



Attack classification using feature selection techniques: a comparative study

Ankit Thakkar¹ · Ritika Lohiya¹

Received: 19 February 2020 / Accepted: 30 May 2020 / Published online: 12 June 2020
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract

The goal of securing a network is to protect the information flowing through the network and to ensure the security of intellectual as well as sensitive data for the underlying application. To accomplish this goal, security mechanism such as Intrusion Detection System (IDS) is used, that analyzes the network traffic and extract useful information for inspection. It identifies various patterns and signatures from the data and use them as features for attack detection and classification. Various Machine Learning (ML) techniques are used to design IDS for attack detection and classification. All the features captured from the network packets do not contribute in detecting or classifying attack. Therefore, the objective of our research work is to study the effect of various feature selection techniques on the performance of IDS. Feature selection techniques select relevant features and group them into subsets. This paper implements Chi-Square, Information Gain (IG), and Recursive Feature Elimination (RFE) feature selection techniques with ML classifiers namely Support Vector Machine, Naïve Bayes, Decision Tree Classifier, Random Forest Classifier, k-nearest neighbours, Logistic Regression, and Artificial Neural Networks. The methods are experimented on NSL-KDD dataset and comparative analysis of results is presented.

Keywords Intrusion detection · Feature selection · Machine learning · Feature extraction · Classification · NSL-KDD dataset

1 Introduction

Information and Communication Technology (ICT) enabled systems or devices such as Internet, wireless networks, smart phones, and other communication channels to manage data flowing through the network. The information flowing through the communication channel of the network might contain sensitive data that are subject to various attacks. The attacks can be internal or external to the network. The attacks are executed by intruders by exploiting system vulnerabilities. Advancement in diversity of attacks has resulted in unintelligible and undetected data breaches. For instance, Yahoo, internet service company, reported two data breaches in 2014 and 2016 affecting 500 million user accounts with a loss of 350 millions (Larson 2016). The attacks are targeted

with an aim to steal data by using sophisticated and intelligent algorithms. The number of local and enterprise network are increasing every year and therefore, attacker are using advanced algorithms to obstinate the target hardware, software, or network topology. The advancement in network attacks raised the need to develop intelligent Intrusion Detection System (IDS) that can proactively detect anomalies, attacks, or vulnerabilities in the network or systems and thereby, adhere to the security policies of the infrastructure.

An IDS can be categorized based on architecture as Host-based IDS and Network-based IDS, whereas based on detection methods it can be categorized as Anomaly-based IDS, Signature-based IDS, and Specification-based IDS (Rajput and Thakkar 2019). An IDS inspects the network behaviour using networking tools such as WireShark, Nmap, and many more (Mandal and Jadhav 2016). The networking tools captures network packets flowing through the devices configured in the network such as routers, switches, gateways, and sensors (Mandal and Jadhav 2016). The captured packets are analyzed to detect attacks and possible loopholes within and outside the network infrastructure. An IDS also examines system activities and network communication within

✉ Ritika Lohiya
18ftphde30@nirmauni.ac.in
Ankit Thakkar
ankit.thakkar@nirmauni.ac.in

¹ Institute of Technology, Nirma University, Ahmedabad, Gujarat, India

the network infrastructure to detect attacks and vulnerabilities within the network. The analysis of system activities is performed by inspecting various log files, files systems, and disk resources of the system. Based on the type of architecture and analysis strategy used, an IDS performs inspection of various network elements. For instance,

- A Network-based IDS inspects network packets, contents of the network packet, and network traffic flowing in and out of the network to detect anomaly.
- A Host-based IDS examines the system activities by inspecting various log files that may include sensor logs, system logs, software logs, and user data associated with local host system to detect system vulnerabilities and anomalies.
- Anomaly-based IDS analyzes network behaviour by applying heuristic techniques to detect anomalies. Anomaly-based IDS are capable of detecting novel attacks, however, such IDS result in high number of false positive rate (Sahani et al. 2018).
- Signature-based IDS derives and matches the signatures/patterns of attack with already available signature/pattern database. Signature-based IDS are also referred as misuse-based IDS and not capable enough to detect novel attack as well as variants of existing attacks (Sahani et al. 2018).
- Specification-based IDS performs anomaly detection based on defined specification and filtering rules for a given network environment.

Apart from these IDS detection mechanisms, stateful protocol analysis can also be performed to detect anomalies at various layers of communication model such as network layer, transport layer, and application layer. Predefined protocol rules and specifications are considered for anomaly detection. Though ML techniques have been applied to enhance the intelligence of IDS, however, there is still a scope of study to explore ML techniques with recent publicly available dataset (Sahani et al. 2018).

The most common challenges with existing solutions based on ML techniques are as follows.

