-KDD	Parameter Learning	Not mentioned	Anomaly , Normal	Multinominal NB	• DR: 96.5% • FAR: 3%
Amiri et al. (2011)	Network-based	KDD CUP 99	Correlation Based	Variable for each attack class	DoS, R2L, U2R, Probe	SVM	• ROC Analysis has been presented for each of the classes of the dataset
Al-Janabi and Saeed (2011)	Anomaly	KDD CUP 99	Flitering	22	DoS, R2L, U2R, Probe	ANN	• ROC Analysis has been presented for each of the classes of the dataset
Norouzian and Merati (2011)	Anomaly	KDD CUP 99	Feature Vector	13	DoS, R2L, U2R, Probe	ANN	• Classification Accuracy: 96.78% • FAR: 2%

Table 6 (continued)

Ref	IDS	Dataset	Feature selection method	Number of features	Attacks detected	Detection method	Result analysis
Lin et al. (2012)	Misuse, Anomaly	KDD CUP 99	Decision Rules	23	DoS, R2L, U2R, Probe	SVM, DT, SA	• Accuracy with 41 features SVM: 99.03%, DT: 98.5% • Accuracy with 23 features SVM: 99.96% DT: 99.6%
Mukherjee and Sharma (2012)	Anomaly	NSL-KDD	CFS, IG, Gain Ratio, FVBRM	CFS:10, IG:14, Gain Ration:20, FVBRM:24	DoS, R2L, U2R, Probe	NB	• Accuracy and RMSE with 10 features: 97.55%, 0.088 • Accuracy and RMSE with 14 features: 95.30%, 0.11 • Accuracy and RMSE with 20 features: 95.21%, 0.12 • Accuracy and RMSE with 24 features: 97.78%, 0.083 • Accuracy and RMSE with 41 features: 95.11%, 0.12

Table 6 (continued)

Ref	IDS	Dataset	Feature selection method	Number of features	Attacks detected	Detection method	Result analysis
Chae et al. (2013)	Anomaly	NSL-KDD	CFS, IG, Gain Ratio, Attribute Ratio	Attribute Ration:22, CFS:25, IG:23, Gain Ratio:19	DoS, R2L, U2R, Probe	DT	• Accuracy with 41 features: 99.763% • Accuracy with 22 features: 99.794% • Accuracy with 25 features: 99.781% • Accuracy with 23 features: 99.781% • Accuracy with 19 features: 99.794%
Thaseen and Kumar (2014)	Anomaly	KDD CUP 99	PCA	10	DoS, Probe	SVM	• For DoS DR: 99.4%, FAR: 0.0015 • For Probe DR: 99.85%, FAR: 0.0030
(Thaseen and Kumar 2014)	Anomaly	NSL-KDD	PCA	10	DoS, R2L, U2R, Probe	SVM	• CC: 0.995 • DR: 99.4% • FAR: 0.0015

Table 6 (continued)

Ref	IDS	Dataset	Feature selection method	Number of features	Attacks detected	Detection method	Result analysis
Subba et al. (2016)	Anomaly	NSL-KDD	PCA	17	DoS, R2L, U2R, Probe	SVM, MLP, C4.5,NB	• Accuracy and DR for 41 features • SVM: 99.63% and 99.16% • MLP: 97.16% and 96.77% • C4.5: 97.35% and 97.98% • NB: 95.16% and 91.65% • Accuracy and DR for 17 features • SVM: 99.13% and 98.68% • MLP: 96.76% and 96.13% • C4.5: 96.85% and 97.23% • NB: 94.56% and 90.75%

Table 6 (continued)

Ref	IDS	Dataset	Feature selection method	Number of features	Attacks detected	Detection method	Result analysis
Ambusaidi et al. (2016)	Misuse, Anomaly	KDD CUP 99,NSL-KDD, Kyoto database	Correlation Based Methods (FMIFS, MIFS, FLCFS)	KDD CUP 99 FMIFS:19, MIFS:25, FLCFS:17 NSL-KDD FMIFS:18, MIFS:23, FLCFS:22 Kyoto FMIFS:4, MIFS:6, FLCFS:7	DoS, R2L, U2R, Probe	LS-SVM	Out of the three correlation based feature selection methods FMIFS outperformed and the results for the three datasets are as follows: • For KDD CUP 99 dataset • DR:99.46% FAR:0.13 Accuracy:99.79% • For NSL-KDD dataset • DR:98.76% FAR:0.28 Accuracy:99.91% • For Kyoto dataset • DR:99.64% FAR:0.13 Accuracy:99.77%
Osanaiye et al. (2016)	Misuse, Anomaly	NSL-KDD	Ensemble-multi Filter	IG:14,Gain Ratio:14, Chi-Square:14, ReliefF:14, EMFFS:13	DoS, R2L, U2R, Probe	J48	Out of all the feature selection methods EMFFS has outperformed and the results are as follows: • Accuracy: 99.67%, DR: 99.76%, FAR: 0.42

Table 6 (continued)

Ref	IDS	Dataset	Feature selection method	Number of features	Attacks detected	Detection method	Result analysis
Ikram and Cherukuri (2016)	Anomaly, Misuse	NSL-KDD, GureKDD	PCA	31	DoS, R2L, U2R, Probe	SVM	• For NSL-KDD, 41 Features • Precision: 0.9471 Recall: 0.9809 F-score: 0.9637 • For NSL-KDD, 31 Features • Precision: 0.9970 Recall: 0.9970 F-score: 0.9970 • For GureKDD, 41 Features • Precision: 0.833 Recall: 0.862 F-score: 0.847 • For GureKDD, 31 Features • Precision: 0.997 Recall: 0.999 F-score: 0.998

Table 6 (continued)

Ref	IDS	Dataset	Feature selection method	Number of features	Attacks detected	Detection method	Result analysis
Kumar and Batth (2016)	Network IDS	NSL-KDD	CFS, Gain ratio, IG	CFS:10, Gain Ratio:18, IG:20	DoS, R2L, U2R, Probe	NB	• CFS with 10 features: • Accuracy: 98.0% FAR: 0.101 Precision: 0.96 F-score: 0.97 • IG with 20 features: • Accuracy: 97.1% FAR: 0.023 Precision: 0.97 F-score: 0.99 • Gain ratio with 18 features: • Accuracy: 98.6% FAR: 0.002 Precision: 0.98 F-score: 0.99
Thaseen and Kumar (2017)	Anomaly	NSL-KDD	Chi-Square	31	DoS, R2L, U2R, Probe	SVM, RBF	Average result analysis for all classes of the dataset are as follows: • TPR:0.987 FPR:0.001 Precision:0.995 Recall:0.996 F-score:0.996
Wang et al. (2017)	Anomaly	NSL-KDD	Logarithm Marginal Density Ratios Transformation	Not mentioned	DoS, R2L, U2R, Probe	SVM	• Accuracy: 99.31%, DR: 99.20%, FAR:0.60

Table 6 (continued)

Ref	IDS	Dataset	Feature selection method	Number of features	Attacks detected	Detection method	Result analysis
Aljawarneh et al. (2018)	Anomaly	NSL-KDD	IG	8	DoS, R2L, U2R, Probe	J48, Meta Pagging, RandomTree, REPTree, AdaBoostM1, Decision Stump and NB	Classification Accuracy for all five classes: • Normal: 99.7, DoS: 99.9, R2L: 96.2, U2R: 99.1, and Probe: 97.9
Napiah et al. (2018)	Anomaly, Misuse	Simulated Dataset	Correlation Based	5	Hello Flood, Sink-Hole, WormHole	MLP, SVM, J48, NB, LR	• The dataset is tested with six ML techniques namely MLP, SVM, J48, Naïve Bayes, Logistic, and Random Forest using the 5 selected features to detect and classify the routing attacks. From all these techniques J48 algorithm has outperformed in detecting the routing attacks with 99% of TPR
Kabir et al. (2018)	Anomaly, Misuse	KDD CUP 99	Optimum Allocation	Not mentioned	Dos, R2L, U2R, Probe	LS-SVM	• Classification Accuracy: 99.78%
Ahmad et al. (2018)	Network-based	NSL-KDD	Not applied	Not applicable	DoS, Probe, R2L, U2R	SVM, RF, ELM	• Performance charts of accuracy, precision, and recall are presented

Table 6 (continued)

Ref	IDS	Dataset	Feature selection method	Number of features	Attacks detected	Detection method	Result analysis
Patgiri et al. (2018)	Network-based	NSL-KDD	RFE	13	DoS, Probe, R2L, U2R	SVM, RF	• Performance charts of accuracy, precision, and recall are presented
Ahmim et al. (2019)	Network-based	CIC-IDS-2017	Not applied	Not applicable	DoS, portscan, brute force, web attack, infiltration, bot	DT, Rule-based	• DR: 94.4% • Accuracy: 96.6% • FAR: 1.1%
Yihunie et al. (2019)	Anomaly-based	NSL-KDD	Not applied	Not applicable	DoS, Probe, R2L, U2R	SVM, RF, LR, Stocastic Gradient Decent, Seqential model	• ROC graphs are presented for performance evaluation
Meftah et al. (2019)	Network-based	UNSW-NB15	RFE	5	DoS, exploits, worms, backdoor, shellcode	SVM, stocastic gradient decent, LR	• Accuracy (DT): 85.4% • Accuracy (NB): 60.70% • Accuracy (SVM): 70.21%
Khraisat et al. (2020)	Hybrid IDS	NSL-KDD, AFDA	Not Applicable	All features	DoS, Probe, R2L, U2R	DT, one-class SVM	• Accuracy for NSL-KDD : 83.24% • Accuracy for AFDA: 97.04%
Rajagopal et al. (2020)	Network-based	UNSW-NB15, UGR-16	Not Applicable	All features	DoS, exploits, worms, backdoor, shellcode	Stacked Classifier	• Accuracy: 94% • Precision: 96% • Recall: 93% • *f*-score: 95%

Table 6 (continued)

Ref	IDS	Dataset	Feature selection method	Number of features	Attacks detected	Detection method	Result analysis
Ambikavathi et al. (2020)	Network-based	CIC-IDS-2017	RF	16	Bot, Brute force, DDoS, DoS, FTP, Infiltration, Portscan, SQL injection, SSH, XSS	RF	• Accuracy: 97.34%
Bhati and Rai (2020)	Network-based	NSL-KDD	Not Applicable	All features	DoS, Probe, R2L, U2R	SVM	• Accuracy for Linear SVM: 96.1% • Accuracy for Quadratic SVM: 98.6% • Accuracy for Fine Gaussian SVM: 98.7% • Accuracy for Medium Gaussian SVM: 98.5%
Krishnaveni et al. (2020)	Network-based	NSL-KDD	IG	10	DoS, Probe, R2L, U2R	SVM	• Accuracy : 96.34%
Thakkar and Lohiya (2021)	Network-based	NSL-KDD	Chi-Square, RFE, IG	Chi-Square: 10, RFE:13, IG:18	DoS, Probe, R2L, U2R	DT, RF, LR, *k*-NN, SVM, NB, ANN	• SVM with RFE outperformed other classifiers

feature selection in Li et al. (2009) and strong and weak classifiers are used for feature selection in Hu et al. (2008). In Chae et al. (2013), the experiments are performed on NSL-KDD dataset (Tavallaee et al. 2009) with 76121 total normal records and 49852 total attack records having 41 features. J48 decision tree classifier is used for performance evaluation. Attribute ratio ranks the features in the order of their frequency of occurring in a class; a total of 22 features out of 41 are selected. The results were formulated and compared with IG, Correlation-based feature selection, and gain ratio. The accuracy results compared with the dataset consisting of full features, and the features selected using attribute ratio. The later showed a significant increase in the accuracy.

The most common limitation of IDS is their inability to detect zero-day attacks or attacks whose signatures are not known. To handle and address novel attacks,

the advantages of misuse-based and anomaly-based IDS are combined in Zhang et al. (2008) and a hybrid IDS model is proposed as shown in Fig. 8 that is capable of detecting known attacks as well as novel attacks. The approach works in two phases, namely, offline and online phase. Signature patterns are constructed that are stored and detected by the misuse-based component whereas, unknown and uncertain actions are depicted by the anomaly-based detection component.

The experiments have been performed on KDD CUP 99 dataset (Brown et al. 2009; McHugh 2000) with five types of services listed by the dataset as FTP, HTTP, telnet, SMTP, and POP with 16919 instances in the training set and 49838 instances in the test set (Zhang et al. 2008). To enhance the performance of the hybrid IDS, random forest classifier was used with variable importance and outlier detection for feature selection that resulted in selecting 34 most significant features from the dataset. These features are then combined in different ways to build patterns for attacks. Two of the most important parameters of random forest classifier are *mtry* (samples of candidate variables at each split) and number of trees, that are optimized with different values to get error rate. The minimum error was recorded with a number of trees as 15 and *mtry* as 34, and overall accuracy of the hybrid approach was 94.7% with the false positive rate as 2.2% (Zhang et al. 2008).

With an increase in the network traffic, ensemble feature selection method is proposed as shown in Osanaiye et al. (2016) that caters the outcome of multiple filters based feature selection methods to achieve optimum selection. In the framework for ensemble-based multi-filter feature classification, four filter-based methods are combined to reduce the feature set of NSL-KDD dataset to 13 features from 41. The proposed framework is shown in Fig. 9 (Osanaiye et al. 2016). Here, four feature selection methods IG, gain ratio, Relief-F, and chi-square are combined to enhance the performance of the model and the decision tree classifier model is built for evaluation. The four feature selection methods as stated in Osanaiye et al. (2016) are used to rank the features present in the original dataset, and based on the rank, best 13 features are selected. A threshold value is defined to measure

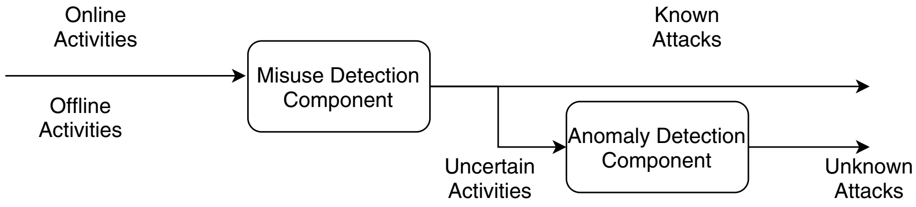

Fig. 8 Hybrid intrusion detection system (Zhang et al. 2008)

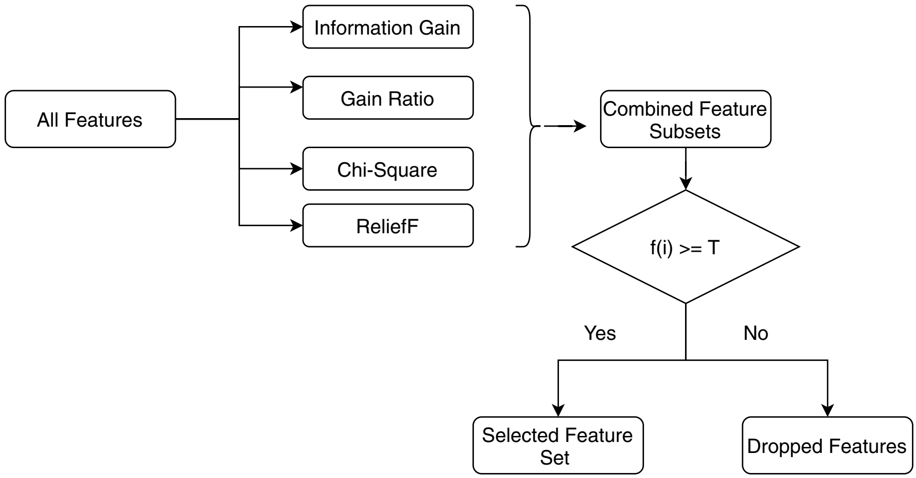

Fig. 9 Combined feature selection method (Osanaiye et al. 2016)

the frequency of the occurrence of each feature and threshold value is determined by using majority voting. While generating the combined feature subsets, a counter is used to determine feature with a value equal to a threshold. The proposed method achieved the accuracy of 99.67%.

An ensemble of DT classifier and rule based approaches are proposed in Ahmim et al. (2019). Three classifiers are used namely REPTree, JRip, and Forest algorithm for classifying the data as attack or normal. The experiments are performed using CICIDS2017. The classifiers use features of the data set to classify the data into normal traffic and specific attack category. There are total 14 attack categories in the dataset. The results are presented in terms of detection rate, accuracy, and false alarm rate. The proposed model achieved 94.4% detection rate, 96.9% accuracy, and 1.1% false alarm rate.

A network-based intrusion detection is performed using RF classifier in Ambikavathi et al. (2020). Here, feature selection is performed by measuring feature importance using RF. Further, RF classifier is applied on the reduced feature set and attack classification is performed. The experiments are performed using CIC-IDS-2017 dataset and RF classifier performance is presented in terms of accuracy, efficiency, and detection rate. The reduced feature set achieved the accuracy of 97.34% with RF classifier.

4.1.2 Naïve Bayes

Bayesian classifier based on the Bayes theorem (Chebrolu et al. 2005) and addresses the classification problems by addressing the prior and posterior probabilities of the instances of the dataset. For example, consider a sample vector Q with $q_1, q_2, \dots, q_n$ instances. For a given sample if the instances are classified into n classes, then for classifying a sample Q to a class C_i the conditional probability is given as $P(Q|C_i) * P(C_i) > P(Q|C_j) * P(C_j)$, where i and j are two different classes of the given sample space. The sample would be classified to the class which has the highest posterior probability. If the dataset contains categorical features then attribute ratio is considered for classifying the data with the frequency of its occurrence. For continuous variables, Gaussian distribution is considered for classification.

Bayes theorem states that the attributes of the dataset are independent of each other as it measures the probability of the predictor variable given a class or posterior probability of the class given the target variable. Naïve Bayes (NB) classifier has shown good progress in the field of email spam detection and text categorization. The drawback of this classification method is the knowledge required for prior probabilities. The prior knowledge required is dependent on the number of instances in the class, and attribute class cardinality relationship. With large datasets, the computational complexity increases.

IDS is a classification problem where NB is applied for classifying attacks by selecting significant features. In Mukherjee and Sharma (2012), four feature selection methods have been considered namely IG, gain ratio, correlation-based, and feature vitality based reduction method. The experiments were performed on NSL-KDD dataset and 24 features were selected using the method proposed in Mukherjee and Sharma (2012). Here, the feature vitality measured by implementing a sequential search to find significant features. Initially, all the features were taken and iteratively feature were removed using the "leave-one-out" strategy until no considerable improvement in the accuracy was observed. To measure the importance of each of the features, experiments were carried out 41 times. This method was based on the accuracy, true positive rate, and false positive rate of the system. The result comparison of the proposed method with other feature selection methods using NB classifier showed considerable improvement in the accuracy of the IDS model (Mukherjee and Sharma 2012).

In Meftah et al. (2019), RFE is used as feature selection techniques for ranking and extracting features based on their importance. The experiments are conducted on UNSW-NB15 dataset and top 5 features are selected for classification. DT, NB, and SVM are used for attack classification. The results showed that pre-processing data and applying feature selection method improve the performance of the classifiers.

Bayesian classifiers are also combined with statistical techniques for feature reduction. For instance, statistical filters like Principal Component Analysis (PCA), random projection, and nominal to binary are combined with the NB classifier for selecting features (Panda et al. 2010). Therefore, the proposed method referred as discriminative parameter learning as the attributes selection is performed by discriminatively measuring frequency of the attributes also called the frequency estimate. Here, two-class classification was performed using the NSL-KDD dataset with 25192 instances and 41 attributes. The results showed that the NB classifier with nominal to binary supervised filtering approach outperformed the other methods.

Hybridization of classifiers can also be used for improving the performance of classifiers. In Farid et al. (2010), NB classifier is combined with DT classifier. The features are ranked based on IG value, and thereafter highest rank features are used for attack classification. Here, five class classification performed using KDD CUP 99 dataset for classifying instances as Denial of Service (DoS), Probe, Remote to Local (R2L), User to Root (U2R), and normal traffic. NB can also be used to detect routing attacks by analyzing the compression header of the network packet. The greedy hill-climbing algorithm is used for selecting features, and classifiers such as NB and J48 implemented to detect a combination of routing attacks such as a sinkhole, wormhole, and syn flood. The results show that NB outperforms in comparison with the other classifiers (Napiah et al. 2018).

4.1.3 Support vector machine

Support Vector Machine (SVM) extensively used in comparison with other machine learning methods. It is a classification technique that segregates the data using hyperplane by maximizing the margin between the data points and hyperplane. The analysis of the underlying problem performed by SVM is optimal as the model is trained by evaluating a linearly constrained quadratic equation (Goeschel 2016). It is based on the statistical learning theory. SVM also works well with the non-linear data as it converts the non-linear data into linearly separable by mapping the data points to a high dimensional feature space using transformation functions (Thaseen and Kumar 2014). These functions are usually referred as kernel tricks.

A large number of patterns can also be learned by SVM using least square SVM (Lever et al. 2016). In this method, linear equation is evaluated during optimization stage to prevent convergence to local minima for the given SVM model. Generally, SVM can be considered as the binary classifier as the data classified contains only two values positive and negative. We can obtain multi-class SVM classifier by combining data into multiple binary problems (Xie et al. 2014).

The effectiveness and feasibility of detecting intrusions can be enhanced by using feature selection with SVM. In study (Thaseen and Kumar 2014; Subba et al. 2016), PCA is used for feature reduction and to find out an optimum feature set. In the pre-processing step, the categorical features mapped to numerical features by feature scaling. NSL-KDD dataset is used with 41 features and after feature selection 23 features are selected. The radial basis function used as the kernel function to handle the high dimensional features of the dataset. The accuracy of the model with and without applying feature selection technique is presented. The results show an increase in accuracy when feature selection is applied. In Thaseen and Kumar (2014), KDD CUP 99 dataset is used; the pre-processing step involves feature normalization and PCA is used with SVM for optimizing the kernel parameters and performing automatic parameter selection.

PCA also applied for feature reduction for NSL-KDD and GureKDD dataset in Ikram and Cherukuri (2016). It reduces the features to 31 out of 41 features and shows enhancement in the classification accuracy as well as low false alarm rate. In Patgiri et al. (2018), RFE is used as the feature selection technique with SVM and RF. The experiments are performed using NSL-KDD dataset and 13 out of 41 features are selected for attack classification. The results show that SVM outperforms RF for given attack categories.

The effectiveness and feasibility of detecting intrusions is also measured using varied sample size of the dataset. For instance, in Ahmad et al. (2018), varied samples size of NSL-KDD dataset is considered. The experiments are performed using SVM, RF, and Extreme Learning Machine (ELM) classifiers. The results indicate that ELM outperforms to other approaches. In Yihunie et al. (2019), binary classifiers such as SVM, stocastic gradient decent, sequential model, LR, and RF are applied on NSL-KDD dataset. The experiments are performed with and without feature encoders and results are recorded. The results showed that RF produced minimum number of false negatives and outperformed the other classifiers.

A linear correlation-based feature selection method cannot establish a relation between input variables and output variables whereas mutual information can build a statistical relationship between the attributes of the dataset (Amiri et al. 2011). This is depicted in Amiri et al. (2011) where two mutual information-based methods are implemented: conditional mutual information maximization (Fleuret 2004) and max-relevance min-redundancy

(Peng et al. 2005). For comparing the results of the two mutual information methods and correlation-based feature selection method, NSL-KDD dataset is discretized and the binary features are fed to the features selection method, which selects 15 best features from the dataset. The selected features are given input to the least square SVM model. The results are compared with all the three methods performing classification for five different classes (Normal, DoS, Probe, R2L, and U2R).

Similarly, the least square SVM model built with feature selection using mutual information; the model was tested on three datasets, namely KDD CUP 99, NSL-KDD, and Kyoto dataset (Ambusaidi et al. 2016). The results showed that the least square SVM model built with mutual information feature selection method is computationally efficient compared to other classification techniques used in Ambusaidi et al. 2016. The detection process of various types of attacks is complex and depends on the study of large datasets. These large datasets can be represented as subgroups using sampling which is also referred to as optimum allocation (Kabir et al. 2018). The framework developed in Kabir et al. (2018) uses optimum allocation along with least square SVM to extract and validate samples for detecting intrusions. The proposed framework implemented on KDD CUP 99 dataset. The results show that the proposed method works effectively for static as well as incremental datasets.