- Considering the dataset being used and the attack categories, ML techniques produce high false alarm rate for wide range of attacks (Sharafaldin et al. 2018).
- Generalization of IDS model using ML technique can not be derived as research studies have used dataset with different dimensions or simulated dataset to measure the performance of IDS (Sharafaldin et al. 2018).
- Moreover, the dataset used for experimentation are either synthesized dataset or realistic dataset with limited network traffic. Therefore, such datasets are deprived of realistic scenario of today's massive network traffic,

communication speed, and dynamics (Sharafaldin et al. 2018).

These challenges form a key motive for our research work, that focuses on evaluating the efficacy of ML techniques by applying feature selection techniques to detect and classify attacks using NSL-KDD dataset.

The contribution of the research work is as follows.

- The study presents a comparative analysis of different ML techniques in terms of accuracy, precision, recall, and f -score for attack classification. The evaluation of adopted ML techniques is performed using NSL-KDD dataset.
- Our research work aims to incorporate feature engineering and analyze the performance of ML techniques in classifying the network traffic as normal or anomalous.
- Chi-Square, Information Gain (IG), and Recursive Feature Elimination (RFE) feature selection techniques are used for selecting significant features for classifying network traffic.
- Detailed description of the considered dataset is presented to exhibit the prominence of the dataset and show the details regarding the network traffic and attack categories.
- Our comparative analysis can provide insights to researchers to select appropriate techniques for building effective and efficient IDS.

1.1 Intrusion detection system

IDS is one of the important security mechanisms that protects network against attacks and malicious activities. An IDS can be deployed in integration with other security mechanism such as firewall, access control, to name a few. These security mechanisms along with IDS analyze the data and secure the network against the anomalies. An IDS deployed in the network examines the network activities and subsequently gain knowledge about the potential threats and vulnerabilities that might cause harm to the system and the network environment.

An IDS can be described as the system that monitors network flow for identifying malicious traffic and anomalies. It also ensures that the security principles of the resources allocated in network environment and the systems are maintained (Sahani et al. 2018). Intrusion detection process consists of following steps: monitoring network traffic, collecting data from network packets, processing data for analysis, identifying signatures and deriving patterns, comparing patterns with the stored signatures and/or patterns, and generating alarm if any unusual behaviour or pattern is recognized (Sahani et al. 2018). The collected data is compared with available detection rules for intrusion detection.

The detection and response mechanism of an IDS is shown in Fig. 1 (Denning 1987). The detection models cannot be instantly constructed from the data available especially when the main task is to classify the behaviour of network activities (Denning 1987). This is because, massive amount of network traffic, unequal distribution of data, lack of knowledge to recognize new attacks, and lack of stability towards constantly changing network flow. Moreover, IDS is not capable enough in managing alarms that are generated in large numbers (Sahani et al. 2018). This results in increased computational time and low detection rate. Therefore, there is a need to study security risks of network pertaining to IDS. This requires processing the data before building the intrusion detection model. The data is processed and refined through feature engineering techniques that filters the data and may increase the efficiency of the IDS model.

Feature engineering can be described as the task to select or extract features from the feature set (Thakkar and Lohiya 2020b). Feature Selection (FS) is the process of selecting the subset of relevant features from the given dataset while feature extraction is the process of generating new features from already available features. Feature engineering helps to remove features that are irrelevant or redundant. This is because irrelevant features do not contribute in classifying an attack and redundant features have similar characteristics and thus, can be removed. Hence, building IDS with FS might represent collected network data in a better way (Olusola et al. 2010).

Many supervised and unsupervised learning methods have been implemented for building a sophisticated and efficient IDS that detects and classifies attacks. It is important to note that the processing time of ML techniques is increased with the rise in the dimension of data. The complexity of ML techniques can be described based on the parameters, model, and accuracy gained by the classifier while measuring the performance (Olusola et al. 2010). In fact, there is a noticeable change in the performance

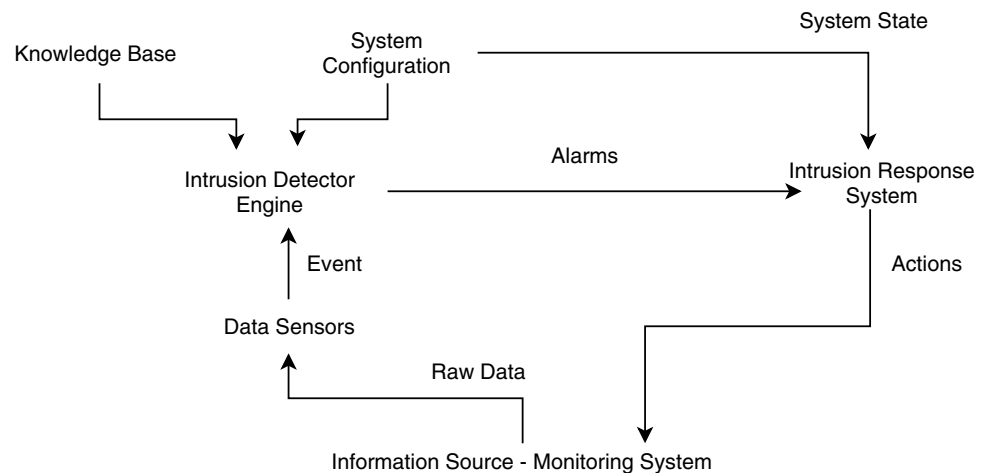
of the algorithm designed for IDS when the size of the dataset is reduced using the FS methods (Olusola et al. 2010). Hence, it is important for the FS methods to extract important features for learning process for detecting the attacks with high accuracy. FS needs to understand the role of each feature for detecting a particular type of attack so as to identify the similar kind of unknown attacks.

There are certain merits of applying ML-based techniques for detecting attacks compared to conventional IDS.

- It is easy to compromise a conventional IDS that identifies anomalies based on signature database. Whereas, ML-based IDS is developed using techniques that are capable of learning traffic behaviour through which even a slight change in the attack pattern can be identified.
- ML-based IDS require low to moderate CPU load, as these systems do not examine all the signatures of the database unlike signature-based IDS (Almseidin et al. 2017).
- ML-based IDS can extract the complex attributes of the network behaviour using which detection accuracy can be enhanced.
- An updation in the signature database of an IDS is required due to evolution in the type of attacks. Whereas, clustering and outlier techniques of ML do not require any such update.

In this paper, we have mainly focused on the use of FS methods and their effect on the performance of ML-based IDS. For performance evaluation of various classifiers, NSL-KDD dataset is used. ML techniques are used to detect and classify attacks using features of the dataset. The experimental model implements FS on the basis of a classification criteria defined for the underlying FS algorithm (Li et al. 2017). For instance, information gain based FS method selects features based on gain ratio for each feature.

Fig. 1 Intrusion detection and response system (Denning 1987)



The paper has been divided into following sections: Sect. 2 discusses the background and related work; Sect. 3 describes the FS methods and its classification. Section 4 lists out various classification techniques that have been implemented. Section 5 presents the dataset description. Sections 6 and 7 describes experimental methodology and result analysis, respectively. We conclude our work with potential future work directions in Sect. 8.

2 Background and related work

The model of an IDS is designed and explained in (Denning 1987). An IDS is designed to address the problems of intrusion occurring internal and external to the network. The various application areas where IDS plays a vital role are illustrated in Table 1 (Agarwal and Hussain 2018).

The intrusion detection model is comprised of the component named knowledge base that is used as basis to perform statistical operations on data gathered from sensors planted in the environment. The knowledge base is comprised of user profile information, configuration details of host and target systems, and attack patterns and signatures. However, a rule-based approach is also proposed that detects intrusions and abnormal behaviour by flagging patterns recognized by studying activities of legitimate users (Ektefa et al. 2010). It is based on prior information of network and system that is considered for applying rule based approach (Smaha 1988). The idea of IDS became conventional with the research conducted by the MIT Lincoln laboratory to create a dataset for evaluating performance of IDS in 1998 and 1999 (McHugh 2000). The dataset recorded many limitations in terms of instances and attributes collected that is described

in Nehinbe (2011) and Thakkar and Lohiya (2020a). Thus, these datasets could not demonstrate instances of real network scenarios and contained attack categories with narrow scope, this lead to more research in developing new datasets with all requirements of assessing a real time attacks and situations. Apart from, classifying instances as benign or attack, there has been a lot of effort towards executing multi-class classification on these dataset by considering feature relevance (Olusola et al. 2010).

Features in dataset can be either discreet or categorical as they are extracted from packet header and packet payload. Selecting features from the set implies addressing features having high relevance. For instance, in (Kloft et al. 2008) a method is proposed to automatically select the features by using optimal mixture coefficients for different group of features. Here, the proposed method has grouped the features in six subsets out of which three sets consists of sequential data streams of the HTTP request packet while the other three represent the token attribute sequence of the HTTP request packet. The tokens are keywords that are associated with the HTTP request packet and attributes are byte sequences of these keywords that are extracted. The experimental results are presented by comparing the accuracy of each subset of features individually and with subset of features obtained by the proposed method. The proposed method concluded that various features obtained from analyzing network traffic can be grouped automatically to form feature subsets for evaluation (Kloft et al. 2008).

Principal Component Analysis (PCA) has been used for feature reduction with Support Vector Machine (SVM) in Heba et al. (2010). Here, the method opted for reducing the dimension of the features used to reduce computational complexity of the model. The implementation has resulted

Table 1 Application areas of intrusion detection system (IDS) (Agarwal and Hussain 2018)