In Thaseen and Kumar (2017), multi-class SVM implemented with chi-square filter-based feature selection on NSL-KDD dataset with 33300 records. SVM is generally referred to as binary classifier but to address problems with multiple classes, a multi-class SVM model can be obtained by combining sets of binary classifiers. The proposed work uses a chi-square feature selection method to rank the features based on importance and frequency and low-rank features are removed from the dataset (Thaseen and Kumar 2017). Before performing the feature selection, the features are normalized using Z-score normalization to obtain the frequency of the attributes. The feature selection method helps to find the high priority features that best classify the data. In the second phase, the dataset is divided into three sets namely, training, test, and validation. The validation set is used to find the kernel parameter and over-fitting constant. These optimal parameters are given as input to the training model to predict the labels of the test data. The results are compared with other classification and clustering techniques that depict an improvement in detection rate and false alarm rate.

Apart from feature selection, feature augmentation is also used to provide high-quality data to the processing unit for training. This high-quality data proves to be an important factor in enhancing the performance of detection. An efficient IDS built using SVM with augmented features is presented in Wang et al. (2017), where feature augmentation is implemented using Logarithmic Marginal Density Ration Transformation (LMDRT) on NSL-KDD dataset. To be precise, the LMDRT method converts the original features with the aim of forming new and better quality features that can enhance the detection capability of the classifier.

The development of the hybrid approach significantly reduces the computation time and complexity associated with feature mapping for selecting important features from the dataset (Aljawarneh et al. 2018). A hybrid model with seven classifiers is presented in Aljawarneh et al. (2018), where the pre-processing is carried out using feature normalization to remove the unwanted noise from the data and the accuracy for each of the classifier is obtained. The classifier with the highest accuracy is chosen for attack classification and features are selected using IG. The model is built using the selected features and the best classifier. The experiments are performed on NSL-KDD dataset and proposed framework gave 99.81% accuracy for binary classification and 98.56% accuracy for multi-class classification. A hybrid approach is also proposed in Lin et al. (2012), wherein SVM is hybridized with DT and Simulated Annealing (SA) algorithm. In this hybrid approach, decision rules are derived from the decision trees that are used for feature selection. The experiments were performed on KDD CUP 99 dataset where 23 features were selected and the detection accuracy of 99.9% was achieved.

A hybrid IDS model is proposed in Khraisat et al. (2020), wherein DT and one-class SVM classifiers are integrated. Here, in the proposed model DT classifier is used for building a signature-based IDS, whereas one-class SVM is used for building anomaly-based IDS. The proposed model is designed to identify existing attacks as well as novel attacks. The experiments are performed using NSL-KDD and AFDA datasets and results are compared in terms of accuracy. The proposed model achieved accuracy of 83.24% for NSL-KDD dataset and 97.04% for AFDA dataset.

A stacked ensemble classification technique is proposed in Rajagopal et al. (2020) for network-based intrusion detection. Stacked classifier consists of RF, LR, k-NN, and SVM classifiers that are used for deriving optimal predictions based on the learning of the classifiers. The experiments are performed using the flow based datasets namely, UNSW-NB-15 and UGR-16. The stacked model for intrusion detection achieves accuracy of 94%. A binary classification technique is implemented using SVM to classify the network traffic as normal or anomalous in Krishnaveni et al. (2020). Here, feature selection is performed using information gain techniques and experiments are performed using NSL-KDD dataset. Ten features out of 41 are selected for training SVM classifier using the radial basis function. The proposed techniques achieves accuracy of 96.34%.

An analytical study based on SVM is performed for intrusion detection in Bhati and Rai (2020). Here, the study involves four basic steps namely, data collection, preprocessing, training and testing using SVM, and data prediction. The experiments are performed using NSL-KDD dataset and performance of variants of SVM namely, linear SVM, quadratic SVM, fine gaussian SVM, and medium gaussian SVM are recorded in terms of accuracy. Accuracy of 96.1%, 98.6%, 98.7% and 98.5% is achieved by linear SVM, quadratic SVM, fine gaussian SVM, and medium gaussian SVM, respectively. A comparative analysis of attack classification using feature selection techniques is performed using NSL-KDD dataset in Thakkar and Lohiya (2021a). Here, in the empirical study, seven ML classifier namely DT, RF, NB, *k*-NN, SVM, LR, and ANN are implemented for intrusion detection. Feature engineering is performed using chi-square, IG, RFE feature selection techniques. The experimental results showed that RFE with SVM outperforms other combinations of identified feature selection techniques and classifiers.

Research have been performed on the use of neural networks in intrusion detection. Neural network techniques like Artificial Neural Networks (ANNs) (Li et al. 2010) and Multi-Layer Perceptron (MLP) (Ahmad and Alghamdi 2009) used to address intrusion detection. These methods proposed for performing statistical analysis of IDS by identifying system users and significant variations from the usual behavior of the network environment. Neural network-based intrusion detection methods have layered designed built with neurons. The architecture consists of an input layer, an output layer, and hidden layers. Each neuron in the layers, acts as the feature vector. Combining these feature vectors attacks are detected for the underlying dataset (Norouzian and Merati 2011).

Multi-layer perceptron used for classifying attacks from the KDD CUP 99 dataset using the feature vectors as input in Norouzian and Merati (2011). The proposed model showed that the computational time of the training increases with a large number of features. The training process of a neural network is long, as it trains the feedforward network for different identified patterns, calculate and backpropagate the associated errors, and adjust the weights accordingly. In fact, the detection accuracy of the neural network framework also depends upon the number of layers considered for building the network (Al-Janabi and Saeed 2011).

4.2 Clustering based techniques

Clustering can be defined as the method of grouping the unlabelled dataset based on their similarity (Chitrakar and Huang 2012). The primary goal of these techniques is to combine the data samples based on their homogeneity. Clustering techniques are simple and computationally less expensive and used to cluster the data based on homogeneity. These methods are effective as they scale linearly and enhance the performance with multiple scans of the data. These methods are capable of handling the outliers present in large datasets.

Clustering algorithms are very sensitive and cannot correlate the knowledge that can interconnect or establish a relationship between the clusters and hence, it might fail if the clusters have complicated form or shape (Dy and Brodley 2000). Moreover, an IDS needs a technique that can adapt the real-time changes in the environment. For instance, techniques implemented for IDS should be capable of gathering new information regarding anomalies for detecting new intrusions. To meet this requirement, incremental learning techniques can be used that possess the ability to retrain the data. The promising aspects to improve the detection process of attacks can be listed as using ensemble-based classifiers, developing collaborative based IDS, and building a real-time IDS. A summary of clustering based methods for IDS derived from the reviewed articles is presented in Table 7.

The commonly used clustering techniques for IDS are fuzzy clustering, k-means, and *k*-NN. The main aim of the fuzzy clustering technique is to divide the group of instances into clusters having similar characteristics. These characteristics can be similarity of data points within the clusters or variance between the cluster data. A framework is proposed in Wang et al. (2010) to demonstrate the working of fuzzy clustering. The proposed framework divided into three modules. In the first module, fuzzy c-means segregate the dataset into several clusters. This helps in reducing the computational complexity of the dataset. In the second module, Artificial Neural Networks (ANNs) are used to identify similar patterns within the training set. This network consists of nodes connected through edges like that of a basic feedforward network. In Wang et al. (2010), backpropagation algorithm is implemented to detect anomalies. The proposed architecture consists of input, output, and multiple hidden layers.

Every node in the network has an input weight and to achieve the global minimum of these weights partial derivatives of these weights are computed and adjusted according to the learning rate using the gradient descent algorithm. In the third module, the results are combined for all the subset of ANNs to increase the detection accuracy by applying fuzzy aggregation. The experiments conducted with all the features of KDD CUP 99 dataset as a vector. These features divided into six subsets with the help of fuzzy clustering and on each subset, a three-layer neural network and fuzzy aggregation are implemented with the node structure as [41:18:5] and [5:13:5], respectively. The sigmoid transfer function used for the evaluation of weights of input and hidden nodes, and a linear transfer function used for the evaluation of the weights of output nodes. This module is effective in terms of using the integrated framework of fuzzy clustering and neural networks for obtaining the subset of features from the dataset.

Table 7 Clustering based methods for IDS

Ref	IDS	Dataset	Feature selection method	Number of features	Attacks detected	Detection method	Result analysis
Chou et al. (2008)	Network IDS	Six UCI Database, DARPA, KDD CUP 99	Correlation Based	21	R2L, U2R, Normal	Fuzzy belief k-NN, C4.5	DR and FAR before and after adding C4.5 decision rules: • Before: FPR = 9.59% DR = 83.21% • After: FPR = 3.11% DR = 83.21%
Shirazi (2009)	Anomaly	KDD CUP 99	Information Theory	8	DoS, R2L, U2R, Probe	SF-KNN, SUS-KNN	• DR and FPR for SF-KNN: 92.56% and 2% • DR and FPR for SUS-KNN: 92.84% and 4.52%
Wang et al. (2010)	Anomaly	KDD CUP 99	Clustering	Variable size of data samples	DoS, R2L, U2R, Probe	Fuzzy Clustering, ANN	• Average Accuracy: 96.71% • Training Time: 2125.4 s
Tang et al. (2010)	Anomaly	KDD CUP 99	IG	23	DoS, R2L, U2R,Probe	k-means, TASVM	• DR: 99.8% • Accuracy: 99.3% • FAR: 2.99%
Wang et al. (2011)	Hybrid. Misuse Anomaly	Simulated dataset	Clustering	24	BlackHole, Worm-Hole, SinkHole	Clustering, BPN	• DR: 90.6% • Accuracy: 91.26% • FPR: 2.06%
Su (2011)	Anomaly	Real-time network traffic	Clustering	35	Flooding Attacks	Weighted k-means	• Classification Accuracy: 95.36%

Table 7 (continued)

Ref	IDS	Dataset	Feature selection method	Number of features	Attacks detected	Detection method	Result analysis
Suresh and Anitha (2011)	Misuse	CAIDA dataset	Chi-Square, IG	8	DDos Attacks	k-NN, k-means, Fuzzy C-means Clustering	Classification Accuracy: • Fuzzy C-means: 98.7% • KNN: 96.6% • k-means: 96.7%
Muniyandi et al. (2012)	Network IDS	KDD CUP 99	K-Cluster	Variable size of data samples	DoS, R2L, U2R, Probe	C4.5+K Means	• TPR: 99.6% • FPR: 0.1 • Precision: 95.6% • Accuracy: 95.8% • F-score: 94.0%
Chitrakar and Huang (2012)	Network IDS	KDD CUP 99	K-Mediods	Variable size of data samples	DoS, R2L, U2R, Probe	NB+K Means	• Accuracy: 96.08% • DR: 96.21% • FAR: 3.11%
Yassin et al. (2013)	Anomaly	ISCX 2012	K-Cluster	Variable size of data samples	DoS, R2L, U2R, Probe	KMC+NB	• Accuracy: 99.0% • DR: 98.8% • FAR: 2.2%
Kumar et al. (2015)	Anomaly	DARPA	Gaussian Similarity	10	DoS, R2L, U2R, Probe	KNN	• Gaussian Similarity is used to reduce the dimension of the dataset. • Space efficiency is achieved by reducing the space complexity. • Time complexity is reduced by handling the high dimensional computations

Table 7 (continued)

Ref	IDS	Dataset	Feature selection method	Number of features	Attacks detected	Detection method	Result analysis
Lin et al. (2015)	Anomaly	KDD CUP 99	Attribute Ratio	19,6	DoS, R2L, U2R, Probe	CANN	• For 6 dimensional dataset: • Accuracy: 99.76% • DR: 99.9% • FAR: 0.003% • For 19 dimensional dataset: • Accuracy: 99.46% • DR: 99.28% • FAR: 2.95%

Table 7 (continued)

Ref	IDS	Dataset	Feature selection method	Number of features	Attacks detected	Detection method	Result analysis
Ambusaidi et al. (2015)	Anomaly	KDD CUP 99, NSL-KDD and Kyoto dataset	Redundancy Penalization	Variable number of features selected for each dataset	DoS, R2L, U2R, Probe	Modified Laplacian, 1NN, SVM	• Best accuracy achieved for KDD CUP 99 with 16 features and 1NN classifier: 87.33%. • Best accuracy achieved for KDD CUP 99 with 16 features and SVM classifier: 90.36%. • Best accuracy achieved for NSL-KDD with 16 features and 1NN classifier: 85.19%. • Best accuracy achieved for NSL-KDD with 16 features and SVM classifier: 89.35%. • Best accuracy achieved for Kyoto with 8 features and 1NN classifier: 96.38%. • Best accuracy achieved for Kyoto with 8 features and SVM classifier: 90.46%

Table 7 (continued)

Ref	IDS	Dataset	Feature selection method	Number of features	Attacks detected	Detection method	Result analysis
Muda et al. (2016)	Network IDS	KDD CUP 99	K-Cluster	Not mentioned	DoS, R2L, U2R, Probe	NB+K Means	• Accuracy: 99.8% • DR: 99.8% • FAR: 0.41%
Landress (2016)	Anomaly	KDD CUP 99	DT	11	DoS, R2L, U2R, Probe	k-means and SOM	• Classification Accuracy: 98.9%
Alhaj et al. (2016)	Anomaly	DARPA	IG	8	DoS, R2L, U2R, Probe	k-means, EM, Hierarchical clustering	Classification Accuracy: • k-means: 77.9% • EM: 90.6% • Hierarchical: 100%
Igbe et al. (2016)	Network IDS	NSL-KDD	IG	8	DoS, R2L, U2R, Probe	NSA-GA	• DR: 98.9% • FAR: 0.017
Harish and Kumar (2017)	Anomaly	KDD CUP 99	PCA	Not mentioned	DoS,R2L, U2R,Probe	RSKFCM	• Accuracy: 86.26% • FPR: 17.04%
Bostani and Sheikhan (2017)	Anomaly	NSL-KDD	k-means clustering	Variable number of samples	DoS, R2L, U2R, Probe	MOPF	• DR: 96.20% • FAR: 1.44%
Li et al. (2018)	Anomaly	KDD CUP 99	DPC	15	DoS, R2L, U2R, Probe	DPNN	• Accuracy is increased to 15% compared to the basic k-nn method for probe. • Efficiemcy is enhanced by 20.688%

Table 7 (continued)

Ref	IDS	Dataset	Feature selection method	Number of features	Attacks detected	Detection method	Result analysis
Khan et al. (2018)	Anomaly	NSL-KDD	Entropy	7	DoS Attacks	Clustering	• Proposed method and calculated granulaity and entropy values of selected features is presented
Bajtoš et al. (2018)	Anomaly	Simulated dataset	Not Applied	Not Applicable	Threat profiling	k-means, PAM, CLARA	•Decomposition graphs of the threat agents with varying number of clusters is presented
Verma et al. (2018)	Anomaly	NSL-KDD	Not Applied	Not Applicable	DoS, Probe, R2L, U2R	XGboost, Adaboost, k-means	• Accuracy: 84.2%
Chandra et al. (2019)	Anomaly	KDD CUP 99	CFS	6	DoS, Probe, R2L, U2R	k-means, SMO	• Accuracy: 84%
Xu et al. (2020)	Network-base	NSL-KDD	PCA	20	DoS, Probe, R2L, U2R	k-means	• Detection Rate: 90.21% • False Alarm Rate: 7.2%
Chen et al. (2020)	Network-based	KDD CUP 99	Standard Deviation	Not mentioned	DoS, Probe, R2L, U2R	k-means	• Accuracy : 98.6% • False Positive Rate: 0.625%
Henriques et al. (2020)	Host-based	NASA HTTP Log dataset	k-means	14	System attacks	k-means, XGboost	• Model Phase time : 250,000 Miliseconds

Research have shown considerably a better performance when combining the clustering methods with methods like ANNs and tree classifiers. For instance, in Muniyandi et al. (2012) *k*-means algorithm is used with decision tree classifier. The *k*-means clustering is used to derive the clusters from the training instances using similarity measures like Euclidean distance between the instances and the cluster center. The *k*-means algorithm is data influential method which performs a greedy search on the data to obtain local minima. In the second phase, a decision tree built for each cluster instances. By combining these methods two major problems of forced assignment and class dominance are eliminated (Muniyandi et al. 2012).

The forced assignment problem occurs when the value of *k* in the *k*-means algorithm chosen is very small compared to the actual groupings of the given dataset. Initializing with a small *k* value will ignore the overlapping instances within a group. Thus, grouping the data points with different characteristics to be part ofthe same cluster. These forced assignments lead to a rise in false positive and reduction in detection rate for intrusion detection. Another problem is when one cluster has a large number of data points of one class compared to other classes identified. This dominance of one class leads to a weak relationship with the other classes (Muniyandi et al. 2012). Thus, combining the clustering algorithm with other techniques resulted in better accuracy and detection rate.

Ambusaidi et al. (2015) modified Laplacian is combined with SVM and 1-NN. The features are reduced using redundancy penalization. The experiments are conducted on KDD CUP99, NSL-KDD, and Kyoto dataset. An accuracy of 87.33% and 90.36% is achieved when experiment is performed on KDD CUP 99 dataset with 16 features using 1-NN classifier and SVM classifier respectively. An accuracy of 85.19% and 89.35% is achieved when experiment is performed on NSL-KDD dataset with 16 features using 1-NN classifier and SVM classifier respectively. An accuracy of 96.38% and 90.46% is achieved when experiment is performed on Kyoto dataset with 8 features using 1-NN classifier and SVM classifier respectively. The dimension of the dataset can also be reduced for better classification.

In Kumar et al. (2015) the training and test set reduced by clustering the dataset according to the available class labels and then measuring the distance between all data samples and cluster centers using the Gaussian function as the similarity measure. At the end, all the distances between the clusters and their nearest neighbors are added for every data points to reduce the test set. In fact, clustering can be used as a feature selection approach for obtaining a reduced number of features for attack classification (Wang et al. 2011; Su 2011).

Feature selection methods such as chi-square, IG, and gain ratio can also be used with the clustering techniques like *k*-NN, *k*-means clustering, and fuzzy clustering for selecting the features. These methods of information theory effectively decrease the computation time without hampering the detection process (Tang et al. 2010; Chou et al. 2008; Shirazi 2009; Suresh and Anitha 2011). In fact, using information theory methods such as entropy and granular computing calculate weight of each feature and important features can be selected with the complexity of $\mathcal{O}(n \log n)$ (Khan et al. 2018).

A hybrid approach is developed using a *k*-medoids algorithm and NB classifier in Chitrakar and Huang (2012), wherein data instances are first clustered based on their behaviors and then classification is carried out using NB classifier. The *k*-medoids algorithm derived from the *k*-means algorithm where in place of the mean the center data point taken as a reference to cluster the instances. Therefore, grouping instances into cluster are dependent on the minimization of the sum of dissimilarity between the data point and center. The algorithm is executed in three steps. In the first step, iteration is carried out by defining *k*-medoids that represents the number of clusters; in the second step, the algorithms considered the non-medoids objects which can be defined by $(n - k)$, where n is the number of instances in the dataset (Chitrakar and Huang 2012). In the last step, swapping cost computed between the medoid and non-medoid which is a dissimilarity measure. The complexity of each step can be given by $\mathcal{O}(k(n - k)^2)$ (Chitrakar and Huang 2012). The implementation results are compared with the k-means algorithm and showed an increase in the accuracy and detection rate with the k-medoid algorithm.

The k-means clustering, Partition Around Mediods (PAM), and Clustering Large Applications (CLARA) algorithms are used for building security incident profiles to simplify the task of attack detection and classification (Bajtoš et al. 2018). The objective of the proposed method is to cluster threat agents based on the similarity using attributes of reported security events. The experiments were conducted on the data collected from the sensors deployed in the environment of the warden system and profiling with and without the analysis of outliers were performed.

The k-means clustering hybridized with NB in Yassin et al. (2013) and Muda et al. (2016), wherein k-means is used as feature selection method. In Yassin et al. (2013), the experiments are performed on ISCX dataset and accuracy rate of 99.0% and false alarm rate of 2.2% have been achieved. While in Muda et al. (2016), experiments are performed on KDD CUP 99 dataset and accuracy rate of 99.8% and false alarm rate of 0.41% is achieved. In Lin et al. (2015), k-means combined with ANN for classification and attribute ratio is used as the feature selection method. The experiments were conducted with a feature set having 6 and 19 features. The results showed that the reduced feature set gave better accuracy, detection rate, and false alarm rate.

A single alert cannot contribute significantly to detect the type of attack. Therefore, clustering can be used to map the relationship between the different alerts generated by the IDS for a given network environment. The pattern of different alerts can be combined based on the features identified to deduce the type of attack. In Verma et al. (2018), XGboost and Adaboost classifiers are used for attack classification. Here, in the proposed method k-means clustering is used to cluster the data points and then XGboost and Adaboost classifiers are used for classifying the corresponding data points. The experiments were performed using NSL-KDD dataset and 84.2% accuracy was achieved.

In Chandra et al. (2019), k-means clustering approach is used along with Sequential Minimal Optimization (SMO). Here, in the proposed method, k-means is used to cluster the data instances and then SMO is applied for attack classification. The proposed method selects 6 features out of 41 features by applying correlarion based feature selection. The experiments were conducted on KDD CUP 99 dataset and results obtained showed that proposed approach outperforms k-means and SVM algorithm in terms of accuracy.

A novel k-means algorithm is proposed in Xu et al. (2020), wherein initial seed of clustering is derived based on density. Here, Kd-tree algorithm is used for dividing the space and storing generalized information of clustering. Kd-tree algorithm prunes the search space and optimizes the operations of k-means clustering to speedup the process of prediction. The experiments are performed on NSL-KDD dataset. The clustering of data samples reveals that the clusters formed are stable and accurate when the number of clusters as well as iterations are kept constant. Moreover, principal components are chosen from the dataset using PCA. For experimentation, an optimal value of k is chosen as 20 with 20 dimensional vectors. The proposed approach achieved the detection rate of 90.21% and false alarm rate of 7.2%.

An efficient hybrid clustering technique is proposed in Chen et al. (2020), where advantages of quantum computing and swarm intelligence are integrated to develop an improved k-means clustering algorithm. The quantum inspired ant lion optimized algorithm is used to initialize and derive an optimal value of *k* for k-means algorithm and further, *k*-means is used for classification and prediction. The experiments are performed on KDD CUP 99 dataset and the dataset is divided into four sets using IBM SPSS stratified random sampling. The results are presented in terms of accuracy and false positive rate. The proposed approach achieved accuracy higher than 98% for all the four sets and average false positive rate of 0.625%.

A scalable framework using parallel computing environment is proposed in Henriques et al. (2020), wherein two models are constructed in parallel using *k*-means and XGboost algorithms. *k*-means algorithm is used for deriving the features using coherent clusters and XGboost model is used for interpreting the rules. The experiments are performed using the NASA HTTP log datasets.

A two-tier feature selection method is implemented in Alhaj et al. (2016), wherein IG is used to find and rank the significant features during the first phase and in the second phase, additional features are used to enhance the clustering accuracy. The selected features from DARPA dataset are implemented with *k*-means, Expectation Maximization (EM), and hierarchical clustering. The results revealed that the feature selection technique used is capable of identifying attack steps involved in executing a particular kind of attack to improve the clustering accuracy (Alhaj et al. 2016). The detection rate of the attacks can also be improved by clustering the network samples based on the neighborhood information.