Application area	Role of IDS
Securing communication protocol	IDS analyzes the network traffic generated by various communication protocols such as HTTP and HTTPS IDS investigates various parameters of web request following through HTTP and HTTPS such as port number, source address, destination address, to name a few
Scrutinizing web applications	IDS investigates the web applications to provide real-time intrusion detection and prevention and data source for post intrusion analysis IDS checks for various parameters such as allowed byte range, regular expressions, and allowed whitelist of variables in web request for underlying web applications
Securing user information	IDS tracks and monitors user's session and keeps the record of session details and authorization policies of the users
Heterogeneity	Different programming languages are used by various web platforms for establishing network communication between source and destination. However, IDS can be built for general as well as specific to a particular programming platform for detecting intrusions
Securing system logs	IDS analyzes the system logs for identifying the presence of any vulnerabilities in the system as well as presence of any malicious activities
Identification of attacks	With evolution in network traffic, there has been a dynamic change in the nature and type of network attacks. IDS aims at handling the continuous change in the web content and the dynamicity of web request for identifying malicious content present in the network

in improved performance of IDS with decrease in computational time. Whereas, in Nguyen et al. (2010) the Correlation based Feature Selection (CFS) method has been optimized by representing the CFS method as polynomial based 0–1 fractional problem. The optimized CFS method has a limitation that it is linear in nature and consists of set of features with many constraints and variables. This linear representation of features is addressed by branch and bound algorithm to derive set of features. The proposed method was compared with different variants of CFS namely best first CFS and genetic algorithm based CFS method. Decision Tree (DT) and BayesNet classification method were used with KDD CUP 99 dataset. The result analysis of proposed algorithm outreached both the algorithms in terms of accuracy.

The enhanced SVM decision function is used for FS in Zaman and Karray (2009). Here, the features are selected based on two criteria: feature weight which is obtained through SVM decision function and establishing correlation between features either by forward selection ranking or backward elimination ranking. The algorithm was tested in terms of classification accuracy, computational time, and detection time (Zaman and Karray 2009). In (Zainal et al. 2009), filter-based selection methods namely, Chi-Square, Information Gain (IG), and Relief-F are compared for DT classifier with KDD CUP 99 dataset. The experimental results were presented using top 5, 10, 15, and 20 features from total 41 features of the dataset. The results clearly showed the effectiveness of IDS using FS against all features. Out of the three FS methods, IG outperformed in terms of classification accuracy.

Intrusion detection dataset consists of high dimension data that may pose problems of ambiguity, uncertainty, and redundancy in collected data instances. These problems are addressed in Chou et al. (2008) using symmetric uncertainty based FS methods named CFS and fast CFS. These methods are combined with Naïve Bayes (NB) and C4.5 for selecting features. For detecting attack type three variants of k-NN classifier are used: k-NN, fuzzy k-NN and evidence-theoretic k-NN. The proposed method called as fuzzy belief k-NN and is implemented with FS methods to compare its performance with other classifiers. The experimental results showed a considerable increase in detection rate in comparison with the other classifiers.

Gradual feature removal method is implemented in Mousavi et al. (2019). Here, in the proposed approach 16 features are selected and ensemble of decision trees is used to build an efficient IDS to classify the network traffic as normal or attack. The experiments are performed using KDD CUP 99 dataset. The ensemble of decision trees with feature selection achieved improved accuracy and value of Matthews Correlation Coefficient (MCC) 0.91 is obtained. An empirical study of ML techniques is presented in Meira et al. (2019), where IDS is built and evaluated using two

public datasets namely, NSL-KDD and ISCX. The study presented in Meira et al. (2019), aimed at understanding the working of ML techniques and how these techniques can be used to build an efficient IDS model. The results showed acceptable performance of ML techniques for intrusion detection and classification. Best features for intrusion detection are selected using random harmony search in Mayurathan et al. (2019). The proposed model is built for cloud environment to detect DoS attack. Here, in the proposed model Restricted Boltzmann Machine (RBM) is used for classification and experiments are performed using KDD CUP 99 dataset. The results show that the proposed model achieves improved performance by incorporating feature selection for attack detection. A host-based IDS is proposed in Besharati et al. (2019), where Logistic Regression (LR) is used as feature selection with combination of ML techniques namely, Artificial Neural Network (ANN), DT, and Linear Discriminant Analysis (LDA) with bagging technique. The experiments are performed using NSL-KDD dataset and executed on CloudSim software. The results showed accuracy of 97.51% for classifying the network traffic as attack or normal.

A new FS method is proposed in Mukherjee and Sharma (2012) based on vitality of features by considering the performance metrics such as accuracy, true positive rate, and false positive rate. Features are removed one at a time by “leave-one-out” strategy using sequential search. The Feature Vitality based Reduction Method (FVBRM) (Mukherjee and Sharma 2012) is used with NB classifier and experimental results are compared with FS methods namely CFS, IG, and gain ratio. The result analysis showed increase in accuracy but recorded high computational time. The feature goodness factor can also be utilized to select features. This can be demonstrated using mutual information based FS method or linear CFS (Amiri et al. 2011). The experimental results showed that the linear CFS selection works better when there is a linear relationship between attributes of the dataset where as mutual information based method is capable of building arbitrary relationship between the attributes by having maximum relevance and minimum redundancy (Amiri et al. 2011). Thus, FS has also been implemented using various other methods such as IG, attribute ratio, Chi-square, and RFE method using various ML classifiers such as NB, DT, k-NN, and SVM.