A variant of the traditional fuzzy c-means algorithm proposed in Harish and Kumar (2017) and referred as Robust Spatial Kernel Fuzzy C-Means (RSKFCM) (Harish and Kumar 2017), that considers neighborhood membership information and kernel distance for grouping the network samples. The framework is divided into three steps: pre-processing, feature selection, and clustering. In the first step, data is pre-processed by eliminating the noise and duplicate records. In the second step, PCA is applied to the data instances for extracting the most significant features from the dataset. Finally, the data samples clustered using the RSKFCM algorithm. The experiments were conducted on the KDD CUP 99 dataset. Generally, traditional fuzzy c-means algorithm considers membership value or distance metric for clustering the data, but the RSKFCM algorithm showed an increase in the accuracy by clustering the data samples based on the neighborhood information. In fact, the traditional method uses Euclidean distance as the distance metric whereas, RSKFCM uses gaussian kernel distance that reduces the noise and increases the accuracy of the IDS model. The data samples can also be grouped based on their homogeneity using the *k*-means clustering algorithm and by applying density peak clustering for feature selection as in Li et al. (2018).

A Modified Optimum-Path Forest (MOPF) algorithm proposed in Bostani and Sheikhan (2017), where the unknown samples classified based on the distance between the unknown samples and the root of every sample. The study focuses on the centrality and the prestige factors that evaluate the interactions and relationships between the different nodes of the graph. The centrality of the nodes in the graph is measured using the Betweenness Centrality (BC) and prestige is measured using the Proximity Prestige (PR). The proposed IDS model is implemented in three steps: partitioning, pruning, and detection. The *k*-means clustering algorithm is used in the partitioning module for combining all the data samples from the dataset into groups that would further be used in the detection phase. The training sets formed by clustering the data samples are pruned by selecting the informative samples using the BC and PR metrics. These metrics used to identify the influential and most informative data samples from the training sets. The pruning module acts as the pre-processing step for the classifier. In the detection phase, classification and detection of attacks carried out using the MOPF classifier. The hybridization of clustering methods with classification methods leads to a high detection rate and reduced false alarm rate for a given IDS model.

The data clustering method, *k*-means is combined with DT classifier J48 in Landress (2016). Here, the DT classifier used for feature selection and self-organizing maps are used for reducing the number of false positives. The techniques based on the Artificial Immune System (AIS) are inspired by the human immune system. These techniques have the capability of solving diverse problems by self-learning and its memory. The Self-NonSelf (SNS) model is an AIS designed model that has been implemented for IDS (Igbe et al. 2016). The model has the capability of detecting insider as well as outsider attacks with an adaptive nature that utilizes its memory and self-learning capability for learning patterns and detecting attacks. It is initialized with random detectors that would recognize any anomalous behavior using the Negative Selection Algorithm (NSA).

The model proposed in Igbe et al. (2016) is implemented in six steps: data capture, feature selection, data pre-processing, detector generation, monitoring, and memory detector distribution. The model was evaluated using NSL-KDD dataset. The dataset consisting of 41 features and reduced to 8 features by applying IG. The selected features were pre-processed and converted into binary form depending on the classification technique used. In the detector generating step, NSA algorithm is used for performing an exhaustive search to find the optimal set of detectors to be used for detecting attacks. Thereafter, network traffic continuously monitored for classifying normal or anomalous traffic flowing in the network. The developed IDS model is distributed, and hence, require a set of detectors to analyze the network traffic at different nodes. A distributed IDS maintains a detector table at each node. The particpating nodes share and update network traffic information among each other in the memory detector distribution phase. The experimental results obtained are compared with other methods like J48, SVM, and NB. The proposed method gave 98.9% detection rate and 1.7% false alarm rate.

4.3 Deep learning based techniques

Large datasets lead to multiple classifications of data and decrease in the efficiency of the IDS model. Moreover, shallow learning is not capable of performing an in-depth analysis of the high dimensional datasets. In contrast, DL techniques have capability to handle the high-dimensional data and develop models to extract information in a more refined and better way (Pareek and Thakkar 2021; Thakkar and Chaudhari 2020c). The rapid advancement in the field of deep learning offered a completely reformed way to build intelligent IDS. The growth in the computational resources have given recognition to DL techniques such as Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNN). These techniques have been used for building efficient IDS models (Lohiya and Thakkar) (Sheikhan et al. 2012; Yin et al. 2017).

A primary difference between the Feed Forward Neural Network (FFNN) and RNN is that an FFNN consist of connections that go from an input node to the output node. In contrast, RNN consist of a recurrent neuron often referred to as a feedback connection. RNN networks have the capability of storing information with the help of feedback connection. It can handle sequential data of arbitrary length very easily (Thakkar and Chaudhari 2021). Therefore, RNN can easily handle the time-series data of the network traffic for intrusion detection. A major disadvantage of RNN is that it does not work well with back-propagating through time, and it can handle very limited contextual information. This drawback is overcome by Long Short Term Memory (LSTM) which has the ability to bridge the time intervals and backflow errors. Similar to LSTM, GRU uses the gating mechanism for iteratively updating the memory and error.

CNN is designed for the data that are multidimensional and hierarchical in structure. It builds strong local correlations between the data but does not work well if the data does not have any positional information. Thus, the performance of DL methods used for IDS is based on the number of epochs and the number of nodes in the layers. A summary of DL based methods for IDS derived from the reviewed articles is presented in Table 8.

IDS models built using the DL techniques are represented in a layered structure and require long training time for learning and classifying attacks. A three-layered RNN is designed in Sheikhan et al. (2012). Here, the input layer consists of 41 nodes based on the number of features that are categorized by the KDD CUP 99 dataset as basic, content, time-based, and host-based traffic features while the output layer consistsof 5 nodes based on the attack types (normal, DoS, Probe, R2L, U2R). The proposed model has two hidden layers that are based on the categorization of features (Sheikhan et al. 2012). A descripted model is built using RNN, bi-directional RNN, LSTM, and bi-directional LSTM for binary as well as multi-class classification in Elsherif et al. (2018). The proposed work aims at classifying unknown threats with low false alarm rate. The experiments are performed on NSL-KDD dataset and results are shown with one and two hidden layers with varying number of neurons.

To meet the computational needs of an IDS, an ANN model is developed that implements IDS in four stages: monitoring the network, detecting the anomalous behavior, classifying the attack, and generating an alert. For carrying out the procedure, pcap tool is used to capture the incoming and outgoing packet in the network and filters the traffic based on protocol type and IP address (Al-Janabi and Saeed 2011).

To address the issue of high computation time, a layered approach for classifying attacks has been proposed in Devaraju and Ramakrishnan (2014). The proposed model implemented four neural network approaches: FFNN, Generalized Regression Neural Network (GRNN), Probabilistic Neural Network (PNN), and Radial Basis Neural Network (RBNN) (Devaraju and Ramakrishnan 2014). The four-layered architecture is represented as layer 1 for DoS, layer 2 for probe, layer 3 for R2L, and layer 4 for U2R. The results are compared with other machine learning classifiers which shows improvement in the efficiency of the proposed model (Devaraju and Ramakrishnan 2014). In fact, to increase the detection rate and accuracy of the model feature set has been used as a vector where basic and traffic based features are selected to identify a particular category of attack such as probe (Ahmad and Alghamdi 2009; Norouzian and Merati 2011; Tang et al. 2018).

Dependency ratio is used for selecting features in Kim and Kim (2015). Here, dependency ratio of every feature is evaluated with respect to every other feature in the dataset, and most significant features are selected. These features are then fed as inputs to the RNN (Kim and Kim 2015). The performance of the IDS model depends on the feature design and feature set that can accurately classify the network traffic. The incapability of the selected feature set to classify the network data leads to a high false alarm rate. Representation learning approach such as Hierarchical Spatial-Temporal features-based Intrusion Detection System (HAST-IDS) proposed in Wang et al. (2018) can be used to address this issue. The proposed method studied the spatial features of the data using CNN and temporal features using LSTM. The feature learning process is carried out automatically that improves the detection capability compared to manually designed features. The experimental results revealed the effectiveness of the proposed method for feature learning and reduced the false alarm rate.

Table 8 Deep learning based methods for IDS

Ref	IDS	Dataset	Feature selection method	Number of features	Attacks detected	Detection method	Result analysis
Sheikhan et al. (2012)	Anomaly IDS	KDD CUP 99	Feature grouping	Variable number of features	DoS, R2L, U2R, Probe	RNN	• DR: 94.1% • FAR: 0.38% • CPE: 0.166
Devaraju and Ramakrishnan (2014)	Anomaly	KDD CUP 99	Layered Approach	36	DoS, R2L, U2R, Probe	FFNN, GRNN, RBNN, PNN	• FFNN: Efficiency: 97.35%, FAR: 2.65% • GRNN: Efficiency: 87.54%, FAR: 12.46% • PNN: Efficiency: 96.66%, FAR: 3.34% • RBNN: Efficiency: 93.05%, FAR: 6.95%
Kim and Kim (2015)	Anomaly	DARPA	Dependency Ratio	32	DoS, R2L, U2R, Probe	RNN	• DR: 95.37% • FAR: 2.1%
Yin et al. (2017)	Anomaly	KDD CUP 99	Layered Approach	Variable number of features in each layer	DoS, R2L, U2R, Probe	RNN	• Binary classification • Accuracy: 99.8% • Time: 5516*s* • Multi-class classification • Accuracy: 99.5% • Time: 11444*s*
Tang et al. (2018)	Anomaly	NSL-KDD	Basic and Traffic based features	6	DoS, R2L, U2R, Probe	GRU-RNN	• Accuracy: 89% • ROC analysis has been shown
Kim and Cho (2018)	Anomaly	Yahoo S5 Webscope dataset	Spatial and temporal information	67 files of A1 class of the dataset	Web traffic anomaly	C-LSTM	• Accuracy: 98.6% • Recall: 89.7%

Table 8 (continued)

Ref	IDS	Dataset	Feature selection method	Number of features	Attacks detected	Detection method	Result analysis
Imamverdiyev and Abdullayeva (2018)	Anomaly	NSL-KDD	Gaussian distribution	38	DoS	Gaussian-Bernoulli RBM	• Accuracy: 73.23% • ROC anlysis has been shown
Zhou (2018)	Anomaly	PE files dataset	RNN	9	Malware detection	CNN	• Accuracy: 97.3%
Wang et al. (2018)	Anomaly & Misuse	DARPA, ISCX-2012	Spatial and temporal learning	16	DoS, R2L, U2R, Probe	HAST-IDS	• DARPA • Accuracy: 99.6% • DR: 97.7% • FAR: 0.07% • ISCX2012 • Accuracy: 99.6% • DR: 96.1% • FAR: 0.22%
Naseer et al. (2018)	Anomaly	NSL-KDD	Auto-encoder	Variable size	DoS, R2L, U2R, Probe	CNN,RNN	• ROC analysis has been shown
Elsherif et al. (2018)	Anomaly	NSL-KDD	Not Applied	Not Applicable	DoS, Probe, R2L, U2R	RNN, BRNN, LSTM, BLSTM	• Results in terms of training time graph is presented with varying number of neurons
Xu et al. (2018)	Network-based	NSL-KDD	Not applied	Not Applicable	DoS, Probe, R2L, U2R	RNN-GRU, MLP	• DR: 99.31% • FPR: 0.84%
Gurung et al. (2019)	Anomaly	NSL-KDD	Sparse auto encoder	Variable size	DoS, Probe, R2L, U2R	Logistic classifier	• Overall accuracy: 87.2%
Al-Emadi et al. (2020)	Network-based	NSL-KDD	Not Applicable	Not Applicable	DoS, Probe, R2L, U2R	CNN, RNN	• Accuracy of CNN: 97.01% • Accuracy of RNN-LSTM: 81.60% • Accuracy of RNN-GRU : 50.25%

Table 8 (continued)

Ref	IDS	Dataset	Feature selection method	Number of features	Attacks detected	Detection method	Result analysis
Elmasry et al. (2020)	Network-based	NSL-KDD, CIC-IDS-2017	PSO	NSL-KDD: 10, CIC-IDS-2017: 23	Attack categories defined in datasets	DNN, LSTM, DBN	•For NSL-KDD: • Accuracy of DNN: 97.72% • Accuracy of LSTM: 98.8% • Accuracy of DBN: 99.79% •For CIC-IDS-2017: • Accuracy of DNN: 97.85% • Accuracy of LSTM: 98.83% • Accuracy of DBN: 99.91%
Hindy et al. (2020)	Network-based	NSL-KDD, CIC-IDS-2017	Correlation	Not mentioned	Attack categories defined in datasets	Auto-Encoder	•For NSL-KDD: • 92.96% • For CIC-IDS-2017: • Accuracy for DoS: 90.01% • Accuracy for DoS-Hulk: 98.43% • Accuracy for Port Scan: 98.47% • Accuracy for DDoS: 99.69%
Hassan et al. (2020)	Network-based	UNSW-NB15	CNN	Not mentioned	DoS, exploits, worms, backdoor, shellcode	LSTM	• Accuracy: 97.17%

Table 8 (continued)

Ref	IDS	Dataset	Feature selection method	Number of features	Attacks detected	Detection method	Result analysis
Dutta et al. (2020)	Network-based	UNSW-NB15	AutoEncoder	Not mentioned	DoS, exploits, worms, backdoor, shellcode	DNN	• Accuracy: 91.29%
Susilo and Sari (2020)	Network-based	BoT-IoT	Not Applied	Not Applicable	DoS	RF, CNN, MLP	• RoC Curve for RF: 0.99 • RoC Curve for CNN: 0.99 • RoC Curve for MLP: 0.80
Gamal et al. (2020)	Network-based	KDD CUP 99	CNN	Not mentioned	DoS, Probe, R2L, U2R	SVM, *k*-NN	• Detection accuracy: 99.3%

Similarly, sparse auto encoder is used as unsupervised feature learning method in Gurung et al. (2019). Here, in the proposed method, logistic classifier is used that learns the features and also adjusts itself according to learned patterns for detecting intrusions. The experiments were conducted on NSL-KDD dataset and 87.2% overall accuracy is achieved. The performance of the IDS model can be enhanced by correlating the current and previous events. This has been depicted in Tang et al. (2018), where GRU-RNN is used with six raw features from NSL-KDD dataset in Software Defined Network (SDN) architecture. The proposed model yields the accuracy of 89% and possess the capability to detect intrusion in SDN.

The datasets of IDS consists of discrete and continuous features. Better results can be obtained by using the DL methods such as deep Restricted Boltzmann Machine (RBM) model. The RBM model can handle the continuous network traffic data by replacing the probability distribution in the layers with Gaussian distribution (Imamverdiyev and Abdullayeva 2018). A comparative analysis of DL techniques such as Bernoulli-Bernoulli RBM, Gaussian-Bernoulli RBM, Deep Belief Network (DBN) for detecting DoS attacks presented in Imamverdiyev and Abdullayeva (2018). The experimental results of the proposed model compared with other ML methods such as SVM (radial basis), SVM (epsilon-SVR), and decision tree. The results showed that Gaussian-Bernoulli RBM outperformed to other methods.

The DL methods also applied in detecting the web traffic anomalies that consist of time series data. A C-LSTM DL method is proposed for efficient learning the spatial and temporal features present in the network traffic (Kim and Cho 2018). The proposed method automatically extracts features from the raw network traffic data by combining CNN, LSTM, and Deep Neural Network (DNN). For reducing the frequency variation in spatial information CNN layer is used, the time series information is modeled using the LSTM layer, and data mapping into high dimensional space is achieved by the DNN layer. The proposed method classifies and extracts features for detecting anomalies in the web traffic data.

CNN also used with the portable executable files dataset for malware detection where RNN used for feature selection (Zhou 2018). The hybridization of RNN and CNN have been applied to NSL-KDD dataset for intrusion detection (Naseer et al. 2018). A DL model is proposed using GRU as the main memory units for RNN, combined with MLP for classying network intrusion (Xu et al. 2018). The experiments performed on NSL-KDD dataset and results showed that using GRU as the memory unit of RNN gave better performance compared to LSTM for detecting intrusions.

A intelligent IDS is designed with CNN and RNN in Al-Emadi et al. (2020) for attack detecting and classification using the NSL-KDD dataset. Here, two type of RNN models are implemented namely, LSTM and GRU-RNN. The DL architectures such as CNN have the characteristics property of extracting features from the dataset. The experimental results revealed that CNN outperforms both the models of RNN namely, LSTM and GRU-RNN in terms of accuracy.

An optimized DL-based IDS is proposed in Elmasry et al. (2020), wherein PSO is used for feature selection and hyperparameter optimization and classification is performed using DL techniques such as DNN, LSTM, and DBN. The experiments are performed using NSL-KDD and CIC-IDS-2017 datasets and results are presented in terms of detection rate and false alarm rate. A novel Auto Encoder based DL model is proposed in Hindy et al. (2020), for detecting zero-day attacks. The experiments are performed using NSL-KDD and CIC-IDS-2017 datasets and simulations are performed considering various threshold value which is chosen by random search hyperparameter optimization.

A weight dropped LSTM network is proposed in Hassan et al. (2020) for network-based intrusion detection. Here, in the proposed network, CNN is used for feature selection and weight dropped LSTM is used to retain the long term dependencies among the selected features. Experiments are performed using UNSW-NB15 dataset and results are presented using accuracy as performance measure. A hybrid DL model for intrusion detection is proposed in Dutta et al. (2020), wherein classical AutoEncoder is used for feature engineering and DNN is used for attack detection and classification. The experiments are performed using UNSW-NB15 dataset and hybrid model achieves an accuracy of 91.29%.

A DL-based IDS model is designed to detect DoS attacks in Susilo and Sari (2020). Here, the experiments are performed using the BoT-IoT dataset with RF, CNN, and MLP classifiers. Experimentation results showed that CNN outperforms other classifier in detecting and classifying intrusions. In Gamal et al. (2020), CNN is used for features engineering whereas SVM and *k*-NN are used for classification. For experimental analysis, KDD CUP 99 dataset is used and results showed that the proposed approach achieved detection accuracy of 99.3%

4.4 Swarm and evolutionary algorithms based techniques

Even though most of the ML and DL techniques perform with great ease with the network, these methods are not very well equipped with the constantly changing network scenarios, novel and complex patterns. Therefore, they are not capable of identifying new attack patterns (Thakkar and Lohiya 2020a). It is not necessary that the new training data obtained would be having fully labeled data and hence, it becomes a tedious task to manually label and classify the data. This raises the demand for proactive techniques which can work in a constantly changing environment and are capable of identifying unknown attacks.

SWEVO algorithms has gained momentum through the way they handle the global optimization problem. These algorithms scale well with large datasets and are well equipped to handle the problem with noisy evaluation functions (Thakkar and Chaudhari 2020a). SWEVO possess the flexibility to optimize and change the procedures and are self-adaptive to find optimal solutions for the given problem (Chaudhari and Thakkar 2019c). We have studied SWEVO and how they play a significant role in selecting an optimal set of features from the given IDS dataset. The use of SWEVO for feature selection can enhance the performance of the underlying model and detection accuracy (Thakkar and Lohiya 2020b).

Apart from stating the advantages of the algorithms in feature selection, it is also necessary to deduce that all these algorithms try to generate an optimal feature subset along with maintaining the classification accuracy of the underlying classifier (Fries 2008). The optimality provided by the meta-heuristics algorithms can be coined in two ways: i)obtaining the best types of problems that can be solved by the algorithm considered, and ii) finding the best algorithm that can produce the set of solutions for the problem considered. There is a high scope of study in regard to applying feature selection in IDS using SWEVO because of the self-learning, self-adapting, and handling multi-objective problems capabilities of these algorithms (Thakkar and Lohiya 2020b). A summary of SWEVO-based methods for feature selection in IDS derived from the reviewed articles is presented in Table 9.

Swarm and Evolutionary algorithms offer such an ability to overcome the limitations of existing techniques. For instance, Genetic Algorithms (GAs) have characteristics to

Table 9 Swarm and evolutionary algorithm based methods for IDS

Ref	IDS	Dataset	Feature selection method	Number of features	Attacks detected	Detection method	Result analysis
Fries (2008)	Misuse	KDD CUP 99	GA	8	DoS, R2L, U2R, Probe	Fuzzy Sets	• DR: 99.6%, FAR: 0.2%
Wang et al. (2009)	Network IDS	KDD CUP 99	PSO	5	DoS, R2L, U2R, Probe	SVM	• Accuracy: 99.8%
Mabu et al. (2011)	Network IDS	DARPA	GNP	41	DoS, R2L, U2R, Probe	Fuzzy Sets	• DR: 94.4%, FAR: 5%
Hoque et al. (2012)	Misuse, Anomaly	KDD CUP 99	GA	32	DoS, R2L, U2R, Probe	GA	• DR: 95%, FPR: 3%
Kannan et al. (2012)	Network IDS	KDD CUP 99	GA	17	DoS, R2L, U2R, Probe	SVM + Fuzzy Sets	• DR: 98.5%, ER: 3.13%
Ganapathy et al. (2013)	Network IDS	KDD CUP 99	PSO	19	DoS, R2L, U2R, Probe	SVM + Rule based	• Detection accuracy of DoS: 99.3%, Probe: 99.3%, Others: 71.9%
Kuang et al. (2014)	Network IDS	KDD CUP 99	KPCA + GA	20	DoS, R2L, U2R, Probe	SVM	• DR: 96.3% • FAR: 0.98% • CC: 0.96
Eesa et al. (2015)	Misuse, Anomaly	KDD CUP 99	CFO	Top significant features were selected in the range from (5 – 35)	DoS, R2L, U2R, Probe	DT	• Best results are obtained when the number of features is equal to or less than 20. • DR: 91.5% • FPR: 3.4% • AR: 92.3%
Tama and Rhee (2015)	Network IDS	KDD CUP 99	PSO	17	DoS, R2L, U2R, Probe	C4.5, Random Forest, CART	• DR: 98.9%, FAR: 2.1%
Enache and Sgârciu (2015)	Network IDS	NSL-KDD	Improved BA	17	DoS, R2L, U2R, Probe	SVM	• DR: 97.6%, FAR: 0.98%

Table 9 (continued)

Ref	IDS	Dataset	Feature selection method	Number of features	Attacks detected	Detection method	Result analysis
Aghdam and Kabiri (2016)	Anomaly	KDD CUP 99, NSL-KDD	ACO	24	DoS, R2L, U2R, Probe	ACO	• Accuracy: 98.9%, FPR: 2.5%
Aslahi-Shahri et al. (2016)	Anomaly	KDD CUP 99	GA	10	DoS, R2L, U2R, Probe	SVM	• TPR: 0973, FPR: 0.017
Aburomman and Reaz (2016)	Anomaly	KDD CUP 99	PSO, LUS, WMA	Variable size subsets	DoS, R2L, U2R, Probe	SVM,k-NN	• PSO based accuracy: 92.5% • LUS based accuracy: 92.7% • WMA based accuracy: 88.6%
Khammassi and Krichen (2017)	Network IDS	KDD CUP 99, UNSW-NB15	GA-LR	variable size of subsets	DoS, R2L, U2R, Probe	DT	• For KDD CUP 99 accuracy: 99.4% • For UNSW-NB15 accuracy: 92.7%
Raman et al. (2017)	Network IDS	KDD CUP 99	HG-GA	Variable size of features	DoS, R2L, U2R, Probe	SVM	• DR: 97.13%, FAR: 0.83%
Vardhini and Sitamahalakshmi (2017)	Anomaly	NSL-KDD	ACO	22	DoS, R2L, U2R, Probe	ACO	• Accuracy: 98.5%
Kabir et al. (2017)	Anomaly	NSL-KDD	GSA	16	DoS, R2L, U2R, Probe	Bayesian network	• Accuracy: 98.26% • Precision: 98.9% • TPR: 98.3% • FPR: 0.7%
Hamamoto et al. (2018)	Network IDS	Real network traffic	GA	6	IP flow anomaly	Fuzzy Logic	• Accuracy: 96.5% • Precision: 95.23% • Recall: 76.5% • F-measure: 84.84% • FPR: 0.56%

Table 9 (continued)

Ref	IDS	Dataset	Feature selection method	Number of features	Attacks detected	Detection method	Result analysis
Vijayanand et al. (2018)	Anomaly	WMN	GA	20	Attack detection	SVM	• Accuracy: 95.7%, FPR: 0.019%
Ali et al. (2018a)	Anomaly	KDD CUP 99	Not Applied	Not Applicable	DoS, Probe, R2L, U2R	Fast learning network, PSO	• Accuracy graph with varying number of neurons in hidden layer is presented
Ali et al. (2018b)	Anomaly	NSL-KDD	Not Applied	Not Applicable	DoS, Probe, R2L, U2R	Extreme Learning Machine, PSO	• Accuracy graph with varying number of neurons in hidden layer is presented
Kondaiah and Sathyanarayana (2018)	Anomaly	Simulated dataset	Not Applied	Not Applicable	Black hole attack	GA, FA, PSO	• 0.04319 sec, 0.691, and 0.769 for delay, detection rate, and throughput, respectively
Rais and Mehmood (2018)	Anomaly	KDD CUP 99	ACO	6	DoS, Probe, R2L, U2R	SVM	• Accuracy: 98.7%
Elhag et al. (2019)	Anomaly	KDD CUP 99	GA	Not mentioned	DoS, Probe, R2L, U2R	Fuzzy associative classifier	• Accuracy: 97.8%
Hosseini and Zade (2020)	Network-based	NSL-KDD	Multi-parent mutation + SVM	4	DoS, Probe, R2L, U2R	ANN + PSO + hybrid gravitational search	• Detection accuracy: 99.3%
Lv et al. (2020)	Misuse-based	KDD CUP 99, UNSW-NB15	PCA	Not mentioned	Attack categories defined in dataset	ELM + DE + Gravitational Search	• Accuracy for KDD CUP 99 : 96.59% • Accuracy for UNSW-NB15 : 89.01%

Table 9 (continued)

Ref	IDS	Dataset	Feature selection method	Number of features	Attacks detected	Detection method	Result analysis
Kalita et al. (2020)	Network-based	KDD CUP 99	Not Applied	Not Applicable	DoS, Probe, R2L, U2R	SVM, PSO	• Precision-Recall curve
Sarvari et al. (2020)	Network-based	NSL-KDD	Mutation Cuckoo Fuzzy	22	DoS, Probe, R2L, U2R	Evolutionary Neural Network	• Accuracy: 98.81%
Ghosh et al. (2020)	Network-based	KDD CUP 99	GA	7	DoS, Probe, R2L, U2R	MLP	• Accuracy : 97.29%
Liu et al. (2020)	Network-based	KDD CUP 99	GA, IG	6	DoS, Probe, R2L, U2R	DT	• TPR: 99.7% • FPR: 0.2%
Kumar (2020)	Network-based	NSL-KDD, ISCX 2012	Not Applied	Not Applicable	Attack categories defined in the datasets	Neural Network	•For NSL-KDD : • Detection Accuracy: 97% • FPR: 2% •For ISCX 2012: • Detection Accuracy: 88% • FPR: 2.4%

overcome the noise in the data, self-learning capability, and tendency to derive rules without any prior knowledge about the data (Kannan et al. 2012). In Fries (2008), GAs are used to optimize the featureselection process. In the first stage, clusters are created using Euclidean distance as the similarity measure and nearest neighbor as the classifier. The later stage includes combining the GA with these clusters to obtain an optimal result. To optimize the features, each chromosome is considered as a set of *k* bits where each bit is representing a cluster. For instance, if the chosen cluster is present at a given position in the set, the value of the chromosome is set as "1" else it is "0". A new set of clusters are obtained by applying the crossover and mutation operations and fitness of these clusters measured by the inter and intra-cluster distance.

In Hoque et al. (2012), GA is used with SVM and fuzzy sets to find features from the dataset and the search process for selecting the features is initiated with all the features or null set; the features are added one by one, but these methods would be caught in local minima, and therefore, random search process is carried out for obtaining an optimal set of features. The fitness value of the features calculated using the crossover and mutation operators and features with the highest fitness value is chosen and fed to the classifier.

GAs can also be used to optimize the parameters of a classifier. For instance, kernel-based PCA is used to extract features and GA is used to optimize the parameters of SVM using the mean absolute percentage error rate as the fitness function (Kuang et al. 2014). Each of the parameters is encoded in binary and represented by the chromosome. The chromosome value is considered as "1" if the parameter is selected and "0" otherwise (Kuang et al. 2014).

A multi-objective evolutionary fuzzy system is proposed in Elhag et al. (2019). Here, in the proposed method, fuzzy associative classifier is combined with GA for rule selection. The experiments are performed on KDD CUP 99 dataset. The advantage of using the proposed method is its efficiency to respond during inference and its ability to analyze the rules associated with attack detection and classification.

A general procedure for selecting features includes feature normalization and feature scaling as the dataset have continuous and discrete variables. To address this limitation and minimize the computation, genetic network programming can be implemented which is an extended version of the genetic algorithm and genetic programming (Mabu et al. 2011). Unlike strings in GAs and trees in genetic programming, directed graphs with the compact structure are used in genetic network programming that enables the ability to reuse the nodes of the derived graph. The method proposed in Mabu et al. (2011) uses genetic network programming with fuzzy association rule mining which is capable of dealing with both continuous and discrete features that can be considered as ideal for the real network scenarios. To maintain the information of the dataset in a complete form, it implements sub-attribute utilization.

GA is implemented along with MLP in Ghosh et al. (2020) for building IDS. Here, GA is used for extracting features from dataset and MLP is used for classification of network traffic as normal or attack. The experiments are performed using KDD CUP 99 dataset containing 41 features from which 7 features were selected using GA. In Liu et al. (2020), GA is integrated with IG for selecting features to detect and classify attacks. Here, in the proposed approach, IG is used to measure the feature importance and then features are arranged according to exponential increase in feature importance. Further, reduced feature set is used by DT algorithm to detect and classify attacks. The experiments are performed using KDD CUP 99 dataset.

A hybrid approach of multi-objective GA and neural network is proposed in Kumar (2020). Here, the proposed approach operates in two phases. In the first phase, the hybrid approach derives a set of non-dominating solutions of the base techniques namely, multi-objective GA and neural network. Whereas, in second phase ensemble solutions are derived. Further, non-dominating solutions and ensemble solutions are aggregated using majority voting classifier. The proposed approach is evaluated using NSL-KDD and ISCX 2012 datasets. The results showed that, hybrid approach achieves detection accuracy of 97% for NSL-KDD dataset and 88% for ISCX 2012 dataset. Moroever, FPR of 2% and 2.4% respectively, obtained for NSL-KDD and ISCX 2012 dataset.

Swarm intelligence approaches such as Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) can also be used for feature selection. These methods are inspired by natural phenomenon such as the communication between the bees and foraging behavior of flocks of birds. Many algorithms have been evolved based on these phenomena such as Queen bee Jung (Ganapathy et al. 2013) was formed based on the reproduction process of the queen bee. This technique was an improvement over the genetic algorithms by enhancing the exploitation and exploration process.

In Xu et al. (2008), a hybrid algorithm of particle swarm and a genetic algorithm was proposed to address the problem of DNA sequence formation under thermodynamic constraints which ensured the randomness of the population by assigning the queen bee as the optimal population. A random population was generated using the exploitation and exploration process of GAs. Inspired by the waggle dances of bees, the problem of the routing of vehicles was addressed by implementing decentralized method at different layers through the BeeHive algorithms (Wedde et al. 2007). To overcome the limitations related to the architecture of conventional networks, the swarm based algorithms can be used for managing networks of internet protocol (Wedde et al. 2007).

A two phase IDS is proposed in Hosseini and Zade (2020), that consists of feature selection phase and detection phase. For feature selection phase, a wrapper based technique namely multi-parent mutation along with SVM is proposed, whereas for detection phase ANN is used which is optimized using hybrid gravitation search and PSO. Experiments are performed using NSL-KDD dataset and the proposed approach records accuracy of 99.3%. A novel misuse-based IDS is proposed in Lv et al. (2020), wherein gravitational search and Differential Evolution (DE) are used to optimize the parameters of Extreme Learning Machine (ELM) classifier and kernel-based PCA is used for feature extraction. The experiments are performed on UNSW-NB15 and KDD CUP 99 dataset.

PSO is used for optimizing parameters of SVM for intrusion detection in Kalita et al. (2020). Here, the parameters of SVM are chosen using multi-objective PSO and SVM is used for attack detection and classification. The experiments are performed using KDD CUP 99 dataset and results are presented in the form of precision-recall curve. To enhance the IDS efficiency, mutation cuckoo fuzzy algorithm is proposed for selecting features in Sarvari et al. (2020). Here, in the proposed work, evolutionary neural network is used for classification. The experiments are performed using NSL-KDD dataset. The feature set is reduced to 22 features from 41 and further, the reduced set is divided in three priority sets based on the feature importance value.

To address the low redundancy coverage and reliable communication properties of WSN, ACO was used to optimize the process of cluster head selection in Thakkar and Kotecha (2011). ACO was combined with SVM in Gao et al. (2005) to optimize the process of feature selection. The experiments were carried out on the benchmark dataset KDD CUP 99 and the results showed a considerable amount of improvement in detection accuracy and computation time. In Li et al. (2011), ant colony and fuzzy clustering were implemented to detect intrusions which improved the detection rate by efficiently improving the problem of local minima and dynamically deriving the clusters with their center.

In Kabir et al. (2012) ant colony based feature selection is used to identify the size of the subset of features from the available set of features. The method was capable of searching globally, as the fitness value was iteratively updated for each features. In Rais and Mehmood (2018), a three level update ACO-based feature slection is used with SVM for binary as well as multiclass classification. The experiments performed on KDD CUP 99 dataset and 6 out of 41 features are selected for classification. The results compared with other feature selection techniques and results showed that ACO based feature selection outperforms other feature selection methods. Similarly, PSO was used to optimize the parameters of SVM for detecting intrusions using four benchmark datasets (Tian and Gu 2010). To measure the accuracy, the ROC curve was derived with the best combination of true positive and false positive rate and results showed the effectiveness in detecting anomaly (Tian and Gu 2010).

A fast learning network based IDS is proposed in Ali et al. (2018a), where PSO is used for optimizing the weights of the fast learning network. In the PSO based optimization for fast learning network, each particle represents one candidate solution for the weights of the network. Performing optimization using PSO is a challenging task as it requires to select both weights and number of neurons in the hidden layer to improve the accuracy of the proposed system. The experiments were performed on KDD CUP 99 dataset and accuracy with varying number of neurons presented. PSO is also used optimizing ELM classifier in Ali et al. (2018b). Here, the experiments performed on NSL-KDD dataset and results compared with traditional ELM classifier. The results showed that with varying number of neurons in the hidden layer, the accuracy of proposed hybrid approach is better compared to traditional ELM classifier.

Firefly algorithm is also a swarm-based algorithm inspired by the lightening characteristic of the fireflies. It is used because of its randomness and attractiveness properties. In Emary et al. (2015), firefly algorithm is used with the *k*-NN classifier to identify the feature subset from the KDD dataset. It finds the combination of features and determines the fitness value of the feature. Based on the fitness value of the feature, the feature with the highest value is selected. The proposed approach proved to be efficient than the PSO and GA algorithms in terms of detection and feature reduction. Another swarm-based algorithm is bat algorithm and used with different tree classifiers and SVM to find the features from the given set (Enache and Sgârciu 2015).

The swarm-based algorithm named ACO has been used for feature selection and classification in Aghdam and Kabiri (2016). In the proposed method, the IDS model consists of modules such as feature extraction and feature selection. The feature extraction is performed to convert the tcpdump data into a feature vector and feature selection is used to select more informative features from the dataset. ACO algorithm used for exploring the feature space and applying the evaluation function for measuring the classification. The best feature set found by ACO is used for the classification of attacks in IDS. The SWEVO algorithms are efficient and effective in classifying and deriving a significant set of features (Aghdam and Kabiri 2016).

Grouping of classifiers is often referred to as an ensemble classifier. An ensemble framework proposed in Aburomman and Reaz (2016) by combining PSO with SVM and k-NN for generating weights to develop an ensemble of classifiers that yields better detection rate for IDS. The proposed method uses local unimodal sampling for optimizing the behavioral parameters of PSO. The ensemble framework constructed using the classifier such as SVM, k-NN along with the weighted majority algorithm. The experimental results showed that the ensemble approach of SVM, k-NN, and PSO outperforms the weighted majority algorithm in terms of classification accuracy (Aburomman and Reaz 2016).

Similarly, a hybrid approach used for enhancing the performance of IDS in Khammassi and Krichen (2017), wherein GA is combined with Logistic Regression (LR) for improving the performance of the IDS. GA is used as a wrapper-based feature selection method to find the best subset of features and LR is used as the classification algorithm. The experiments are carried out on KDD CUP 99 and UNSW-NB15 dataset (Duarte and Farruca 2010). The efficiency of the selected features measured using the decision tree classifier and the results were compared with the other feature selection method (Khammassi and Krichen 2017).

The GA is also hybridized with SVM for IDS in Aslahi-Shahri et al. (2016). The framework proposed in Aslahi-Shahri et al. (2016) selects 10 significant features from 41 features of KDD CUP 99 dataset. The GA divides the features into three priorities based on the importance of the features. The features with the highest importance are given the first priority and features with the least importance are given the third priority. The feature has 4 features in first priority, 4 features in second priority, and 2 features in third priority. The experimental results showed that the proposed algorithm gave a true positive rate of 0.97 and a false positive rate of 0.017 (Aslahi-Shahri et al. 2016).

Apart from grouping the classifiers for developing ensemble approach, advanced tools and methods have been proposed to ensure the security of the network environment. However, the advantages and drawbacks of these methods make the development of IDS challenging. For instance, an adaptive technique for IDS developed in Raman et al. (2017) using the Hypergraph based Genetic Algorithm (HG-GA) with SVM. Here, the HG-GA method used for parameter optimization and feature selection. The population is initialized using the hyper-clique property of HG which speeds up the search process for an optimal solution. The proposed method utilizes the weighted objective function to balance the maximization in detection rate and minimization in the false alarm rate. The HG-GA method with SVM evaluated using the NSL-KDD dataset with all features and selected features obtained through the HG-GA method.

The applicability of computer networks pose the need for ensuring integrity and availability to users and the network enterprise. The analysis and study of network flow are carried out by building efficient IDS using GA and fuzzy logic in Hamamoto et al. (2018). GA is used to form the digital signature of the network IP segment using flow analysis. GA extracts the information from the data packet to predict the behavior of the network traffic for a given time series. Later, fuzzy logic is used to detect whether the given instance is anomalous or not. The proposed framework possess the capability of monitoring the traffic flowing through the network and generating alerts when any unusual behavior is detected. The experimental results showed the accuracy of 96.53% and false positive rate of 0.56% (Hamamoto et al. 2018).

The EA is also applied to wireless networks such as wireless mesh networks. These networks consist of mesh nodes that play a vital role in handling different types of attacks posed on the network. A simulated dataset for Wireless Mesh Network (WMN) is presented in Vijayanand et al. (2018). SVM is used for building the IDS model. The simulated dataset consists of a large number of redundant and irrelevant features that can deteriorate the performance of the IDS. Therefore, features selection is performed using GA to increase the accuracy of the system. A hybrid framework of GA-SVM gathers the informative features of each category of attack namely grey hole, black hole, data flooding, hello flooding, and jamming. The simulated dataset is developed using the Network Simulator 3 (Carneiro 2010) and considers packet delivery ratio, throughput, and end-to-end delay as network parameters. The proposed system exhibits high accuracy in detecting attacks for WMN (Vijayanand et al. 2018).

A genetic neuro fuzzy system is proposed in Kondaiah and Sathyanarayana (2018) for detecting intrusions in mobile ad-hoc networks. Here, in the proposed work, FA is hybridized with PSO to derive optimal secure routing paths in the network. During the initial phase of the proposed work, GA based neuro fuzzy system is used for differentiating between trust nodes and malicious and later, hybrid FA and PSO method used for establishing a secure routing path between the source and destination along the trust nodes. The experiments performed using Network Simulator 2 (Carneiro 2010), and results presented in terms of detection rate, throughput, and delay.

5 Datasets for intrusion detection systems

For measuring the performance of any IDS; there is a high requirement of a standard dataset that can validate the comparison of different classifiers. For instance, in the 1998, MIT Lincoln laboratory developed DARPA-98 dataset under DARPA funded projects. This dataset used over the last two decades for evaluating the performance of the IDS models. Based on the analysis, many drawbacks were identified such as presence of duplicated records, an imbalance in the records of training and test datasets, and consideration of synthetic traffic (McHugh 2000).

To overcome these limitations, research being carried out for creating more refined versions of the dataset such as NSL-KDD (Brown et al. 2009). In spite of so many efforts towards dataset generation, there has been research to find one partial alternative to the available dataset. One of the limitations of the datasets used for IDS is that they are deprived of the real traffic or updated knowledge of the novel attacks (Shiravi et al. 2012; Thakkar and Lohiya 2020a). The time to record the network traffic while the creation of the dataset also plays a significant role. Many detection algorithms consider cyclostationary evolution of traffic which means the difference in traffic between day time/night time or weekdays/weekends for evaluation, and therefore, a long trace is required (Shiravi et al. 2012). This paper lists a brief review of the datasets used for evaluating the IDS and the requirements that dataset should comply for IDS evaluation.

5.1 Datasets used for performance evaluation of IDS

This section discusses the datasets referred in the literature along with their characteristics and limitations.

5.1.1 DARPA

This dataset was the part of the DARPA[1] funded project of MIT Lincoln Laboratory and was developed to meet the requirements of network traffic analysis under the attack scenarios and normal traffic. It includes service categories such as SMTP, FTP, HTTP, and Telnet. It has four attack categories namely DoS, Probe, R2L, and U2R. This dataset has

[1] DARPA https://www.ll.mit.edu/r-d/datasets Last accessed: 24, August, 2020.

certain limitations such as it is deprived of the real network traffic, the data captured do not exhibit regularity, and absence of false positives. This dataset is not capable of evaluating the performance of IDS for identifying and classifying novel attacks (Brown et al. 2009; McHugh 2000).

5.1.2 KDD CUP 99

KDD CUP 99[2] dataset is refined version of DARPA and created by critically analyzing the tcpdump files of the network traffic. The dataset consists of both normal and anomalous network traffic with different attack types of DoS, Probe, R2L, and U2R. The limitation of this dataset is the presence of a huge number of redundant network traffic that leads to inconsistency in the results and produces skewed results because of the redundant data (Tavallaee et al. 2009). A more refined form of KDD CUP 99 dataset was derived by removing these redundant records. This dataset is known as NSL-KDD[3] (McHugh 2000) and overcomes the disadvantages of the KDD CUP 99 dataset.

5.1.3 DEFCON

DEFCON[4] dataset was formed by the Shmoo Group in the year 2000 by capturing the traffic produced during the "capture the flag" competition (Migliavacca et al. 2010). It consists of attacks like port scanning, buffer overflow, sweep, unauthorized access, telnet and FTP protocol attacks, and bad packets. This dataset has the limitation of being unreal from the actual network traffic. It can be utilize to implement many alert correlation methods (Migliavacca et al. 2010; Nehinbe 2009).

5.1.4 CAIDA (Center of Applied Internet Data Analysis)

The CAIDA[5] has built a dataset which is very specific to attack types and the payload information they provide regarding the source and destination of the network packet. It was first built in the year 2002, by using OC48 link in San Jose by analyzing data. Then CAIDA DDoS was formed, which consists of DDoS attack traffic captured from pcap files and CAIDA internet traces in the year 2016. This dataset consist of passive network traffic from CAIDA Equinix-Chicago monitor sniffed on a high-speed internet backbone. These datasets can not be used as standard datasets as they have a narrow scope of attack types. The drawbacks are studied in detail in Tavallaee et al. (2009) and Shiravi et al. (2012).

5.1.5 Lawrence Berkeley National Laboratory (LBNL) Dataset

LBNL[6] dataset was created by capturing the traces of network traffic with TCP, UDP, and ICMP protocol services. The captured packets lack the information regarding the packet

[2] KDD CUP 99
https://www.kdd.org/kdd-cup/view/kdd-cup-1999/Data Last accessed: 24, August, 2020.

[3] NSL-KDD https://www.unb.ca/cic/datasets/nsl.html Last accessed: 24, August, 2020.

[4] DEFCON https://www.defcon.org/html/links/dc-torrent.html Last accessed: 24, August, 2020.

[5] CAIDA https://www.caida.org/data/about/downloads/ Last accessed: 24, August, 2020.

[6] LBNL https://powerdata.lbl.gov/download.html Last accessed: 24, August, 2020.

payload and the dataset has a limitation of anonymization (Nechaev et al. 2004). It is applied for the internal and external applications such as web, email, name services, and network file services.

5.1.6 CDX

CDX[7] dataset was created by the United States Military Academy by capturing the traces of the warfare competitions which were converted into the labeled dataset for network analysis. It is used for applications like web, email, and DNS lookup. Intruders can perform malicious activities with the help of network tools such as Nessus, Nikto, and web scarab on the systems configured for collecting the network traffic. This dataset was utilized to validate alert rules for IDS but it has a very narrow scope in terms of magnitude and variance (Sangster et al. 2009).

5.1.7 Kyoto

Kyoto[8] dataset was developed by the Kyoto University in the year 2009 with the help of honeypots. As this was generated by capturing network traces using honeypot, only the attacks targeted at the honeypot were analyzed. The dataset consists of analyzed network traffic with a small amount of realistic user behaviour such as DNS and mail traffic. As a result of this, the dataset consists of negligible false positives which is essential for decreasing the alerts generated (Song et al. 2011; Sato et al. 2012; Chitrakar and Huang 2012).

5.1.8 Twente

Twente[9] dataset was developed in the University of Twente in the year 2009 by Sperotto. The data was captured by the honeypot network by Net-Flow using auth/ident. It includes network data which covers services like OpenSSH, Proftp, and Apache web server. The captured data is not completely intrusive or normal. It also contains continuous traffic of ICMP and Internet Relay Chat (IRC). Dataset generated is labeled with a minimum correlation between the alerts generated and has a narrow scope of diversity of the intrusions and volume (Sperotto et al. 2009).

5.1.9 UMASS

The University of Massachusetts developed the UMASS[10] dataset by sniffing some network traces from their wireless applications (Nehinbe 2011). The dataset is generated by considering intrusion scenario where only one TCP based connection is observed. It is not capable of detecting or preventing intrusions as it has a very limited variety of attacks and network data (Prusty et al. 2011).

[7] CDX http://www.fit.vutbr.cz/~ihomoliak/asnm/ASNM-CDX-2009.html Last accessed: 24, August, 2020.

[8] Kyoto https://www.takakura.com/Kyoto_data/ Last accessed: 24, August, 2020.

[9] Twente https://www.utwente.nl/en/eemcs/ps/research/dataset/ Last accessed: 24, August, 2020.

[10] UMASS http://traces.cs.umass.edu/index.php/Network/Network Last accessed: 24, August, 2020.

5.1.10 ISCX2012

ISCX2012[11] dataset is developed in two phases. In the first phase, multi-stage intrusion attacks were carried out and stored. This profile was referred to as Alpha profile. In the second phase, normal traffic was generated with background noise and this was referred to as Beta profile. It consists of network protocol services such as FTP, HTTP, SMTP, POP3, SSH, and IMAP with full network packet payload information. As a result, it does not correspond well with the current network scenarios as most of the network traffic is HTTPS (Shiravi et al. 2012).

5.1.11 ADFA

ADFA[12] dataset was developed by the University of New South Wales by storing ten attacks per vector using training and validation set (Creech and Hu 2013). The attacks can be listed as FTP and SSH password stealing using brute force, java interpreter, misusing administrative privilege by adding superuser, Linux meterpreter, and C100 web shell attacks. The attacks are not well distinguished by the benign network traffic. This dataset does not identify different attacks categories (Xie and Hu 2013; Xie et al. 2014).

5.1.12 UNSW-NB15

UNSW-NB15[13] dataset was developed using attack automatic generation tool named IXIA perfect storm in the Cyber Range lab of the Australian Centre for Cyber Security (Moustafa and Slay 2015). To sniff the raw network traffic tcpdump tool (Duarte and Farruca 2010) was used. Other tools like Argus and Bro-IDS (Mehra 2012) were used to develop realistic network scenarios and extracting features. It includes 9 attack categories with realistic network scenarios and diversified attack types. Moreover, the dataset consists of 49 network traffic features with a detailed analysis of packet payload and network traffic. The dataset was partitioned into two sets namely training and test sets for analyzing the data. This dataset is statistically complex as it contains homogenous patterns for attack and normal traffic (Heck et al. 2013).