To address the issues of high dimensionality of data and feature relevance in IDS we have implemented different FS methods such as Chi-square, IG, and RFE for NSL-KDD dataset. These FS techniques are implemented with ML classifiers namely, NB, DT, Random Forest (RF), k-NN, LR, SVM, and ANN. The dataset consist of 41 features which were reduced using feature relevance capability of the FS algorithms and further given to the classifier for attack detection and classification.

3 Feature selection

For detecting intrusions, data is collected from the network packets. Therefore, it is a burdensome task, to manually classify massive amount of network data captured by the system. Apart from collecting the network data, analyzing data is also a challenging task as data consists of large amount of behavioural patterns and features. To secure the network against intrusions, real-time intrusion detection is required that can be achieved by finding significant features from available dataset. The reduced set of features can efficiently increase the detection rate of intrusions (Li et al. 2017).

Feature selection can be performed in many ways. For instance, filtering data that does not contribute in the detection process for classifying the attacks, grouping data into clusters on the basis of their similarity to identify the hidden patterns from the data for classification, and removing the irrelevant features from the feature set by applying FS algorithms.

Selecting the features is the process of identifying subset of features from the available dataset. This is achieved by evaluating the features on the basis of some criteria such as distance measures, dependency measures, information measures, and consistency measures (Dash and Liu 1997). Feature selection minimizes the size of the dataset that is further given as an input to the classification algorithm. Selecting features enhances the understand-ability and visualization of learning algorithm for intrusion detection. In general, FS is a process that removes unwanted noise, redundancy, and irrelevant features from dataset. This section describes basic overview of FS, types of FS methods, and FS process.

3.1 Overview of feature selection

The basic task of FS algorithm is to identify the features or group of features on the basis of their relevance from the underlying dataset. Pre-processing the data using FS algorithm enhances the efficiency of the classifier as well as refines the data for better analysis. In fact, the computational time for training the model can also be reduced. In study, (Olusola et al. 2010; Jović et al. 2015) it has been shown

that training the classifier with entire feature set demeans the performance of the model. Any dataset, consists of three types of features, namely, relevant features that reveals the maximum information about the data, irrelevant features that holds no information that can be used for training the model, and redundant features that gives information which is already given by the other features in the dataset and hence, do not contribute in predicting data patterns. The feature set consists of features that can be correlated with others features of the target class. These features are called as class-correlated features. Hence, eliminating such irrelevant, redundant, and class-correlated features enhances the accuracy of the IDS model.

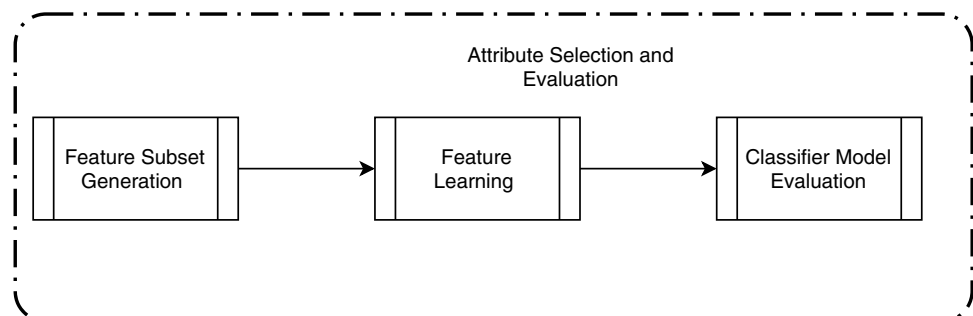
3.2 Feature selection techniques

The FS techniques are categorized into three categories namely wrapper, filter, and embedded. The wrapper based FS method is dependent on the classification algorithm used for the classification as shown in Fig. 2. The classification algorithm is used as the black box for selecting features from the feature set (Kumari and Swarnkar 2011). The subset of features are generated using FS techniques and thereafter, evaluated based on some criteria or evaluation function (Kumari and Swarnkar 2011).

The wrapper based FS can therefore be considered to have two parts named as search and evaluation (Balasaraswathi et al. 2017). The search process deals with parameter initialization that are used for evaluation of features using evaluation function. As the wrapper based FS algorithm is dependent on the classification algorithm, it interacts with the classifier to derive the importance of features for feature selection. The wrapper based FS method is slower in comparison to filter and embedded based FS techniques. It consists of forward selection and backward elimination for searching the features.

The forward selection strategy initializes an empty set of features and iteratively evaluates features one by one. For every step, the feature that gets the maximum value of the evaluation function compared to the available set is included (Balasaraswathi et al. 2017). The search process for selecting the features stops once no improvement

Fig. 2 Wrapper-based feature selection



in the evaluation function is found. While the backward elimination method initiate the feature selection process with the entire features available in the dataset and removes the features one by one in each iteration, if the elimination of that particular feature improves the performance. The search process terminates if elimination of the feature decreases the evaluation function value. (Balasaraswathi et al. 2017).