5.2 CIC-IDS-2017 Dataset

Pertaining to the study of the earlier datasets as discussed, most of the datasets lack addressing the real world scenarios of new attack types. Moreover, these datasets do not reflect the current trends, variations in the network traffic and in-depth information about the packet payload, feature set, and metadata (Thakkar and Lohiya 2020a). To overcome the above drawbacks alongwith to meet the needs of modern-day attacks and network traffic, Canadian Institute for Cybersecurity (Sharafaldin et al. 2018) developed a dataset

[11] ISCX2012 https://www.unb.ca/cic/datasets/ids.html Last accessed: 24, August, 2020.

[12] ADFA https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-IDS-Datasets/ Last accessed: 24, August, 2020.

[13] UNSW-NB15 https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/ Last accessed: 24, August, 2020.

CIC-IDS-2017[14] which meets the needs of new updated attacks and realistic network scenarios based on certain evaluation criteria. The data were collected over the span of a week with different labels and attack categories with 80 features as listed in Table 10 (Sharafaldin et al. 2018).

5.2.1 Dataset analysis and feature description

The analysis of the dataset can be accomplished in four steps as shown in Fig. 10. In the first step network traffic features are extracted using the CICFlowMeter (Gharib et al. 2016). The pcap files captured during the network flow are studied to extract features. The extracted features are flow-based and consist of labels such as protocol name, source port number, destination port number, source IP address, and destination IP address. The dataset is collected for a week, and therefore, labeling is performed according to the day wise data collected. In total 80 features are extracted from the pcap files (Lashkari et al. 2017).

Further to obtain the most significant feature set for each attack category defined by the dataset. This is achieved by Random Forest Regressor algorithm which works by calculating the IG of each feature, and then computing the product of feature importance and mean value of each feature split for each class. The next step is defining the subset of features for each of the attack classified by the dataset. For instance, flow duration and inter-arrival time features like minimum, mean, and maximum are capable of describing DoS attack. The evaluation of the dataset is carried out with performance metrics like precision, recall, and F-score. Validation of the dataset is carried out by considering the evaluation criteria as discussed in Gharib et al. (2016). The criteria described in Table 11 (Gharib et al. 2016) have covered the requirements necessary for the intrusion detection dataset which are missing in the previous datasets. Similarly, the feature set for each attack is listed in Table 12 (Sharafaldin et al. 2018).

6 Performance measures

The model evaluation can be carried out in many ways based on the datasets being used (Basnet et al. 2008). Generally, the evaluation of the IDS can be described using the efficiency and effectiveness of the IDS model (Akinyelu and Adewumi 2014). These measures consider the resources used along with the memory storage and computational time. These measures showcase how effectively the model is able to classify the data. The classification tasks can be categorized as binary classification, multi-class classification, and multi-label classification (Blum et al. 2010). The binary classification task classifies the given data into one of the two classes.

Binary classification evaluation is subjective and ambiguous (Kelleher et al. 2015). However, the identified classes are well-defined (Blum et al. 2010). Multi-class classification as the name suggests has *n* number of classes in which the data needs to be classified. This category of classification can be objective or subjective and well-defined or ambiguous, respectively (Kelleher et al. 2015). Multi-label classification deals with classifying the data into the target labels present in the dataset, it is similar to perform tagging (Obermeyer and Emanuel 2016). Generally, the genuineness of the classification model can

[14] CIC-IDS-2017 https://www.unb.ca/cic/datasets/ids-2017.html Last accessed: 24, August, 2020.

Table 10 Labelling of the CIC-IDS Dataset (Sharafaldin et al. 2018)

Type	Duration	Label	Profile
Normal Traffic	Monday	Benign	It is the normal network traffic
Brute Force	Tuesday (Morning – Afternoon)	BForce, SFTP and SSH	It is the attack performed for stealing sensitive information from the system or web application like passwords
DoS Attack	Wednesday (Morning –Afternoon)	DoS and Hearbleed Attacks slowloris, Slowhttptest, Hulk and GoldenEye	Scanning the system for vulnerabilities and exploiting them to misuse the memory storage and system resources
Web Attack	Thursday (Morning)	Web BForce, XSS and SQL Inject	It is the kind of attack performed to exploit the web application
Infiltra-tion Attack	Thursday (Afternoon)	Infiltration Dropbox Download and Cool disk	It is injecting malicious files into the system by attaching it through mail or transferring it through flash memory to execute portscan
Botnet Attack	Friday (Morning)	DDoS LOIT, Botnet ARES	It is a software-based attack which can create a loophole into the system so that the attacker can have remote access to upload/ download or misuse any system information
DDoS Attack and Port Scan	Friday (Afternoon)	PortScans (sS,sT,sF ,sX,sN,sP, sV, sU,sO, sA, sW, sR, sL and B)	It is way by which attacker scans the victim system for any open ports or backdoor to implement any attack or get remote access

Fig. 10 Framework of CIC-IDS 2017 Dataset

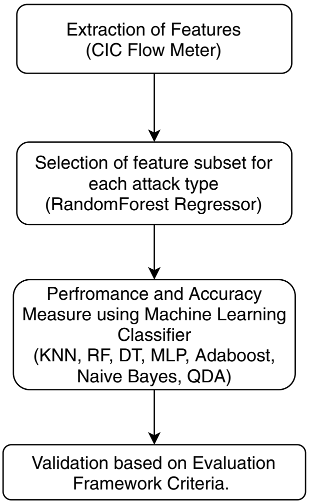
Extraction of Features
(CIC Flow Meter)
Selection of feature subset for each attack type
(RandomForest Regressor)
Perfromance and Accuracy Measure using Machine Learning Classifier
(KNN, RF, DT, MLP, Adaboost, Naive Bayes, QDA)
Validation based on Evaluation Framework Criteria.

Table 11 Evaluation criteria for Dataset CIC-IDS 2017 (Gharib et al. 2016)

Sr No	Evaluation criteria	Description
1	Complete Network Configuration	It includes having detail information about the network topology adopted with the operating system information
2	Network Traffic	It includes complete network flow details between the configured attack system and the victim system
3	Labeled Dataset	The dataset includes the details about the time duration which was considered for capturing the network traffic with all the necessary labels required for describing the dataset

Table 11 (continued)

Sr No	Evaluation criteria	Description
4	Complete Interaction	It defined as having complete communication between the internal and external network
5	Complete Capture	It is ensuring that all the network traffic is stored at the server using network capturing tools
6	Available Protocols	The dataset should be well equipped with all the protocol services of current network communication
7	Attack Diversity	It should have a wide variety of attacks which are listed and known as well as attacks that can happen by exploiting the available system vulnerabilities
8	Heterogeneity	The data collected should not have similar characteristics and it should exhibit variance
9	Feature Set	The main task in building the IDS dataset is feature set. To extract the necessary features from the pcap files obtained by capturing the traffic from the tools
10	MetaData	In detail analysis and explanation of the dataset and features

Table 12 Features for each Attack Type Sharafaldin et al. (2018)

Label	Feature	Description
Benign	B.Packet LenMin	Minimum Length of the backward network packet
	SubflowF.Bytes	Length of the flow of forward network packets in bytes
	Total Len F.Packets	Total length of the forward network packets
	F.Packet Len Mean	Mean of the length of the forward packet
DoSGoldenEye	B.Packet Len Std	Standard deviation of the backward packet length
	Flow IAT Min	Minimum of the flow inter arrival time features
	Fwd IAT Min	Minimum of forward inter arrival time features
	Flow IAT Mean	Mean of flow inter arrival time features
Heartbleed	B.Packet Len Std	Standard deviation of length of backward packet
	SubflowF.Bytes	Length of the flow of forward network packets in bytes
	Flow Duration	Describes the time period of the flow
	Total Len F.Packets	Total length of the forward network packets
DoS Hulk	B.Packet Len Std	Standard deviation of the backward packet length
	B.Packet Len Std	Standard deviation of the backward packet length
	Flow Duration	Describes the time period of the flow
	Flow IAT Std	Standard deviation of forward inter arrival time features
DoSSlowhttp	Flow Duration	Describes the time period of the flow
	Active Min	Active minimum value of the features
	Active Mean	Active mean value of the features
	Flow IAT Std	Standard deviation of forward inter arrival time features
DoSslowloris	Flow Duration	Describes the time period of the flow
	F.IAT Min	Minimum of forward inter arrival time features
	B.IAT Mean	Mean of backward inter arrival time features
	F.IAT Mean	Mean of forward inter arrival time features

Table 12 (continued)

Label	Feature	Description
SSH-Patator	Init Win F.Bytes	Initialization of the window size for the forwarding bytes
	SubflowF.Bytes	Length of the flow of forward network packets in bytes
	Total Len F.Packets	Total length of the forwarding packet
	ACK Flag Count	Count of the acknowledgement flag
FTP-Patator	Init Win F.Bytes	Initialization of the window size for the forwarding bytes
	F.PSH Flags	Value of the forwarding push flags
	SYN Flag Count	Count of the synchronization flag
	F.Packets/s	Forward packets per second
Web Attack	Init Win F.Bytes	Initialization of the window size for the forwarding bytes
	SubflowF.Bytes	Length of the flow of forward network packets in bytes
	Init Win B.Bytes	Initialization of the window size for the backwarding bytes
	Total Len F.Packets	Total length of the forwarding packet
Infiltration	SubflowF.Bytes	Length of the flow of forward network packets in bytes
	Total Len F.Packets	Total length of the forwarding packet
	Flow Duration	Describes the time period of the flow
	Active Mean	Active mean value of the features
Bot	SubflowF.Bytes	Length of the flow of forward network packets in bytes
	Total Len F.Packets	Total length of the forwarding packet
	F.Packet	Forwarding network packet

Table 13 Confusion matrix

Actual	Predicted normal	Predicted attack
Normal	*TP* (true positive)	*FN* (false negative)
Attack	*FP* (false positive)	*TN* (true negative)

be presented by evaluation metrics which is defined by constructing the confusion matrix. Table 13 represents the confusion matrix for IDS. The confusion matrix constitutes the count of the correctly classified instances for normal and attack class known as true positive and true negative, respectively. Instances which are incorrectly classified for normal and attack class are known as false positive and false negative, respectively.

6.1 Performance metrics for classification models

To measure the representative power of any IDS and to compare the experimental results of different techniques, performance metrics are used and are derived from the confusion matrix. Most of the metrics are primarily defined for the binary classification problems. Therefore, to measure the performance of multi-class, and multi-label classification problems the weighted average or sampling of instances of the dataset is considered (Kelleher et al. 2015). The weighted average can be calculated by considering the weighted score of each instance of the class by its presence in the dataset. Some of the metrics used for evaluating classification models are as follows.

1. Accuracy: Accuracy can be defined as the classification rate of the model which is given by the proportion of correctly classified instances $(TP + TN)$ to the total number of instances in the dataset $(TP + TN + FP + FN)$. Accuracy can be computed using Eq. 1 (Kelleher et al. 2015).

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

2. Precision: It is the measure of covariance of any model which is given by the ratio of correctly identified instances (TP) to the sum of instances which are classified as correct $(TP + FP)$. It includes repeatability and reproducibility of the resources. Precision can be evaluated using Eq. 2 (Kelleher et al. 2015).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

3. False Positive Rate: It is an error metric which is defined as the ratio of number of instances which are misclassified (FP) to the sum of false positive and true negative $(FP + TN)$ as shown in Eq. 3 (Kelleher et al. 2015).

$$\text{False Positive Rate} = \frac{FP}{FP + TN} \tag{3}$$

4. False Negative Rate: It is an error metric which is defined as the ratio of number of instances which are misclassified (FN) to the sum of false negative and true positive $(FN + TP)$ as shown in Eq. 4 (Kelleher et al. 2015).

$$\text{False Negative Rate} = \frac{FN}{FN + TP} \tag{4}$$

5. True Positive Rate: This is referred to as sensitivity and is defined as the measure, evaluated by calculating the ratio of correctly classified (TP) to sum of true positives ($TP + FN$) as shown in Eq. 5 (Kelleher et al. 2015).

$$\text{True Positive Rate} = \frac{TP}{TP + FN} \tag{5}$$

6. True Negative Rate: This is referred to as specificity and is defined as the measure, evaluated by calculating the ratio of correctly classified samples (TN) to sum of true negative and false positive ($TN + FP$) as shown in Eq. 6 (Kelleher et al. 2015).

$$\text{True Negative Rate} = \frac{TN}{TN + FP} \tag{6}$$

7. Balanced Accuracy: This is called as the area under the curve which is a regarded as a summary statistic. ROC is the trade off between the true positive rate and the false positive rate of the classification model which is build. It is equal to the arithmetic mean of true positive rate and true negative rate as presented in Eq. 7 (Kelleher et al. 2015).

$$\text{Balanced Accuracy} = \frac{1}{\sum \hat{w}_i} \sum 1(\hat{y}_i = y_i)\hat{w}_i \tag{7}$$

where y_i is actual value of sample i, $\hat{y}_i$ is the predicted value of sample i, and $\hat{w}_i$ is the corresponding weight of sample i.

8. Cohen Kappa: It is a statistical measure used for comparing categorical instances. It is used for binary and multiclass classification problems. Its value ranges from −1 to 1. If the score is more than 0.8 than it is considered to be good classification (McHugh 2012).
9. Hamming Loss: It is generally used with the multi-class classification. It refers to the hamming distance between the true value and the predicted value. Its value is between 0 and 1. Hamming loss is computed using Eq. 8 (Dembczynski et al. 2013).

$$L_{Hamming} = \frac{1}{n_{\text{labels}}} \sum_{j=0}^{n_{\text{labels}}-1} 1(\hat{y}_j \neq y_j) \tag{8}$$

where y_j, $\hat{y}_j$, and $L_{Hamming}$ are the actual value of sample j, predicted value of sample j and Hamming loss between two samples for a given dataset respectively, and n_{labels} is the number of labels.

10. Jaccard Similarity Coefficient Score: It is a statistical measure used for comparing the similarity and diversity of the instances in the given dataset. It is obtained by the Jaccard distance measured by the ratio of the difference of sizes of union and intersection of the sets consisting of the sample by the size of the union of the sets. The Jaccard similarity coefficient score is derived using the Eq. 9 (Choi et al. 2010).

$$J(y_i, \hat{y}_i) = \frac{|y_i \cap \hat{y}_i|}{|y_i \cup \hat{y}_i|}. \tag{9}$$

where y_i, $\hat{y}_i$ denote the actual value of sample i and predicted value of sample i, respectively. $J(y_i, \hat{y}_i)$ is the Jaccard Similarity Coefficient between two samples for a given dataset.

11. F-score: It is defined as the weighted harmonic mean of precision and recall. It is computed using Eq. 10 (Kelleher et al. 2015).

$$F - score = 2 * \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{10}$$

12. Hinge Loss: It is the distance measure between the classification model and the dataset being used. It takes only predicted errors into consideration as shown in Eq. 11. This is used with classifiers which have maximum margin like SVMs (Dembczynski et al. 2013).

$$L_{\text{Hinge}}(y_w, y_t) = \max\left\{1 + y_t - y_w, 0\right\} \tag{11}$$

Here, y_w is the predicted decision for the true label, y_t is the maximum of the predicted decisions for all other labels, and L_{Hinge} is the Hinge Loss.

13. Logarithmic Loss: It is referred to as the cross entropy loss based on the probability estimation given by Eq. 12 (Dembczynski et al. 2013), where y is the true labels of the samples and p is the estimated probability of the samples. This performance metric is mostly considered with logistic regression, expectation maximization, and neural network models (Dembczynski et al. 2013). It can be applied to both binary and multi-class classification problem.

$$L_{\log}(y, p) = -\log \Pr(y|p) = -(y \log(p) + (1 - y) \log(1 - p)) \tag{12}$$

14. Matthews Correlation Coefficient (MCC): It measures the quality of the classes identified. As shown in Eq. 13, TP, FP, TN, and FN are taken into consideration for the evaluation of MCC. This metric can be used with the classes of variable sizes and its value ranges from −1 and +1. If the value of the coefficient is +1 then it denotes perfect classification, 0 denotes average random classification and −1 denotes inverse classification. It is also known as the phi coefficient. This can be used with binary and multi-class classification problem (Kelleher et al. 2015).

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{13}$$

15. Explained Variance Score: It is a measure that statistically measures the portion of the learning model responsible for the variation of the data provided to the model. The expected value is estimated to be 1. It is derived using the Eq. 14 (Tsoumakas et al. 2010).

$$\text{Explained Variance Score}(y, \hat{y}) = 1 - \frac{Var\{y - \hat{y}\}}{Var\{y\}} \tag{14}$$

Here, $\hat{y}$ is the estimated target output, y is corresponding correct target output, and Var is Variance.

16. Mean Absolute Error (MAE): It is the difference between two continuous variables of a given dataset. It measures the risk factor with respect to the target value of the absolute error loss (Tsoumakas et al. 2010). It is computed using the Eq. 15.

$$\text{MAE} = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} |y_i - \hat{y}_i| \tag{15}$$

for the given n_{samples}, $\hat{y}_i$ is the predicted value of the *ith* sample and y_i is the true value of the *ith* sample.

17. Mean Squared Error (MSE): It is a risk function that measures the average of the squared errors given by Eq. 16. The value of MSE is always positive and it measures the quality of the learning model (Tsoumakas et al. 2010).

$$\text{MSE} = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (y_i - \hat{y}_i)^2 \tag{16}$$

for given n_{samples}, $\hat{y}_i$ is the predicted value of the *ith* sample and y_i is the true value of the *ith* sample.

18. R^2 Score (coefficient of determination): It represents the proportion of variance explained by the independent variables in the model. It provides an estimation of how well unknown samples are likely to be predicted by the model, through the proportion of explained variance. The R^2 score is computed using the Eq. 17 (Tsoumakas et al. 2010).

$$R^2 = 1 - \frac{\sum_{i=0}^{n_{\text{samples}}-1} (y_i - \hat{y}_i)^2}{\sum_{i=0}^{n_{\text{samples}}-1} (y_i - \bar{y})^2} \tag{17}$$

For a given n_{samples}, $\hat{y}_i$ is the predicted value of the *ith* sample and y_i is the true value of the *ith* sample. The value of $\bar{y}$ is given by 18.

$$\bar{y} = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} y_i \tag{18}$$

6.2 Performance metrics for clustering models

Performance metrics used for clustering problems measure the quality of the clusters based on the similarity between the data samples. Some of the performance metrics considered for clustering problems are as follows.

1. Adjusted Rand Index (ARI): It is an enhanced version of Rand Index given by Eq. 19 which is similar to calculating the accuracy of the model even when the classes are not defined (Yang et al. 2016). It uses permutation to measure the similarity between the data clusters formed by the learning model. Its value lies between 0 and 1. The adjusted rand index is computed using Eq. 20 (Yang et al. 2016).

$$\text{RI} = \frac{a+b}{C_2^{n_{samples}}} \tag{19}$$

$$\mathrm{ARI} = \frac{\mathrm{RI} - E[\mathrm{RI}]}{\max(\mathrm{RI}) - E[\mathrm{RI}]} \tag{20}$$

With C as the class a and b can be given as the number of pairs of elements that are in the same set in C and different sets in C, respectively. E[RI] is the expected value of RI, and $C_2^{n_{samples}}$ is the total number of possible pairs.

2. Mutual Information Based Scores: Mutual Information (MI) can be defined as the metric that measures the dependence of the attributes of the class based on the amount of information obtained from them. It has two forms of Adjusted Mutual Information (AMI) shown in Eq. 21 and Normalized Mutual Information (NMI) shown in Eq. 22 (Yang et al. 2016).

$$\mathrm{AMI} = \frac{\mathrm{MI} - E[\mathrm{MI}]}{\mathrm{mean}(H(U), H(V)) - E[\mathrm{MI}]} \tag{21}$$

$$\mathrm{NMI}(U, V) = \frac{\mathrm{MI}(U, V)}{\mathrm{mean}(H(U), H(V))} \tag{22}$$

Here, U and V are the two class assignments and $H(U)$ and $H(V)$ are the entropy measure of the respective classes. E[MI] is the expected value of MI.

3. Entropy Analysis: It is concerned with calculating the homogeneity and completeness of the learning model given by Eqs. 23 and 24, respectively. Homogeneity refers to the fact that a cluster consists of data samples having similar characteristics and completeness refers that data instance of the same class belong to the same cluster. V-measure is given by Eq. 25 that represents the harmonic mean of homogeneity and completeness (Vinh et al. 2010).

$$h = 1 - \frac{H(C|K)}{H(C)} \tag{23}$$

$$c = 1 - \frac{H(K|C)}{H(K)} \tag{24}$$

$$v = 2 \cdot \frac{h \cdot c}{h + c} \tag{25}$$

Here, h, c, and v denote the homogeneity, completeness, and V-measure of a given class, respectively. $H(C|K)$is the conditional entropy of the class given the cluster assignments and $H(C)$ is the entropy of the class.

4. Fowlkes Mallows scores (FMI): FMI score is used with hierarchical clustering, which can be represented as the geometric mean of precision and recall (Fowlkes and Mallows 1983). It is given by Eq. 26.

$$\mathrm{FMI} = \frac{\mathrm{TP}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})}} \tag{26}$$

5. Silhouette Coefficient: It is a method to evaluate the quality of the identified clusters when labels are not known. It is a combination of two scores namely the mean distance between the data point with other data points in the same cluster and mean distance

between the given data point and all the other point in the neighbouring cluster (Rousseeuw 1987). The silhouette coefficient is computed using the Eq. 27.

$$s = \frac{b - a}{max(a, b)} \tag{27}$$

where a is the mean distance between a sample and all other points in the same class and b is the mean distance between a sample and all other points in the neighbouring cluster.

6. Calinski-Harabaz Index: It is a measure for evaluating the quality of the model which has well-defined clusters. It is also known as the variance ratio criteria for a given model with k-clusters (Halkidi et al. 2001).
7. Davies Bouldin Index: It gives the average similarity between the clusters. It signifies how well the clusters are separated from each other. It is calculated by the average distance between each data sample in the given cluster and the cluster center and distance between all the cluster centers which are defined. It is given by Eq. 28 (Halkidi et al. 2001).

$$s(k) = \frac{B_k}{W_k} \times \frac{N - k}{k - 1} \tag{28}$$

Here, N is the number of points in the dataset, k is the clusters, B_k is the between the group dispersion matrix and W_k is within cluster dispersion matrix.

6.3 Performance metrics for swarm and evolutionary algorithms

Assuming the number of generations of SWEVO algorithm for a given problem to be optimized the following performance measures can be defined in context with the quality of the solution obtained irrespective of the convergence.

1. Likelihood of Optimality: To obtain an optimal solution for a given problem with p generations, n number of runs are needed then the likelihood of optimality $L_{opt}(p)$ at pth generation can be given by the estimated probability as $\frac{n}{q}$ (Riquelme et al. 2015).
2. Average Fitness Value: For every algorithm executed for p generations in every q runs, the average fitness value $f(p)$ at the pth generation is given by the average of the best fitness values derived within p generations in q runs (Riquelme et al. 2015).
3. Likelihood of Evolution Leap: For a given generation if the optimal solution derived is better than the obtained best solution before the p generation then that particular generation is called evolution leap (Riquelme et al. 2015). For a given algorithm executed with p generations and q runs, suppose n is the average number of leaps found within the q runs then the likelihood of the evolution leap can be given by the estimated probability as $\frac{n}{q}$ (Riquelme et al. 2015).
4. Convergence Metrics: For a multi-objective SWEVO algorithm the performance can be measured with two metrics Convergence and Diversity Metrics. Convergence metrics can be defined as the evaluation from a set of optimal solutions to obtain the best solution for a given population. And Diversity metrics are defined to determine the scatter of the solutions from the available set of solutions for a given population (Riquelme et al. 2015).

6.4 Statistical tests and cross validation

For comparing the performance of predictive models for a given dataset statistical tests can be used and to determine the generalized performance of the model cross validation can be applied. In cross-validation, data is partitioned to certify that the performance is measured using independent dataset, whereas statistical tests do not partition the data. Both of these methods can be used for selecting the model and performing the in-depth analysis of the results obtained. For instance, data samples for performance evaluation can be obtained through cross-validation and statistical tests can be used to compare the significance of the models implemented.

A hybrid intrusion detection method is proposed using the random forest and k-means clustering on ISCX dataset in Soheily-Khah et al. (2018). Here, the dataset is partitioned based on the application layer services involved such as HTTP, ICMP, SSH, FTP to name a few. The data is clustered based on the feature vectors and the validation of the data clusters formed is measured using the Silhouette coefficient, which implies that clusters are more isolated if the value of the coefficient is high. The training set and test set is derived by using cross-validation with $k = 5$ (Lever et al. 2016). The proposed hybrid method is compared with other classifiers such as NB, SVM, and DT using accuracy, detection rate, and false alarm rate. To prove the significance of the proposed method Wilcoxon signed rank test is used, which is the statistical hypothesis test. The statistical hypothesis test effectively determines that the proposed method is better than the other methods (Soheily-Khah et al. 2018). In this way, statistical tests and cross-validation can be used for model selection and evaluation.

7 Applications

IDSs have been a subject of study in varied domains because of their characteristics such as modality, scalability, and perspective to easily adapt to naive challenges. IDS is an area of research which is growing at a very fast pace and needs to be scrutinized because of its wide applicability in domains such as web and cloud technologies, banking sector, social networking, to name a few (Srinivas et al. 2019; Rajput and Thakkar 2019). There are many learning techniques which have been employed to solve problems of varied applications. This section highlights some of the most common applications where IDS can be applied to detect any intrusion or attack. The section is composed of applications like phishing (Akinyelu and Adewumi 2014), SMS spam detection (Wang 2010), cyber attacks, healthcare security (Blum et al. 2010), and IoT system security (Dasgupta 2012; Lohiya and Thakkar 2020).

7.1 Phishing

Phishing is a type of security breach in which the attacker tries to gain sensitive information, for instance, user credentials or user personal information through email or other communication channels. Probably a user would be conned by sending a message through an email. The message would contain malicious code which when clicked by the user would direct the user to malicious links to gain their personal information such as passwords or credit card credentials. Phishing is easy to implement because the links enclosed in the messages appear from a legitimate source, as the organization content and logo is spoofed.

An IDS detects phishing attack by studying various features of the web page and web link utilized for executing the attack. Generally, features extracted by studying the web page or web link can be classified based on URL, domain, content, and web page (Islam and Abawajy 2013). However, all the features may not contribute in improving the overall accuracy of IDS for phishing detection. For instance, content-based features cannot develop a fast detection mechanism, instead, they work well with domain name identification whereas, web page based features do not contribute to analyze the domains registered by the website. Hence, the detection process highly depends on the purpose and type of features to be used.

7.2 Digital forensics and evidence collection

Digital Forensics is a part of forensic science to investigate the digital findings and recovers information from digital devices. It is sub-branch of forensic science that focuses on evidence collection for legal investigation and proceedings (Casey 2011). The information obtained from digital devices such as computer systems and networks holds significance in digital forensics. The growing demand for storing and retrieving information from digital devices leads to an increase in the requirement of advanced technology for securing and maintaining the information (Xie et al. 2016). Therefore, an IDS can be used to secure the digital evidence as it requires the administrator to gather the information about the loopholes in the system.

The evidence obtained using IDS are recorded at the time of attack or when the vulnerabilities are exploited to compromise the system. The information collected can be about an open network connection, processes running in the system, files and system call (Schneier and Kelsey 1999). Thus, the information collected by the IDS can be directly presented as the evidence for legal procedures against any attack performed to harm the system and the digital devices (Sommer 1999).

7.3 SMS spam detection

Short Message Service (SMS) is a mode of communication in which a message is sent electronically. Reduced cost of the SMS service has given rise to spam attacks (Chen et al. 2015). Basically, a spam message can be termed as an unwanted message sent to the user's device. It can be in the form of advertisements, promotions, or marketing services. An intruder can send the malicious link in the text message to steal sensitive information of the user. There are many detection mechanisms which are implemented for detecting spam such as biometric identification, QR code, ML-based IDS, knowledge-based and authentication based solutions (Wang 2010). The most common spam messages are payment protection insurance, debt forgiveness, pension review, and quick loans (Kolari et al. 2006).

Feature selection plays an integral role for an IDS to filter spam messages. Firstly, the features are correlated to different message type for improving the accuracy of spam detection. Messages can be of two types of legitimate and spam message. Identification of a good feature is a challenging task for efficient spam detection. Some of the features can be listed as mathematical symbols present in the message, URL present in the message, special symbols, emotional symbols, lowercase letters, uppercase letters, numerals, keywords, and message length (Wang 2010). The detection of spam SMS is a binary classification problem where various features are used to train the classifier. Various publicly available datasets such as SMS Corpus can be explored to conduct spam detection experiments (Wang 2010).

7.4 Cyber attacks and software breaches

Smart technical devices and user systems exhibit the tendency to be affected by the loopholes in the code and software. Intruders can exploit the vulnerabilities of the system to cause potential harm. As an upshot, it affects not only an individual but also the entire nation or region. As stated in Dasgupta (2012), there are intelligent computer viruses that are capable of modifying drone code, changing their behavior, and can penetrate targets.

There is a high demand for developing IDS, which can tackle and search the possibilities to be implemented to repair these breaches and vulnerabilities as well as protect, against the novel attacks. For instance, many research projects like DARPA, KDD CUP 99 have been carried out to build datasets so that the system could be trained with potential attacks (Dasgupta 2012). Many research have been carried out to handle cyber attacks and software failures (Srinivas et al. 2019). For instance, a startup named ForAllSecure from Pittsburgh has launched a security bot in 2016 DARPA Cyber Grand Challenge (Song and Alves-Foss 2016).

An Automatic Exploit Generation is created by the CMU team to handle end-to-end system vulnerabilities exploitation. It has the capability of recognizing whether a bug is exploitable or not; if the bug is exploitable then it will generate a working flow to secure vulnerabilities. This can be illustrated by automated signature generation algorithm that takes a set of strings describing exploits as input and can recognize the exploits and its types.

On the other side, predictive analysis of Computer Science and Artificial Intelligence Laboratory (CSAIL) of MIT and PatternEx a machine learning startup designed a platform named AI^2 (Veeramachaneni et al. 2016). It can detect attacks significantly with the help of continuous input from human experts. It is built on the feedback given by the analysts. The system can be termed as Active Contextual Modeling that is capable of predicting and learning in real-time. PatternEx researchers have developed purely machine learning based solution and found that their work algorithmically increases detection rate by the factor of 10 as compared to other machine learning solutions (Dasgupta 2012).

7.5 Securing and preventing crime

The police departments in most of the countries are utilizing computer statistics for predictive policing. The predictive policing is based on artificial intelligence and is a systematic way of handling organization. Machine learning and predictive analytics have pioneered crime analysis tools such as IDS (Sharbaf 2018). For instance, Avata Intelligence based in California has created a software named Armorway which has diversified its applicability to healthcare and other areas are using machine learning with game theory to predict the activities of terrorist and other attacks (Sharbaf 2018). It has the capability of using data sources which contains information regarding passenger load numbers to traffic changes and based on this it creates a schedule that makes it impossible for the terrorist to depict the presence of police.

7.6 IoT systems security

Predicting and preventing malicious activities is the major goal of IDS. It is challenging to create an automatic and smart network where we can analyze and predict the event before its occurrence. To achieve the same, AT&T is working with how to utilize and increase the predictive services within their data centers (Chakraborty et al. 2018). For instance, telecommunication companies have implemented machine learning based solutions which takes input from contacts, chat, and voice operations to analyze the data to perform predictive real-time analysis.

Based on the analysis, officials can monitor to depict any anomalous behavior along with preparing a questionnaire to know satisfaction level of their customers. Sentiment analysis of their customers is performed by using machine learning and updating their services by taking necessary actions. Predictive analysis can turn out to be beneficial for telecommunication industries in the near future. It can also help in maintaining their feet (Shah and Issac 2018).

Another industry where IDS is playing a major role is the Internet of Things (Dasgupta 2012) where it utilizes cost-effective solutions for maintaining a number of complex assets. For instance, IBM's Watson use IoT and ML across the network which is capable of predicting real-time failure based on any asset condition. Thus, ML seems poised to become industry standard over the next decade (Li et al. 2019).

7.7 Commercial applications of IDS

Apart from the discussed application areas of IDS, there are various commercial applications that use IDS for securing the network. The Cisco Computer Security Incident Response Team uses lancope stealthwatch examining, monitoring and countering the network threats. The network solution developed by Cisco renders detail insights about network communication and aggregates the analyzed data for examining (Bollinger et al. 2015). The developed system is capable of detecting anomalous events connected with the applications, system resources, vulnerabilities, and DDoS attacks. It also tries to reduce false positives triggered by the system. To meet the demand of increasing breach detection and security automation, Cisco has developed a next-generation Intrusion Prevention System that is capable of providing signature-based and signature-less security with immediate visibility of attack targets and system vulnerabilities (Woland et al. 2018).

Plixer developed a flow-based intrusion response system named as Scrutinizer (Umer et al. 2017). The tool developed was able to analyze the behavioural characteristics of the network. The developed system augmented information regarding intrusion detection and other network-based communication. Due to the behavioural analysis of the network activities, it is capable of performing real-time intrusion detection. It also had the predefined algorithms to identify attacks such as flooding attack, port scan, and other malicious attacks.

A network security tool named as Flowmon Anomaly Detection System is a platform to control the modern day attacks (Kamisiński and Fung 2015). The tool performs rigorous network analysis using ML techniques. It examines the data flowing through the network in search of anomalies and unveils unusual patterns found in the network. It also provides a comprehensive solution to the not known or particular category of attacks whose signature or pattern is not available in the database.

The IBM QRadar Security Intelligence Platform is an integrated service that manages the logs, network monitoring activities, vulnerability, and risk management, network forensic investigation, intrusion response system, and advanced threat detection (Gupta et al. 2016b). It is a collaborative security approach to analyze the data and deliver better solutions in terms of anomaly detection.

Juniper Networks is a networking platform that has developed a complete set of networking tools for monitoring and analyzing the network (Sheth et al. 2015). One of their products named as Network Behaviour Anomaly Detection is used for identifying the malicious servers and applications in the system. Another tool for intrusion detection is the Bro IDS that is a comprehensive open source platform for network traffic analysis and anomaly detection (Udd et al. 2016).

8 Challenges and future research directions

Though there has been good research in the field of IDS, the success of these methods is highly dependent on the realistic environment. Unlike in other fields, ML algorithms have outperformed and have shown that manual inspection of data rendering makes it infeasible if the data quantities are high. It can be concluded that this discrepancy arises due to the fact that IDS unveil certain properties which makes deployment of ML algorithms difficult in many contexts. In this section, we discuss some of the challenges and future research directions to derive key issues posed by IDS in the operating environment. The schematic of challenges and future research directions is shown in Fig. 11.

- Evaluation Strategy: Research in the field of IDS have been carried out on public datasets for evaluation, however, there is no common test to show that the generated dataset consists of instances of real network traffic. Also, no validation test has been carried out to ensure that the attacks detected are the reflection of real network attacks.
- Dataset Upgradation: The DARPA and KDD CUP 99 benchmark datasets classify only four attack types for any IDS. The datasets created by the Lincoln Laboratory are outworn as they cannot exhibit the wide scope of the network attacks which has changed significantly over the last few years.

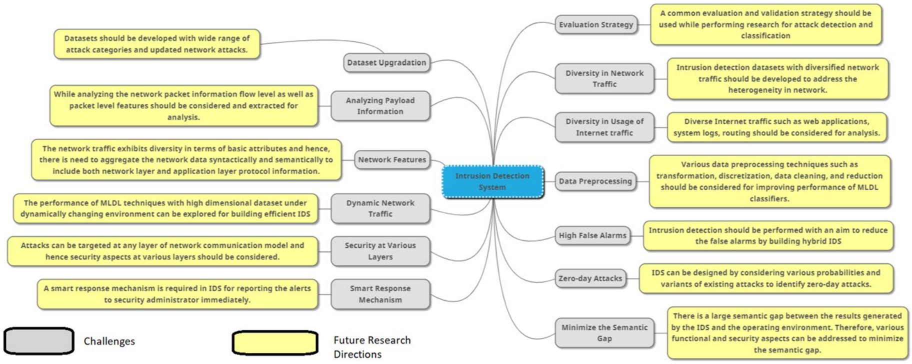
Datasets should be developed with wide range of attack categories and updated network attacks.
Dataset Upgradation
While analyzing the network packet information flow level as well as packet level features should be considered and extracted for analysis.
Analyzing Payload Information
The network traffic exhibits diversity in terms of basic attributes and hence, there is need to aggregate the network data syntactically and semantically to include both network layer and application layer protocol information.
Network Features
The performance of MLDL techniques with high dimensional dataset under dynamically changing environment can be explored for building efficient IDS
Dynamic Network Traffic
Attacks can be targeted at any layer of network communication model and hence security aspects at various layers should be considered.
Security at Various Layers
A smart response mechanism is required in IDS for reporting the alerts to security administrator immediately.
Smart Response Mechanism
Intrusion Detection System
Evaluation Strategy
A common evaluation and validation strategy should be used while performing research for attack detection and classification
Diversity in Network Traffic
Intrusion detection datasets with diversified network traffic should be developed to address the heterogeneity in network.
Diversity in Usage of Internet traffic
Diverse Internet traffic such as web applications, system logs, routing should be considered for analysis.
Data Preprocessing
Various data preprocessing techniques such as transformation, discretization, data cleaning, and reduction should be considered for improving performance of MLDL classifiers.
High False Alarms
Intrusion detection should be performed with an aim to reduce the false alarms by building hybrid IDS
Zero-day Attacks
IDS can be designed by considering various probabilities and variants of existing attacks to identify zero-day attacks.
Minimize the Semantic Gap
There is a large semantic gap between the results generated by the IDS and the operating environment. Therefore, various functional and security aspects can be addressed to minimize the semantic gap.
Challenges
Future Research Directions

Fig. 11 Schematic of challenges and future research directions

- Diversity in Network Traffic: Analyzing the audit log files of web servers is a good source of detecting attacks but they limit in giving essential information as HTTP request packets contain very less information.
- Analyzing Payload Information: Payload information from the packets, which are generated through real-time simulation is a good source of information but they raise privacy issues and hence, their usage is limited.
- Diversity in Usage of Internet Traffic: Packet header features are very useful for monitoring internal networks or performing behavioral analysis of the network but they are not sufficient for performing web-based anomaly detection (Umer et al. 2017).
- Network Features: The dataset developed by analyzing the captured network packets consists of different features that are used for deriving attack patterns and attack signatures. The features of the given dataset are not capable enough of identifying novel attacks.
- Data Preprocessing: Many data processing techniques such as data transformation, discretization, data cleaning, and reduction are of limited usage (Dewa and Maglaras 2016), which may increase the efficiency and accuracy of the system and detection models.
- Dynamic Network Traffic: The performance of computational intelligence techniques with high dimensional dataset under dynamically changing environment is yet to be explored. Moreover, ML methods are widely used on the public dataset and have shown significant results for detection rate and accuracy, however comparison of performance with the private datasets is to be explored.
- High False Alarms: Even though most of the techniques have achieved high detection rates, the techniques also exhibit high false alarm rate. Preventive measures should be taken to control false alarms. Moreover, a high false positive rate results in high cost as major time is spent to analyze the reported activity that eventually turns out to be normal network traffic.
- Zero-day Attacks: ML algorithms consider intrusion detection as a classification problem. These algorithms learn from the data based on the similarities rather than finding the outliers in the data (Hodge and Austin 2004). This suggests that designing an efficient approach for identifying zero day attacks is yet to be explored.
- Minimize the Semantic Gap: IDS faces a key challenge of transforming the reported results into practical and functional reports for the security administrator (Shah and Issac 2018). Hence, there is a large semantic gap between the results generated by the IDS and the operating environment (Shah and Issac 2018).
- Security at Various Layers: The network traffic exhibits diversity in terms of basic attributes of the network packets, and hence, there is need to aggregate the network data syntactically and semantically to include both network layer and application layer protocol information. Normally intrusion detection is carried out at the network layer. Therefore, constantly evolving network behavior and pattern of the web services should be taken into consideration.
- Smart Response Mechanism: It is very crucial to evaluate IDS in terms of finding the appropriate data for learning and classification as well as interpreting the results for reliable detection of the attack. Hence, appropriate dataset and performance metrics need to be used for evaluating the IDS.

Based on the challenges and the study performed in the paper, future research directions can be summarized as follows:

- A common evaluation and validation strategy must be used for performance comparison and evaluation of designed IDS. This might result in better attack detection and classification accuracy.
- There is a need to use representative datasets that consists of wide range of attacks and data instances. Moreover, variants of existing attacks can also be explored and data samples of such variants can be augmented and used for analysis.
- Network traffic should be collected from varied sources to maintain the diversity in the data. This can result in enhanced learning of IDS for detecting wide range of attack categories.
- While extracting features from network traffic, flow level as well as packet level features should be extracted to attain varied features for deriving patterns of network traffics.
- Network traffic should be collected from varied sources such as system logs, web applications, web server, routing tables, sensor nodes, to name a few.
- Features extracted should demonstrate the characteristics and details of various layers of network communication model such as network layer, transport layer, application layer, to name a few.
- Computationally efficient preprocessing techniques should be developed to handle high dimensional data of high speed networks without affecting its intrusion analysis capability.
- The performance of MLDL techniques with high dimensional dataset under dynamically changing environment can be explored by designing an efficient feature fusion technique that can even enhance the performance of classifier for intrusion detection.
- To reduce false alarms generated by IDS, adaptive approaches need to be proposed that can exhibit automated learning and have capability to handle constantly changing network data.
- Attacks are targeted at various layers of network communication model and therefore, security aspects of various layers should be explored by defining and discovering various attacks at different layers of communication model.
- To ensure reliability and scalability in the network as well as for detecting zero-day attacks, intrusion profiling should be performed to learn about different patterns of unknown attacks.
- Along with detecting intrusions, a smart response mechanism is needed for IDS that can report and perform action immediately as soon as intrusion is detected and can alert the security administrator for the same.
- To improve the attack detection and classification rate ensemble and hybrid techniques should be explored by syntactically and semantically analyzing working and understanding of functional aspects in context with individual techniques. This can help in minimizing semantic gap between IDS and operating environment.

9 Concluding remarks

The objective of this survey paper is to develop a clear understanding of Intrusion Detection System (IDS) and how it has been used for attack detection. We have considered Machine Learning (ML), Deep Learning (DL), and Swarm and Evolutionary Algorithms (SWEVO) based IDS techniques for attack detection and classification. We have conducted a study on taxonomy of IDS, feature engineering approaches, computational intelligence

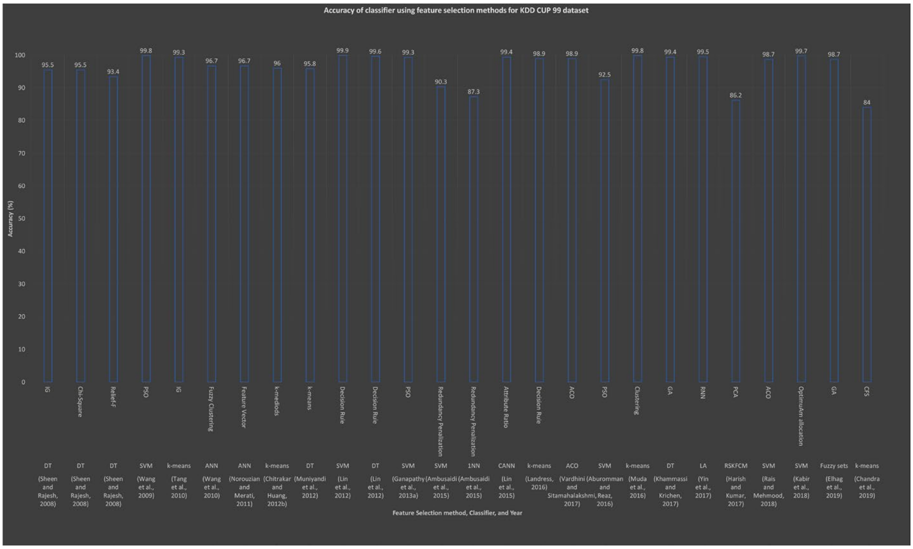

Fig. 12 Accuracy for ML, DL, and SWEVO techniques for KDD CUP 99 dataset

techniques used for IDS, datasets developed and used for performance evaluation of IDS, performance measures considered for evaluating the capability of IDS, and applicability of IDS in various fields is also discussed.

Over the years ML, DL, and SWEVO have proven their expertise in the varied field of research including security and intrusion detection. This paper is a comprehensive survey of the applicability of these methods in the field of Intrusion Detection. Classification and clustering techniques have touched the aspects of network monitoring analysis, performance optimization, and traffic engineering of IDS. We have summarized the literature work of IDS and have explored the viability and utility of different feature selection methods.

Network features are extracted from the network packets that reveal significant information regarding the attack pattern or attack signature. Hence, there is a need of representative dataset for evaluating the performance of IDS. Detection of an attack in the network can be carried out with a set of predefined rules and network features. Hence, selecting a good set of features can result in a high detection rate.

We have covered a range of research papers from 2008-2020 which include classification-based techniques, clustering-based techniques, DL-based techniques, and SWEVO-based techniques for IDS. These techniques can also be applied feature engineering to select significant features to improve the performance of IDS. These techniques have used different IDS datasets for performance evaluation. Experimental datasets such as KDD CUP 99 and NSL-KDD have been used as a benchmark dataset for the evaluation of the IDS. The accuracy and detection rate achieved by diffirent ML, DL, and SWEVO techniques with different feature selection methods for KDD CUP 99 dataset is shown in Figs. 12 and 13, respectively, and for the NSL-KDD dataset is shown in Figs. 14 and 15, respectively.