The filter based FS method eliminates features that does not contribute significantly in analysis of the data. The schematic of filter-based method is shown in Fig. 3. It is not dependant on the classifier for evaluating the features (Thakkar and Lohiya 2020b). The features are selected based on evaluation criteria described in Dash and Liu (1997). This method forms feature sets with a higher number of features and even entire feature set is selected sometimes Thakkar and Lohiya 2020b). Hence, a selection criteria need to be set for choosing subset of features.

The embedded feature selection method is hybridization of wrapper and filter based selection approaches as shown in Fig. 4. It implicitly or explicitly uses FS technique to improve the performance of the classifier. Some of the examples of this method are DT and NB classifier. Table 2 lists the advantages and disadvantages of the filter, wrapper, and embedded FS methods.

3.3 Feature selection process

To select features in FS process, initially entire feature set is considered for classification. The features are then selected by applying the FS methods. The selected features are used along with classification algorithm for evaluation of the model. The basic steps in the FS process are given as:

- Generating the subset of features
- Evaluation of the generated feature set
- Termination criterion
- Validating the results obtained for the given subset of features

Figure 5 (Kumari and Swarnkar 2011) shows the basic steps of FS technique.

3.4 Feature selection techniques used for experimentation

The main aim of our study is to demonstrate that selecting the right set of features for classification can help in enhancing the performance of ML techniques. The advantage of incorporating feature selection is twofold: the feature selection techniques can help to mitigate the curse of dimensionality with large datasets and computing importance of features can help in data interpretation. The feature selection techniques used in our study are Chi-Square, IG, and RFE

Fig. 3 Filter-based feature selection

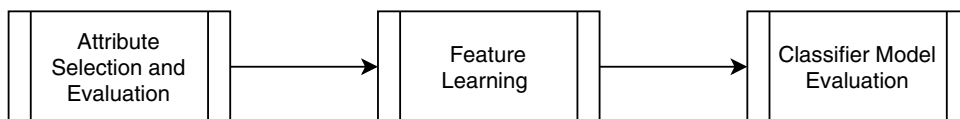


Fig. 4 Embedded feature selection

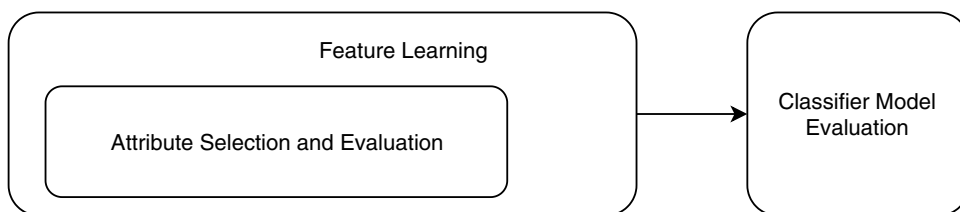


Table 2 Feature selection methods

Feature selection method	Advantages	Disadvantages
Filter based	Interaction with the classifier Computationally cost effective Good generalization ability	It is not dependent on the classifier
Wrapper based	Interaction with the classifier Computationally cost effective It derives the feature dependencies	It is dependent on the classifier used.
Embedded	Interacts with the classifier algorithm It derives the feature dependencies	It is costly in terms of computation It is dependent on the classifier used

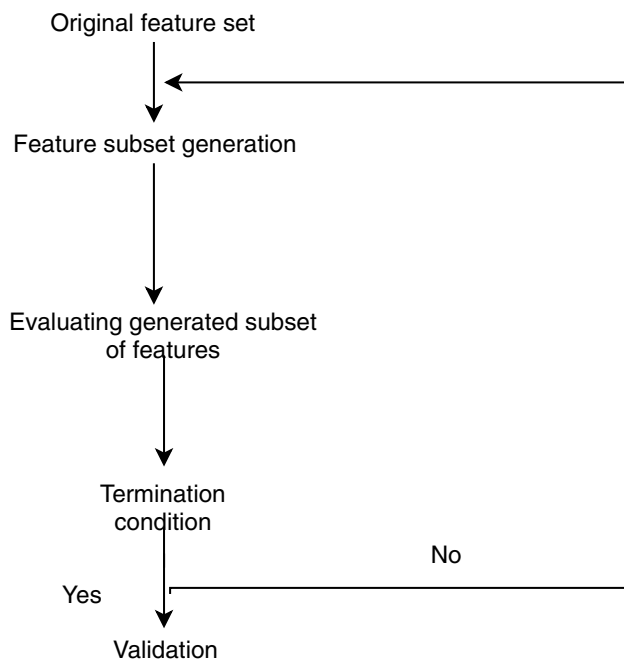


Fig. 5 Steps of feature selection process (Kumari and Swarnkar 2011)

and are explained in brief in the following sections. We have implemented the feature selection techniques by considering the functions and parameters defined in scikit-learn library (Hackeling 2017).

3.4.1 Chi-Square feature selection

The Chi-Square feature selection evaluate the independence of events for a given set of data. In Chi-Square feature selection technique, the independence of two events namely occurrence of feature and occurrence of class is evaluated. The chi-square score for each feature is calculated using Eq. 1 (Li et al. 2017).

$$\chi^2 = \frac{(\text{Observed Frequency} - \text{Expected Frequency})^2}{\text{Expected Frequency}} \quad (1)$$

For feature selection, χ^2 score of each feature and target is calculated and features with optimal χ^2 scores are selected. The intuition behind calculating χ^2 score is that, if a feature has low χ^2 score, then it is independent of target class which implies that it is uninformative for classifying data samples.

3.4.2 Information gain (IG)

Information gain selects features based on the entropy value of each feature of the dataset. In IG based feature selection technique, each feature is considered individually and its gain value is measured that reveals the importance of

feature with respect to the target variable. Features of the dataset are ranked based on their information gain value. A feature with low information gain value implies that it does not have much effect on classification of data. Therefore, features with low information gain value can be ignored without much affecting the performance of the classification model. The information gain value for class C and input feature X can be calculated using equation 2 (Li et al. 2017).

$$IG(C, X) = H(C) - H(C|X) \quad (2)$$

Here, $H(C)$ is the entropy of class label C and $H(C|X)$ is the probability of class C for a given feature X .

3.4.3 Recursive feature elimination (RFE)

RFE technique is a cross validated feature selection technique that recursively selects subset of features based on the feature ranking. RFE aims at removing the dependency and collinearity among features. A recursive feature elimination process recursively builds the model by removing the features and building model with the remaining features until all the features in the dataset are exhausted. The iterative process considered by RFE can be given as follows.

- Building the classification model by considering sub optimal feature set.
- Computing the feature importance for ranking features of the dataset.
- Removing the features with lowest rank based on feature importance.

4 Classification techniques

For a given classification problem, the main task is to identify a class label for each instance of the test dataset to minimize classification error. Classification techniques are used to build models that can easily recognize classes precisely for a given dataset (Allahyari et al. 2017). Generally, classification is carried out in two phases namely training phase and testing phase. In training phase a classification model is built by learning from the training dataset. Thereafter, a test dataset is used in the testing phase to measure the performance of the underlying classifier. Thus, a classification model is used to assign class labels to the unknown data which is given input to the model (Allahyari et al. 2017). The ML techniques used for experimentation are DT, RF, NB, LR, k-NN, SVM, and ANN. These techniques are adopted as they are widely used for developing efficient and effective IDS (Biswas 2018; Aljawarneh et al. 2018). Each ML technique used for attack classification is described in this section along with their advantages and disadvantages.

4.1 Decision tree (DT) classifier

A DT classifier exhibits a structure which resembles to a tree with nodes and edges. Each node in the tree represents a category of the problem to be classified and each edge denotes the decision taken on the basis of data evaluated. DT classifier can be regarded as the predictive model of machine learning which shows mapping between the features of the dataset and its values (Song and Ying 2015). Every split in DT denotes possible value to be considered for a given category. The nodes of the tree are identified based on the entropy calculations of attributes of the dataset. The attribute with highest entropy value is regarded as the root node. Commonly used DT models are ID3, C4.5, and CART (Breiman 2017).

4.1.1 Advantages

Interpretation of DT classifier is simple and can be easily understood with a brief explanation. Inferences can be derived based on the different probability estimation and costs (Breiman 2017). Based on these inferences precise outputs can be obtained. It can be easily collaborated with other classification model for obtaining accurate results. It exhibits better performance for the methods already known to the model (Song and Ying 2015). For instance, if the model is aware about the available intrusion methods and scenarios it gives more accurate results in classifying the attacks.

4.1.2 Disadvantages

DT classifier is not adaptive to small changes in the data. A small change in data fed to the classifier can result in highly unstable decision tree structure (Phutane and Pathan 2015). While dealing with similar kind of data they have relatively lower accuracy. It is very complex in deriving the nodes if data is linked or uncertain. It is not suitable for the problems having very less information about the data (Phutane and Pathan 2015). Like in IDS, DT cannot be implemented for the attacks which are not known a priori.

4.2 Random forest (RF) classifier

RF classifier is mainly regression trees with unpurged classification (Breiman 2017). They construct a group of decision trees by applying bagging technique to select appropriate parameter for constructing each of the decision trees. The features are selected randomly from the training dataset. For detecting different attack types from the intrusion detection dataset, the features are randomly selected based on the importance of each feature. RF constructs many decision trees based on the instances of class and root of random forest tree is selected by suitable voting from each class of

the tree constructed. The error rate of the RF classifier is minimized by using bootstrapping. The result optimization in RF is directly dependent on the correlation and similarity between the two trees (Breiman 2017). If the correlation is high then the error rate also increases whereas, if the strength of the forest is increased then the error rate is decreased.

4.2.1 Advantages

RF can be used even with the large datasets having numerous features. It calculates the importance of each feature. It does not overfit the data and can even handle unbalanced datasets (Breiman 2017).