The limitations of each of the dataset are also discussed that suggests the need of a new dataset with a broad view of attack categories. The contribution of this survey can be summed as: representative literature survey of machine learning, deep learning, and

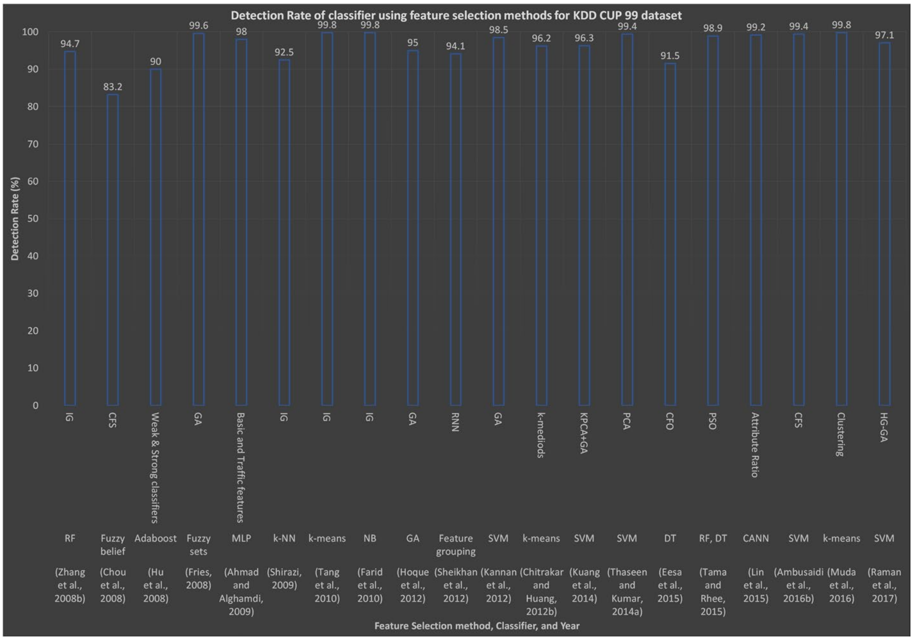

Fig. 13 Detection Rate for ML, DL, and SWEVO techniques for KDD CUP 99 dataset

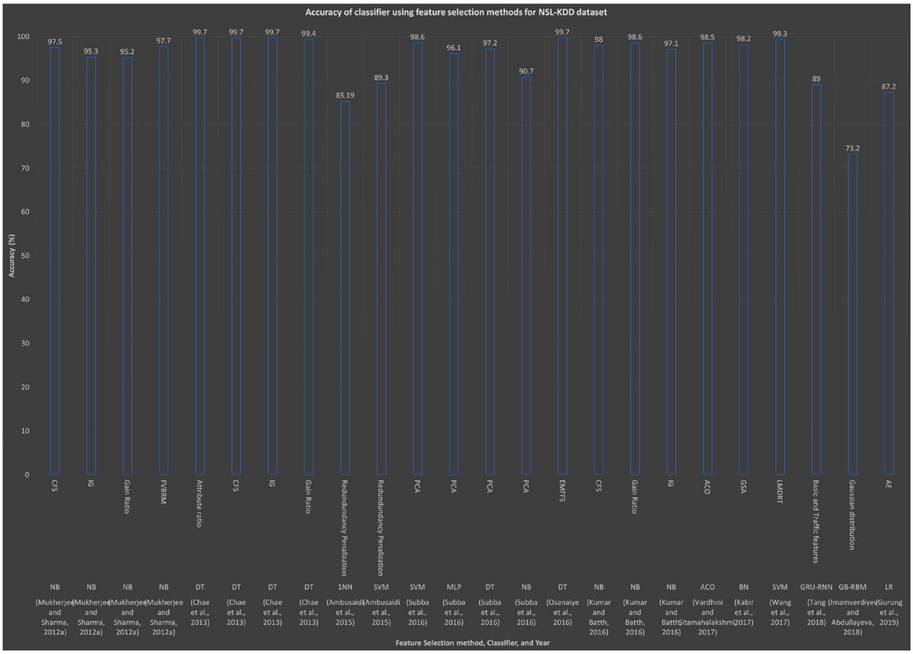

Fig. 14 Accuracy for ML, DL, and SWEVO techniques for NSL-KDD dataset

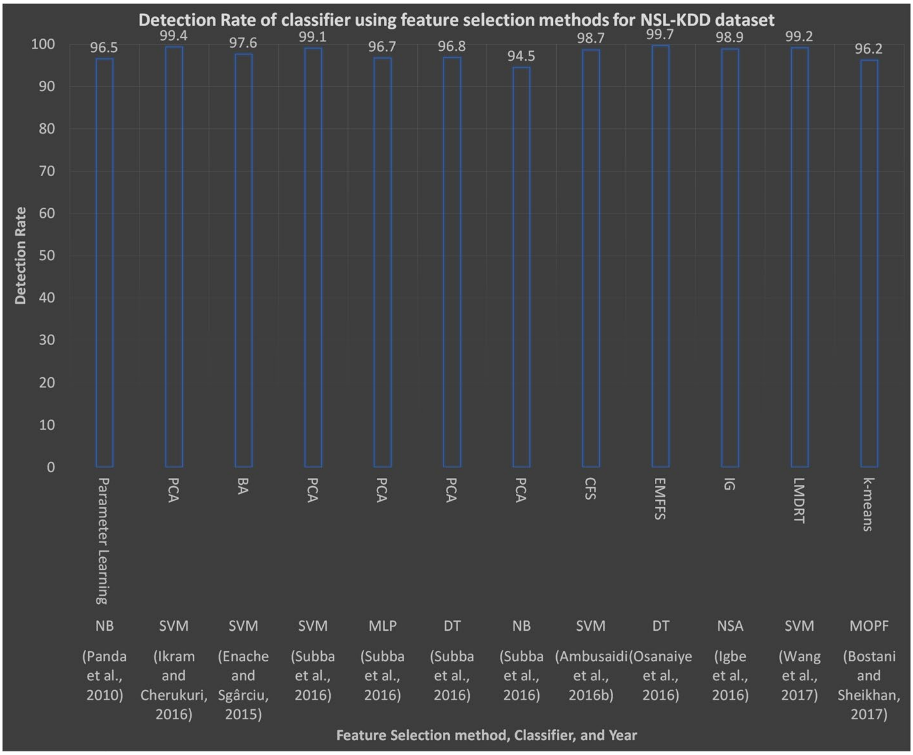

Fig. 15 Detection Rate for ML, DL, and SWEVO techniques for NSL-KDD dataset

swarm and evolutionary algorithms in IDS, importance of processing the data before using it for building the underlying model, significance of feature selection techniques to imporve the performance of a given model, survey of the trademark datasets and discussion on CIC-IDS-2017 dataset, study on model specific performance metrics, and applicability of IDS in various fields.

Hence, the future work will reinforce the study to explore different capabilities for using and leveraging the information provided in context with attack detection. Though ML techniques have showcased promising results for intrusion detection, still their scalability with real-time intrusion detection needs to be envisioned. The performance of ML techniques depends on different algorithmic parameters, type of dataset characteristics, and feature engineering techniques. Moreover, DL techniques exhibit flexibility in learning the data and represent the data in abstract and hierarchical form. Unlike ML techniques, DL techniques learn the features of the underlying dataset incrementally and this characteristic of DL techniques eliminates the need for incorporating feature selection or feature extraction method. However, a huge amount of training data is needed to achieve good performance by applying DL techniques. This results in increased training time and to minimize the same high-end computational devices are needed. Performance of ML and DL techniques may be enhanced by hybridizing these techniques with SWEVO algorithms. Existing work focuses on predicting the type of attack with the given dataset. Hence, these techniques should be implemented and extended to check the detection rate of single attacks among the listed attacks in the dataset. In the survey, we have also discussed the challenges and future research directions with the current IDS scenarios that may pave paths to explore research in the field of IDS.

Declarations

Conflict of Interest The authors declare that they have no conflict of interest.

References

Aburomman AA, Reaz MBI (2016) A novel SVM-kNN-PSO ensemble method for intrusion detection system. Appl Soft Comput 38:360–372

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

Agrawal S, Agrawal J (2015) Survey on anomaly detection using data mining techniques. Procedia Comput Sci 60:708–713

Ahmad AB Iftikhar and, Alghamdi AS (2009) Application of artificial neural network in detection of probing attacks. In: IEEE symposium on industrial electronics and applications, 2009. ISIEA 2009, vol 2. IEEE, pp 557–562

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

Ahmed M, Mahmood AN, Hu J (2016) A survey of network anomaly detection techniques. J Netw Comput Appl 60:19–31

Ahmim A, Maglaras L, Ferrag MA, Derdour M, Janicke H (2019) A novel hierarchical intrusion detection system based on decision tree and rules-based models. In: 2019 15th international conference on distributed computing in sensor systems (DCOSS). IEEE, pp 228–233

Akinyelu AA, Adewumi AO (2014) Classification of phishing email using random forest machine learning technique. J Appl Math 2014:1–6

Al-Emadi S, Al-Mohannadi A, Al-Senaid F (2020) Using deep learning techniques for network intrusion detection. In: 2020 IEEE international conference on informatics, IoT, and enabling technologies (ICIoT). IEEE, pp 171–176

Al-Janabi STF, Saeed HA (2011) A neural network based anomaly intrusion detection system. In: Developments in E-systems engineering (DeSE), 2011. IEEE, pp 221–226

Alelyani S, Tang J, Liu H (2018) Feature selection for clustering: a review. In: Data clustering. Chapman and Hall/CRC, pp 29–60

Alhaj TA, Siraj MM, Zainal A, Elshoush HT, Elhaj F (2016) Feature selection using information gain for improved structural-based alert correlation. PLoS ONE 11(11):e0166017

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

Ali MH, Fadlizolkipi M, Firdaus A, Khidzir NZ (2018b) A hybrid particle swarm optimization-extreme learning machine approach for intrusion detection system. In: 2018 IEEE student conference on research and development (SCOReD). IEEE, pp 1–4

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

Ambikavathi C, Srivatsa SK et al (2020) Predictor selection and attack classification using random forest for intrusion detection. J Sci Ind Res (JSIR) 79(05):365–368

Ambusaidi MA, He X, Nanda P (2015) Unsupervised feature selection method for intrusion detection system. In: 2015 IEEE Trustcom/BigDataSE/ISPA, IEEE, vol 1, pp 295–301

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

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

Ampah NK, Akujuobi CM, Sadiku MN, Alam S (2011) An intrusion detection technique based on continuous binary communication channels. Int J Secure Netw 6(2–3):174–180

Anwar S, Mohamad Zain J, Zolkipli MF, Inayat Z, Khan S, Anthony B, Chang V (2017) From intrusion detection to an intrusion response system: fundamentals, requirements, and future directions. Algorithms 10(2):39

Aslahi-Shahri B, Rahmani R, Chizari M, Maralani A, Eslami M, Golkar M, Ebrahimi A (2016) A hybrid method consisting of GA and SVM for intrusion detection system. Neural Comput Appl 27(6):1669–1676

Bajtoš T, Gajdoš A, Kleinová L, Lučivjanská K, Sokol P (2018) Network intrusion detection with threat agent profiling. Security and Communication Networks (2018)

Barbhuiya FA, Bansal G, Kumar N, Biswas S, Nandi S (2013) Detection of neighbor discovery protocol based attacks in ipv6 network. Netw Sci 2(3–4):91–113

Barrett M (2018) Framework for improving critical infrastructure cybersecurity. Technical Report, National Institute of Standards and Technology, Gaithersburg, MD, USA

Basnet R, Mukkamala S, Sung AH (2008) Detection of phishing attacks: a machine learning approach. In: Soft computing applications in industry. Springer, pp 373–383

Bennasar M, Hicks Y, Setchi R (2015) Feature selection using joint mutual information maximisation. Expert Syst Appl 42(22):8520–8532

Bhati BS, Rai C (2020) Analysis of support vector machine-based intrusion detection techniques. Arab J Sci Eng 45(4):2371–2383

Bhuyan MH, Bhattacharyya DK, Kalita JK (2014) Network anomaly detection: methods, systems and tools. IEEE Commun Surv Tutor 16(1):303–336

Blum A, Wardman B, Solorio T, Warner G (2010) Lexical feature based phishing url detection using online learning. In: Proceedings of the 3rd ACM workshop on artificial intelligence and security. ACM, pp 54–60

Bollinger J, Enright B, Valites M (2015) Crafting the InfoSec playbook: security monitoring and incident response master plan. O'Reilly Media Inc, Newton

Bostani H, Sheikhan M (2017) Modification of supervised OPF-based intrusion detection systems using unsupervised learning and social network concept. Pattern Recogn 62:56–72

Brown C, Cowperthwaite A, Hijazi A, Somayaji A (2009) Analysis of the 1999 Darpa/Lincoln laboratory IDs evaluation data with netadhict. In: IEEE symposium on computational intelligence for security and defense applications (2009), CISDA 2009. IEEE, pp 1–7

Buczak AL, Guven E (2016) A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Commun Surv Tutor 18(2):1153–1176

Bujlow T, Riaz T, Pedersen JM (2012) A method for classification of network traffic based on C5. 0 Machine Learning Algorithm. In: 2012 international conference on computing, networking and communications (ICNC). IEEE, pp 237–241

Callado A, Kamienski C, Szabó G, Gero BP, Kelner J, Fernandes S, Sadok D (2009) A survey on internet traffic identification. IEEE Commun Surv Tutor 11(3):37–52

Canzanese R, Mancoridis S, Kam M, (2015) System call-based detection of malicious processes. In: IEEE international conference on software quality, reliability and security. IEEE, pp 119–124

Carneiro G (2010) NS-3: Network simulator 3. In: UTM Lab Meeting April, vol 20, pp 4–5

Carrasquilla U (2010) Benchmarking algorithms for detecting anomalies in large datasets. MeasureIT, Nov pp 1–16

Casey E (2011) Digital evidence and computer crime: forensic science, computers, and the internet. Academic Press, London

Chae Hs, Jo Bo, Choi SH, Park Tk (2013) Feature selection for intrusion detection using NSL-KDD. In: Recent advances in computer science, pp 184–187

Chakraborty A, Bhattacharjee S, Marsden JR, Shankar R, Katz ES, Vallee WL Jr (2018) Predictive models to measure the impact of fiber-optic broadband speeds on local towns and communities. Telematics Inform 35(5):1408–1420

Chandala V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. University of Minnesota, ACM Computing Surveys

Chandra A, Khatri SK, Simon R (2019) Filter-based attribute selection approach for intrusion detection using k-means clustering and sequential minimal optimization technique. In: 2019 Amity international conference on artificial intelligence (AICAI). IEEE, pp 740–745

Chaudhari K, Thakkar A (2019a) A comprehensive survey on travel recommender systems. Arch Comput Methods Eng 27:1–27

Chaudhari K, Thakkar A (2019b) Survey on handwriting-based personality trait identification. Expert Syst Appl 124:282–308

Chaudhari K, Thakkar A (2019c) Travelling salesman problem: an empirical comparison between ACO, PSO, ABC, FA and GA. In: Emerging research in computing, information, communication and applications. Springer, pp 397–405

Chebrolu S, Abraham A, Thomas JP (2005) Feature deduction and ensemble design of intrusion detection systems. Comput Secur 24(4):295–307

Chen L, Yan Z, Zhang W, Kantola R (2015) Trusms: a trustworthy SMS spam control system based on trust management. Fut Gener Comput Syst 49:77–93

Chen J, Qi X, Chen L, Chen F, Cheng G (2020) Quantum-inspired ant lion optimized hybrid k-means for cluster analysis and intrusion detection. Knowl Based Syst 203:106167

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

Choi SS, Cha SH, Tappert CC (2010) A survey of binary similarity and distance measures. J Syst Cybern Inform 8(1):43–48

Chou TS, Yen KK, Luo J (2008) Network intrusion detection design using feature selection of soft computing paradigms. Int J Comput Intell 4(3):196–208

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

Daş R, Karabade A, Tuna G (2015) Common network attack types and defense mechanisms. In: 2015 23nd signal processing and communications applications conference (SIU). IEEE, pp 2658–2661

Dasgupta D (2012) Artificial immune systems and their applications. Springer, Berlin

Dash M, Koot PW (2009) Feature selection for clustering. In: Encyclopedia of database systems. Springer, pp 1119–1125

Dembczynski K, Jachnik A, Kotlowski W, Waegeman W, Hüllermeier E (2013) Optimizing the f-measure in multi-label classification: plug-in rule approach versus structured loss minimization. In: International conference on machine learning, pp 1130–1138

Deshpande P, Sharma SC, Peddoju SK, Junaid S (2018) HIDS: a host based intrusion detection system for cloud computing environment. Int J Syst Assur Eng Manag 9(3):567–576

Devaraju S, Ramakrishnan S (2014) Performance comparison for intrusion detection system using neural network with KDD dataset. ICTACT J Soft Comput 4(3):106167

Dewa Z, Maglaras LA (2016) Data mining and intrusion detection systems. Int J Adv Comput Sci Appl 7(1):62–71

Doak J (1992) CSE-92-18-an evaluation of feature selection methods and their application to computer security

Duarte V, Farruca N (2010) Using libpcap for monitoring distributed applications. In: 2010 international conference on high performance computing and simulation. IEEE, pp 92–97

Duch W, Winiarski T, Biesiada J, Kachel A (2003) Feature selection and ranking filters. In: International conference on artificial neural networks (ICANN) and international conference on neural information processing (ICONIP), Citeseer, vol 251, p 254

Dutta V, Choraś M, Pawlicki M, Kozik R (2020) Hybrid model for improving the classification effectiveness of network intrusion detection. In: Proceedings of the 13th international conference on computational intelligence in security for information systems (CISIS 2020), Burgos, Spain, pp 18–20

Dy JG, Brodley CE (2000) Feature subset selection and order identification for unsupervised learning. In: ICML, Citeseer, pp 247–254

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

Elhag S, Fernández A, Alshomrani S, Herrera F (2019) Evolutionary fuzzy systems: a case study for intrusion detection systems. In: Evolutionary and swarm intelligence algorithms. Springer, pp 169–190

Elmasry W, Akbulut A, Zaim AH (2020) Evolving deep learning architectures for network intrusion detection using a double pso metaheuristic. Comput Netw 168:107042

Elsherif A et al. (2018) Automatic intrusion detection system using deep recurrent neural network paradigm. J Inf Secur Cybercrimes Res 1(1):21–31

Emary E, Zawbaa HM, Ghany KKA, Hassanien AE, Parv B (2015) Firefly optimization algorithm for feature selection. In: Proceedings of the 7th Balkan conference on informatics conference. ACM, p 26

Enache AC, Sgârciu V (2015) An improved bat algorithm driven by support vector machines for intrusion detection. In: International joint conference. Springer, pp 41–51

Farid DM, Harbi N, Rahman MZ (2010) Combining naive bayes and decision tree for adaptive intrusion detection. arXiv preprint arXiv:10054496

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

Fausett LV et al (1994) Fundamentals of neural networks: architectures, algorithms, and applications, vol 3. Prentice-Hall, Englewood Cliffs

Fleuret F (2004) Fast binary feature selection with conditional mutual information. J Mach Learn Res 5(Nov):1531–1555

Fowlkes EB, Mallows CL (1983) A method for comparing two hierarchical clusterings. J Am Stat Assoc 78(383):553–569

Fries TP (2008) A fuzzy-genetic approach to network intrusion detection. In: Proceedings of the 10th annual conference companion on Genetic and evolutionary computation. ACM, pp 2141–2146

Gamal M, Abbas H, Sadek R (2020) Hybrid approach for improving intrusion detection based on deep learning and machine learning techniques. In: Joint European-US workshop on applications of invariance in computer vision. Springer, pp 225–236

Ganapathy S, Kulothungan K, Muthurajkumar S, Vijayalakshmi M, Yogesh P, Kannan A (2013) Intelligent feature selection and classification techniques for intrusion detection in networks: a survey. EURASIP J Wirel Commun Netw 1:271

Gao HH, Yang HH, Wang XY (2005) Ant colony optimization based network intrusion feature selection and detection. In: Proceedings of 2005 international conference on machine learning and cybernetics, vol 6, 2005. IEEE, pp 3871–3875

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

Ghosh J, Kumar D, Tripathi R (2020) Features extraction for network intrusion detection using genetic algorithm (GA). In: Modern approaches in machine learning and cognitive science: a walkthrough. Springer, pp 13–25

Goeschel K (2016) Reducing false positives in intrusion detection systems using data-mining techniques utilizing support vector machines, decision trees, and naive Bayes for off-line analysis. In: SoutheastCon 2016. IEEE, pp 1–6

Grzonka D, Jakobik A, Kołodziej J, Pllana S (2018) Using a multi-agent system and artificial intelligence for monitoring and improving the cloud performance and security. Future Gener Comput Syst 86:1106–1117

Gupta B, Agrawal DP, Yamaguchi S (2016a) Handbook of research on modern cryptographic solutions for computer and cyber security. IGI Global, Hershey

Gupta S, Chaudhari BS, Chakrabarty B (2016b) Vulnerable network analysis using war driving and security intelligence. In: 2016 international conference on inventive computation technologies (ICICT), vol 3. IEEE, pp 1–5

Gurung S, Ghose MK, Subedi A (2019) Deep learning approach on network intrusion detection system using NSL-KDD dataset. Int J Comput Netw Inf Secur (IJCNIS) 11(3):8–14

Halkidi M, Batistakis Y, Vazirgiannis M (2001) On clustering validation techniques. J Intell Inf Syst 17(2–3):107–145

Hamamoto AH, Carvalho LF, Sampaio LDH, Abrão T, Proença ML Jr (2018) Network anomaly detection system using genetic algorithm and fuzzy logic. Expert Syst Appl 92:390–402

Hamed T, Ernst JB, Kremer SC (2018) A survey and taxonomy of classifiers of intrusion detection systems. In: Computer and network security essentials. Springer, pp 21–39

Harish B, Kumar SA (2017) Anomaly based intrusion detection using modified fuzzy clustering. IJIMAI 4(6):54–59

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

Heck RH, Thomas S, Tabata L (2013) Multilevel modeling of categorical outcomes using IBM SPSS. Routledge, London

Henriques J, Caldeira F, Cruz T, Simões P (2020) Combining k-means and xgboost models for anomaly detection using log datasets. Electronics 9(7):1164

Hindy H, Atkinson R, Tachtatzis C, Colin JN, Bayne E, Bellekens X (2020) Towards an effective zero-day attack detection using outlier-based deep learning techniques. arXiv preprint arXiv:200615344

Hodge V, Austin J (2004) A survey of outlier detection methodologies. Artif Intell Rev 22(2):85–126

Hodo E, Bellekens X, Hamilton A, Tachtatzis C, Atkinson R (2017) Shallow and deep networks intrusion detection system: a taxonomy and survey. arXiv preprint arXiv:170102145

Hoque MS, Mukit M, Bikas M, Naser A et al. (2012) An implementation of intrusion detection system using genetic algorithm. arXiv preprint arXiv:12041336

Hosseini S, Zade BMH (2020) New hybrid method for attack detection using combination of evolutionary algorithms, SVM, and ANN. Comput Netw 173:107168

Hu W, Hu W, Maybank S (2008) Adaboost-based algorithm for network intrusion detection. IEEE Trans Syst Man Cybern Part B Cybern 38(2):577–583

Igbe O, Darwish I, Saadawi T (2016) Distributed network intrusion detection systems: an artificial immune system approach. In: 2016 IEEE first international conference on connected health: applications, systems and engineering technologies (CHASE). IEEE, pp 101–106

Ikram ST, Cherukuri AK (2016) Improving accuracy of intrusion detection model using PCA and optimized SVM. J Comput Inf Technol 24(2):133–148

Imamverdiyev Y, Abdullayeva F (2018) Deep learning method for denial of service attack detection based on restricted Boltzmann machine. Big Data 6(2):159–169

Inayat Z, Gani A, Anuar NB, Khan MK, Anwar S (2016) Intrusion response systems: foundations, design, and challenges. J Netw Comput Appl 62:53–74

Index CVN (2017) Global mobile data traffic forecast update, 2016–2021 white paper. Cisco, San Jose

Islam R, Abawajy J (2013) A multi-tier phishing detection and filtering approach. J Netw Comput Appl 36(1):324–335

Javaid A, Niyaz Q, Sun W, Alam M (2016) A deep learning approach for network intrusion detection system. In: Proceedings of the 9th EAI international conference on bio-inspired information and communications technologies (formerly BIONETICS), ICST (Institute for Computer Sciences, and Social-Informatics), pp 21–26

John GH, Kohavi R, Pfleger K (1994) Irrelevant features and the subset selection problem. In: Machine learning proceedings. Elsevier, pp 121–129

Jović A, Brkić K, Bogunović N (2015) A review of feature selection methods with applications. In: 2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO). IEEE, pp 1200–1205

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

Kabir MM, Shahjahan M, Murase K (2012) A new hybrid ant colony optimization algorithm for feature selection. Expert Syst Appl 39(3):3747–3763

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

Kalita DJ, Singh VP, Kumar V (2020) SVM hyper-parameters optimization using multi-PSO for intrusion detection. In: Social networking and computational intelligence. Springer, pp 227–241

Kamisiński A, Fung C (2015) Flowmon: detecting malicious switches in software-defined networks. In: Proceedings of the 2015 workshop on automated decision making for active cyber defense. ACM, pp 39–45

Kannan A, Maguire GQ, Sharma A, Schoo P (2012) Genetic algorithm based feature selection algorithm for effective intrusion detection in cloud networks. In: 2012 IEEE 12th international conference on data mining workshops (ICDMW). IEEE, pp 416–423

Kelleher JD, Mac Namee B, D'arcy A, (2015) Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies. MIT Press, Cambridge

Khammassi C, Krichen S (2017) A GA-LR wrapper approach for feature selection in network intrusion detection. Comput Secur 70:255–277

Khan S, Gani A, Wahab AWA, Singh PK (2018) Feature selection of denial-of-service attacks using entropy and granular computing. Ara J Sci Eng 43(2):499–508

Khraisat A, Gondal I, Vamplew P, Kamruzzaman J, Alazab A (2020) Hybrid intrusion detection system based on the stacking ensemble of c5 decision tree classifier and one class support vector machine. Electronics 9(1):173

Kim J, Kim H (2015) Applying recurrent neural network to intrusion detection with hessian free optimization. In: International workshop on information security applications. Springer, pp 357–369

Kim TY, Cho SB (2018) Web traffic anomaly detection using C-LSTM neural networks. Expert Syst Appl 106:66–76

Knight W (2018) MIT technology review. Serious quantum computers are finally here What are we going to do with them

Kolari P, Java A, Finin T, Oates T, Joshi A, et al. (2006) Detecting spam blogs: a machine learning approach. In: Proceedings of the national conference on artificial intelligence, vol 21. MIT Press, Cambridge, p 1351

Kondaiah R, Sathyanarayana B (2018) Trust based genetic neuro-fuzzy system for intrusion detection and self adaptive firefly integrated particle swarm optimization algorithm for secure routing in manet. Int J Appl Eng Res 13(8):5722–5735

Krishnaveni S, Vigneshwar P, Kishore S, Jothi B, Sivamohan S (2020) Anomaly-based intrusion detection system using support vector machine. In: Artificial intelligence and evolutionary computations in engineering systems. Springer, pp 723–731

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

Kumar G (2020) An improved ensemble approach for effective intrusion detection. J Supercomput 76(1):275–291

Kumar K, Batth JS (2016) Network intrusion detection with feature selection techniques using machine-learning algorithms. Int J Comput Appl 150(12):1–13

Kumar GR, Mangathayaru N, Narasimha G (2015) An improved k-means clustering algorithm for intrusion detection using Gaussian function. In: Proceedings of the the international conference on engineering & MIS 2015. ACM, p 69

Landress AD (2016) A hybrid approach to reducing the false positive rate in unsupervised machine learning intrusion detection. In: SoutheastCon 2016. IEEE, pp 1–6

Lashkari AH, Draper-Gil G, Mamun MSI, Ghorbani AA (2017) Characterization of tor traffic using time based features. In: ICISSP, pp 253–262

Lever J, Krzywinski M, Altman N (2016) Points of significance: model selection and overfitting

Li Y, Wang JL, Tian ZH, Lu TB, Young C (2009) Building lightweight intrusion detection system using wrapper-based feature selection mechanisms. Comput Secur 28(6):466–475

Li J, Liu Y, Gu L (2010) DDoS attack detection based on neural network. In: 2010 2nd international symposium on aware computing (ISAC). IEEE, pp 196–199

Li WS, Bai XM, Duan LZ, Zhang X (2011) Intrusion Detection based on ant colony algorithm of Fuzzy clustering. In: 2011 international conference on computer science and network technology (ICCSNT), vol 3. IEEE, pp 1642–1645

Li L, Zhang H, Peng H, Yang Y (2018) Nearest neighbors based density peaks approach to intrusion detection. Chaos Solitons Fractals 110:33–40

Li W, Tug S, Meng W, Wang Y (2019) Designing collaborative blockchained signature-based intrusion detection in IoT environments. Future Gener Comput Syst 96:481–489

Lin SW, Ying KC, Lee CY, Lee ZJ (2012) An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection. Appl Soft Comput 12(10):3285–3290

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

Liu H, Motoda H (2012) Feature selection for knowledge discovery and data mining, vol 454. Springer, Berlin

Liu Y, Liang S, Fang W, Zhou Z, Hu R, Zhou H, Hou J, Wang Y (2020) A hybrid feature selection algorithm combining information gain and genetic search for intrusion detection. J. Phys. Conf. Ser. 1601:032048

Lohiya R, Thakkar A (2020) Application domains, evaluation datasets, and research challenges of IoT: a systematic review. IEEE Internet Things J

Lohiya R, Thakkar A (In press) Intrusion detection using deep neural network with antirectifier layer. In: International conference on applied soft computing and communication networks (ACN’20), ISBN 978-981-33-6173-7_7

Lv L, Wang W, Zhang Z, Liu X (2020) A novel intrusion detection system based on an optimal hybrid kernel extreme learning machine. Knowl Based Syst 105648

Mabu S, Chen C, Lu N, Shimada K, Hirasawa K (2011) An intrusion-detection model based on fuzzy class-association-rule mining using genetic network programming. IEEE Trans Syst Man Cybern Part C Appl Rev 41(1):130–139

Mandal N, Jadhav S (2016) A survey on network security tools for open source. In: 2016 IEEE international conference on current trends in advanced computing (ICCTAC). IEEE, pp 1–6

McHugh J (2000) Testing intrusion detection systems: a critique of the 1998 and 1999 Darpa intrusion detection system evaluations as performed by Lincoln laboratory. ACM Trans Inf Syst Secur (TISSEC) 3(4):262–294

McHugh ML (2012) Interrater reliability: the kappa statistic. Biochemia medica: Biochemia medica 22(3):276–282

Meftah S, Rachidi T, Assem N (2019) Network based intrusion detection using the UNSW-NB15 dataset. Int J Comput Digit Syst 8(5):478–487

Mehra P (2012) A brief study and comparison of snort and bro open source network intrusion detection systems. Int J Adv Res Comput Commun Eng 1(6):383–386

Migliavacca M, Papagiannis I, Eyers DM, Shand B, Bacon J, Pietzuch P (2010) DEFCON: high-performance event processing with information security. In: Proceedings of the 2010 USENIX conference on USENIX annual technical conference, USENIX Association, pp 1–1

Mitra P, Murthy C, Pal SK (2002) Unsupervised feature selection using feature similarity. IEEE Trans Pattern Anal Mach Intell 24(3):301–312

Moustafa N, Slay J (2015) UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: 2015 military communications and information systems conference (MilCIS), IEEE, pp 1–6

Muda Z, Yassin W, Sulaiman M, Udzir N (2016) K-means clustering and Naive Bayes classification for intrusion detection. J IT Asia 4(1):13–25

Mukherjee S, Sharma N (2012) Intrusion detection using Naive Bayes classifier with feature reduction. Procedia Technol 4:119–128

Mungra D, Agrawal A, Thakkar A (2020) A voting-based sentiment classification model. In: Intelligent communication, control and devices. Springer, pp 551–558

Muniyandi AP, Rajeswari R, Rajaram R (2012) Network anomaly detection by cascading k-Means clustering and C4. 5 decision tree algorithm. Procedia Eng 30:174–182

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

Naseer S, Saleem Y, Khalid S, Bashir MK, Han J, Iqbal MM, Han K (2018) Enhanced network anomaly detection based on deep neural networks. IEEE Access 6:48231–48246

Nechaev B, Allman M, Paxson V, Gurtov A (2004) Lawrence Berkeley National Laboratory (LBNL)/ICSI enterprise tracing project. LBNL/ICSI, Berkeley

Nehinbe JO (2009) A simple method for improving intrusion detections in corporate networks. In: International conference on information security and digital forensics. Springer, pp 111–122

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

Nguyen TT, Armitage G (2008) A survey of techniques for internet traffic classification using machine learning. IEEE Commun Surv Tutor 10(4):56–76

Ni X, He D, Chan S, Ahmad F (2016) Network anomaly detection using unsupervised feature selection and density peak clustering. In: International conference on applied cryptography and network security. Springer, pp 212–227

Nigam K, McCallum AK, Thrun S, Mitchell T (2000) Text classification from labeled and unlabeled documents using EM. Mach Learn 39(2–3):103–134

Nisioti A, Mylonas A, Yoo PD, Katos V (2018) From intrusion detection to attacker attribution: a comprehensive survey of unsupervised methods. IEEE Commun Surv Tutor 20(4):3369–3388

Niu Z, Shi S, Sun J, He X (2011) A survey of outlier detection methodologies and their applications. In: International conference on artificial intelligence and computational intelligence. Springer, pp 380–387

Norouzian MR, Merati S (2011) Classifying attacks in a network intrusion detection system based on artificial neural networks. In: 2011 13th international conference on advanced communication technology (ICACT). IEEE, pp 868–873

Novaković J (2016) Toward optimal feature selection using ranking methods and classification algorithms. Yugoslav J Oper Res 21(1):119–135

Obermeyer Z, Emanuel EJ (2016) Predicting the future-big data, machine learning, and clinical medicine. New Engl J Med 375(13):1216

OpenDNS L (2016) PhishTank: an anti-phishing site. https://www.phishtank.com

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 Wirel Commun Netw 1:130

Panda M, Abraham A, Patra MR (2010) Discriminative multinomial Naive Bayes for network intrusion detection. In: 2010 sixth international conference on information assurance and security (IAS). IEEE, pp 5–10

Pandya R, Pandya J (2015) C5.0 algorithm to improved decision tree with feature selection and reduced error pruning. Int J Comput Appl 117(16):18–21

Pareek P, Thakkar A (2021) A survey on video-based human action recognition: recent updates, datasets, challenges, and applications. Artif Intell Rev 54(3):2259–2322

Patel C, Patel R, Thakkar A (2012a) Object detection and segmentation using local and global property. Int J Comput Sci Res Appl 2(02):02–10

Patel R, Patel CI, Thakkar A (2012b) Aggregate features approach for texture analysis. In: 2012 Nirma University international conference on engineering (NUiCONE). IEEE, pp 1–5

Patgiri R, Varshney U, Akutota T, Kunde R (2018) An investigation on intrusion detection system using machine learning. In: 2018 IEEE symposium series on computational intelligence (SSCI). IEEE, pp 1684–1691

Peddabachigari S, Abraham A, Grosan C, Thomas J (2007) Modeling intrusion detection system using hybrid intelligent systems. J Netw Comput Appl 30(1):114–132

Peng H, Long F, Ding C (2005) Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans Pattern Anal Mach Intell 27(8):1226–1238

Peng J, Choo KKR, Ashman H (2016) User profiling in intrusion detection: a review. J Netw Comput Appl 72:14–27

Potluri S, Henry NF, Diedrich C (2017) Evaluation of hybrid deep learning techniques for ensuring security in networked control systems. In: 2017 22nd IEEE international conference on emerging technologies and factory automation (ETFA). IEEE, pp 1–8

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

Rais HM, Mehmood T (2018) Dynamic ant colony system with three level update feature selection for intrusion detection. IJ Netw Secur 20(1):184–192

Rajagopal S, Kundapur PP, Hareesha KS (2020) A stacking ensemble for network intrusion detection using heterogeneous datasets. Secur Commun Netw 2020:1–9

Rajput D, Thakkar A (2019) A survey on different network intrusion detection systems and countermeasure. In: Emerging research in computing, information, communication and applications. Springer, pp 497–506

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. Knowl-Based Syst 134:1–12

Ranshous S, Shen S, Koutra D, Harenberg S, Faloutsos C, Samatova NF (2015) Anomaly detection in dynamic networks: a survey. Wiley Interdiscip Rev Comput Stat 7(3):223–247

Riquelme N, Von Lücken C, Baran B (2015) Performance metrics in multi-objective optimization. In: 2015 Latin American computing conference (CLEI). IEEE, pp 1–11

Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 20:53–65

Sabahi F, Movaghar A (2008) Intrusion detection: a survey. In: 3rd International Conference on Systems and Networks Communications, 2008. ICSNC'08. IEEE, pp 23–26

Saeys Y, Inza I, Larrañaga P (2007) A review of feature selection techniques in bioinformatics. Bioinformatics 23(19):2507–2517

Sahu S, Mehtre BM (2015) Network intrusion detection system using J48 Decision Tree. In: 2015 international conference on advances in computing, communications and informatics (ICACCI). IEEE, pp 2023–2026

Sánchez-Maroño N, Alonso-Betanzos A, Tombilla-Sanromán M (2007) Filter methods for feature selection–a comparative study. In: International conference on intelligent data engineering and automated learning. Springer, pp 178–187

Sangkatsanee P, Wattanapongsakorn N, Charnsripinyo C (2011) Practical real-time intrusion detection using machine learning approaches. Comput Commun 34(18):2227–2235

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

Sarvari S, Sani NFM, Hanapi ZM, Abdullah MT (2020) An efficient anomaly intrusion detection method with feature selection and evolutionary neural network. IEEE Access 8:70651–70663

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

Schneier B, Kelsey J (1999) Secure audit logs to support computer forensics. ACM Trans Inf Syst Secur (TISSEC) 2(2):159–176

Shah SAR, Issac B (2018) Performance comparison of intrusion detection systems and application of machine learning to snort system. Future Gener Comput Syst 80:157–170

Sharafaldin I, Lashkari AH, Ghorbani AA (2018) Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: ICISSP, pp 108–116

Sharbaf M (2018) Artificial intelligence and cybersecurity. Bus Strategy Artif Intell Econ 5

Sharma R, Rajvaidya H, Pareek P, Thakkar A (2019) A comparative study of machine learning techniques for emotion recognition. In: Emerging research in computing, information, communication and applications. Springer, pp 459–464

Sheen S, Rajesh R (2008) Network intrusion detection using feature selection and Decision tree classifier. In: TENCON 2008—2008 IEEE Region 10 conference. IEEE, pp 1–4

Sheikhan M, Jadidi Z, Farrokhi A (2012) Intrusion detection using reduced-size RNN based on feature grouping. Neural Comput Appl 21(6):1185–1190

Sheth N, Yong L, Callon R, Black D (2015) Juniper networks

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

Shirazi HM (2009) Anomaly intrusion detection system using information theory, K-NN and KMC algorithms. Aust J Basic Appl Sci 3(3):2581–2597

Snapp SR, Brentano J, Dias G, Goan TL, Heberlein LT, Ho CL, Levitt KN (2017) DIDS (distributed intrusion detection system)-motivation, architecture, and an early prototype

Soheily-Khah S, Marteau PF, Béchet N (2018) Intrusion detection in network systems through hybrid supervised and unsupervised machine learning process: a case study on the ISCX dataset. In: 2018 1st international conference on data intelligence and security (ICDIS). IEEE, pp 219–226

Sommer P (1999) Intrusion detection systems as evidence. Comput Netw 31(23–24):2477–2487

Song J, Alves-Foss J (2016) The DARPA cyber grand challenge: a competitor's perspective, part 2. IEEE Secur Priv 14(1):76–81

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

Song Q, Ni J, Wang G (2013) A fast clustering-based feature subset selection algorithm for high-dimensional data. IEEE Trans Knowl Data Eng 25(1):1–14

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

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

Søhoel H, Jaatun MG, Boyd C (2018) OWASP Top 10-Do Startups Care? In: 2018 international conference on cyber security and protection of digital services (Cyber Security). IEEE, pp 1–8

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

Su MY (2011) Using clustering to improve the KNN-based classifiers for online anomaly network traffic identification. J Netw Comput Appl 34(2):722–730

Subba B, Biswas S, Karmakar S (2016) Enhancing performance of anomaly based intrusion detection systems through dimensionality reduction using principal component analysis. In: 2016 IEEE international conference on advanced networks and telecommunications systems (ANTS). IEEE, pp 1–6

Sung AH, Mukkamala S (2004) The feature selection and intrusion detection problems. In: Annual Asian computing science conference. Springer, pp 468–482

Suresh M, Anitha R (2011) Evaluating machine learning algorithms for detecting DDoS attacks. In: International conference on network security and applications. Springer, pp 441–452

Susilo B, Sari RF (2020) Intrusion detection in IoT networks using deep learning algorithm. Information 11(5):279

Talavera L (2005) An evaluation of filter and wrapper methods for feature selection in categorical clustering. In: International symposium on intelligent data analysis. Springer, pp 440–451

Tama BA, Rhee KH (2015) A combination of PSO-based feature selection and tree-based classifiers ensemble for intrusion detection systems. In: Advances in computer science and ubiquitous computing. Springer, pp 489–495

Tang P, Jiang Ra, Zhao M (2010) Feature selection and design of intrusion detection system based on k-means and triangle area support vector machine. In: Second international conference on future networks, 2010. ICFN'10. IEEE, pp 144–148

Tang TA, Mhamdi L, McLernon D, Zaidi SAR, Ghogho M (2018) Deep recurrent neural network for intrusion detection in SDN-based networks. In: 2018 4th IEEE conference on network softwarization and workshops (NetSoft). IEEE, pp 202–206

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

Thakkar A, Chaudhari K (2020a) A comprehensive survey on portfolio optimization, stock price and trend prediction using particle swarm optimization. Arch Comput Methods Eng 28(4):2133-–2164

Thakkar A, Chaudhari K (2020b) Crest: cross-reference to exchange-based stock trend prediction using long short-term memory. Procedia Comput Sci 167:616–625

Thakkar A, Chaudhari K (2020c) Predicting stock trend using an integrated term frequency-inverse document frequency-based feature weight matrix with neural networks. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2020.106684

Thakkar A, Chaudhari K (2021) Fusion in stock market prediction: a decade survey on the necessity, recent developments, and potential future directions. Inf Fusion 65:95–107

Thakkar A, Kotecha K (2011) Bio-inspired based optimized algorithm for cluster head election using RSSI and LQI. Int J Comput Sci 1(02):19–29

Thakkar A, Lohiya R (2020a) A review of the advancement in intrusion detection datasets. Procedia Comput Sci 167:636–645

Thakkar A, Lohiya R (2020b) Role of swarm and evolutionary algorithms for intrusion detection system: a survey. In: Swarm and evolutionary computation, p 100631

Thakkar A, Lohiya R (2021a) Attack classification using feature selection techniques: a comparative study. J. Ambient Intell Human Comput 12(1):1249–1266

Thakkar A, Lohiya R (2021b) A review on machine learning and deep learning perspectives of ids for IoT: recent updates, security issues, and challenges. Arch Comput Methods Eng 28(4):3211–3243

Thakkar A, Jivani N, Padasumbiya J, Patel CI (2013) A new hybrid method for face recognition. In: 2013 Nirma University international conference on engineering (NUiCONE). IEEE, pp 1–9

Thakkar A, Mungra D, Agrawal A (2020) Sentiment analysis: an empirical comparison between various training algorithms for artificial neural network. Int J Innov Comput Appl 11(1):9–29

Thaseen IS, Kumar CA (2014) Intrusion detection model using fusion of PCA and optimized SVM. In: 2014 international conference on contemporary computing and informatics (IC3I). IEEE, pp 879–884

Thaseen IS, Kumar CA (2017) Intrusion detection model using fusion of chi-square feature selection and multi class SVM. J King Saud Univ Comput Inf Sci 29(4):462–472

Tian J, Gu H (2010) Anomaly detection combining one-class SVMs and particle swarm optimization algorithms. Nonlinear Dyn 61(1–2):303–310

Tsoumakas G, Katakis I, Vlahavas I (2010) Data mining and knowledge discovery handbook. Mining multilabel data

Udd R, Asplund M, Nadjm-Tehrani S, Kazemtabrizi M, Ekstedt M (2016) Exploiting bro for intrusion detection in a SCADA system. In: Proceedings of the 2nd ACM international workshop on cyber-physical system security. ACM, pp 44–51

Umer MF, Sher M, Bi Y (2017) Flow-based intrusion detection: techniques and challenges. Comput Secur 70:238–254

Vardhini KK, Sitamahalakshmi T (2017) Enhanced intrusion detection system using data reduction: an ant colony optimization approach. Int J Appl Eng Res 12(9):1844–1847

Vasilomanolakis E, Karuppayah S, Mühlhäuser M, Fischer M (2015) Taxonomy and survey of collaborative intrusion detection. ACM Comput Surv (CSUR) 47(4):55

Veeramachaneni K, Arnaldo I, Korrapati V, Bassias C, Li K (2016) AI 2: training a big data machine to defend. In: 2016 IEEE 2nd international conference on big data security on cloud (BigDataSecurity), IEEE international conference on high performance and smart computing (HPSC), and IEEE international conference on intelligent data and security (IDS). IEEE, pp 49–54

Verma P, Anwar S, Khan S, Mane SB (2018) Network intrusion detection using clustering and gradient boosting. 2018 9th international conference on computing, communication and networking technologies (ICCCNT). IEEE, pp 1–7

Vidal JM, Monge MAS, Monterrubio SMM (2020) Anomaly-based intrusion detection: adapting to present and forthcoming communication environments. In: Handbook of research on machine and deep learning applications for cyber security. IGI Global, pp 195–218

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

Vinh NX, Epps J, Bailey J (2010) Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. J Mach Learn Res 11(Oct):2837–2854

Wang AH (2010) Detecting spam bots in online social networking sites: a machine learning approach. In: IFIP annual conference on data and applications security and privacy. Springer, pp 335–342

Wang G, Hao J, Ma J, Huang L (2010) A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering. Expert Syst Appl 37(9):6225–6232

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

Wang J, Hong X, Ren Rr, Li Th (2009) A real-time intrusion detection system based on PSO-SVM. In: Proceedings. The 2009 international workshop on information security and application (IWISA 2009), Citeseer, p 319

Wang SS, Yan KQ, Wang SC, Liu CW (2011) An integrated intrusion detection system for cluster-based wireless sensor networks. Expert Syst Appl 38(12):15234–15243

Wang W, Sheng Y, Wang J, Zeng X, Ye X, Huang Y, Zhu M (2018) HAST-IDS: learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. IEEE Access 6:1792–1806

Wedde HF, Lehnhoff S, van Bonn B, Bay Z, Becker S, Böttcher S, Brunner C, Büscher A, Fürst T, Lazarescu AM, et al. (2007) Highly dynamic and adaptive traffic congestion avoidance in real-time inspired by honey bee behavior. In: Mobilität und Echtzeit, Springer, pp 21–31

Woland A, Santuka V, Harris M, Sanbower J (2018) Integrated security technologies and solutions-volume I: Cisco security solutions for advanced threat protection with next generation firewall, intrusion prevention, AMP, and content security. Cisco Press

Xie M, Hu J (2013) Evaluating host-based anomaly detection systems: a preliminary analysis of ADFA-LD. In: 2013 6th international congress on image and signal processing (CISP), vol 3. IEEE, pp 1711–1716

Xie M, Hu J, Slay J (2014) Evaluating host-based anomaly detection systems: application of the one-class SVM algorithm to ADFA-LD. In: 2014 11th international conference on fuzzy systems and knowledge discovery (FSKD). IEEE, pp 978–982

Xie Y, Feng D, Tan Z, Zhou J (2016) Unifying intrusion detection and forensic analysis via provenance awareness. Future Gener Comput Syst 61:26–36

Xin Y, Kong L, Liu Z, Chen Y, Li Y, Zhu H, Gao M, Hou H, Wang C (2018) Machine Learning and Deep Learning Methods for Cybersecurity. IEEE Access 6:35365–35381

Xu C, Zhang Q, Li J, Zhao X (2008) A bee swarm genetic algorithm for the optimization of DNA encoding. In: 3rd international conference on innovative computing information and control, 2008. ICICIC'08. IEEE, pp 35–35

Xu C, Shen J, Du X, Zhang F (2018) An intrusion detection system using a deep neural network with gated recurrent units. IEEE Access 6:48697–48707

Xu J, Han D, Li KC, Jiang H (2020) A k-means algorithm based on characteristics of density applied to network intrusion detection. Comput Sci Inf Syst 00:14–14

Yang Z, Algesheimer R, Tessone CJ (2016) A comparative analysis of community detection algorithms on artificial networks. Sci Rep 6:30750

Yassin W, Udzir NI, Muda Z, Sulaiman MN, et al. (2013) Anomaly-based intrusion detection through k-means clustering and Naives Bayes classification. In: Proceedings of 4th international conference on computing and informatics, ICOCI, vol 49, pp 298–303

Yihunie F, Abdelfattah E, Regmi A (2019) Applying machine learning to anomaly-based intrusion detection systems. In: 2019 IEEE Long Island systems, applications and technology conference (LISAT). IEEE, pp 1–5

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

Zhang J, Zulkernine M, Haque A (2008) Random-forests-based network intrusion detection systems. IEEE Trans Syst Man Cybern Part C Appl Rev 38(5):649–659

Zhang W, Yang Q, Geng Y (2009) A survey of anomaly detection methods in networks. In: International symposium on computer network and multimedia technology (2009), CNMT 2009. IEEE, pp 1–3

Zhou H (2018) Malware detection with neural network using combined features. In: China cyber security annual conference. Springer, pp 96–106

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