4.2.2 Disadvantages

It cannot be interpreted easily. As it generates many trees, real-time classification prediction becomes difficult. It is a slower model if large number of trees are generated (Breiman 2017).

4.3 *k*-Nearest neighbour (*k*-NN) classifier

k-nearest neighbour is a supervised classification technique that classifies the data based on the class of its nearest neighbour. The algorithm computes the classes of data based on the value of *k* (Maillo et al. 2017). It predicts the class of data sample based on consistency and distance with the closest neighbour. Distance metrics such as Euclidean distance and Manhattan distance are considered for evaluating the data points distance with the nearest neighbours (Chomboon et al. 2015). All the data points are stored in the memory and therefore, this method is often referred to as a memory based technique. To enhance the performance of the algorithm weights can be assigned to the training points based on their distances from the data points (Chomboon et al. 2015). Computational complexity and memory requirements are the two major concerns for the classification technique that can be handled by reducing the size of the dataset or eliminating the data points that do not add to the repeated patterns.

4.3.1 Advantages

It is a cost effective algorithm as no cost is required to learn the data (Chomboon et al. 2015). It does not assume anything about the properties of the data provided for training (Chomboon et al. 2015). It can work well even with the large datasets by assuming local approximation using simple methods.

4.3.2 Disadvantages

Interpretation of the model is very complex as there is no description of the data provided for training. It is an expensive learning method as it is a tedious task to find the k -nearest neighbour if very large amount of data is available and also the data points are stored in the memory for computation (Chomboon et al. 2015). It might require a very large dataset. The performance of the algorithm is completely dependent upon the attributes selected for computation and hence, it results in curse of dimensionality for the dataset considered (Witten et al. 2016).

4.4 Logistic regression (LR) classifier

LR classifies data based on an algebraic approach (Harrell Jr 2015). The LR model is built based on the probabilities of instances of a class. These probabilities are derived using a logistic function applied to each class of the dataset. Logistic function is derived from linear regression where the probability of a particular data point in the class is represented by a linear function. Logistic regression however uses the logistic probability function and therefore, a logistic regression model can also be represented by a logit function (Harrell Jr 2015). It is often referred to as the logit function, and classification is called as the log-linear classification (Harrell Jr 2015).

4.4.1 Advantages

LR Models are efficient as they do not need any additional resources for computation. These models can use features without applying scaling or parameter tuning, and therefore, they are highly interpretable (Mansournia et al. 2017). It is easy to train and works as a good baseline that can be utilized to evaluate the performance of complex problem.

4.4.2 Disadvantages

LR does not work well with non-linear problem. It highly depends upon how the data is presented. It can work well with dataset having categorical features. It is prone to overfit the data. Moreover, identifying the important variables is a major task (Mansournia et al. 2017).

4.5 Naïve Bayes (NB) classifier

NB classifier is the probabilistic classifiers based on Bayes theorem (Puga et al. 2015) wherein the probability of occurrence of one event is independent from the other (Fadlil et al. 2017). The algorithm is based on the fact that the presence of any attribute is not dependent on any other attribute of the class. The probabilities of all the features of the class

independently contribute to classify the unknown data into defined classes. The algorithm works by applying the Bayes theorem to find the posterior probability of the class given the probability of predictor variable $P(c|x)$. The posterior probability can be calculated from the prior probability of the class $P(c)$, prior probability of the predictor variable of the class $P(x)$, and the maximum likelihood of predictor given class $P(x|c)$. The class c is independent of all the probabilities of other variables present in the class (Witten et al. 2016).

4.5.1 Advantages

NB algorithm is highly scalable and quick in classification. It can be applied to both binary and multi-class classification problems. It also has many variants like GaussianNB, MultinomialNB, and BernoulliNB (Fadlil et al. 2017). It is very simple to implement as it depends on simple counts of class conditional independence. It works well with small dataset.

4.5.2 Disadvantages

As NB algorithm depends on assumption that the features are independent of each other so it cannot derive any relationship between the features of the class. With large datasets the algorithm implementation becomes more complex (Witten et al. 2016).

4.6 Support Vector machine (SVM) classifier

SVM has been applied to variety of applications such as text categorization, image processing, attack classification to name a few (Suthaharan 2016). It is used for both classification as well as regression problems. It is a statistical learning method that minimizes the risk by defining the support vectors which best segregates the classes by passing a hyperplane through these points. SVM algorithm decreases the generalization error by maximizing the margin between the hyperplane and the support vectors (Meyer and Wien 2015). SVM performs better with linearly separable data and also handles non-linear data by transforming the data using kernel function to high dimensional feature space. Such functions are often referred to as kernel tricks. Based on this learning, the dataset can be separated into two parts as working set and set of free variables. Initially SVM was designed to address binary classification problems but it can also be used for multi-class classification problem by decomposing the multi-class problems into several two class problems that can be addressed further by several SVMs (Suthaharan 2016).

4.6.1 Advantages

SVMs does not depend upon the probability of the attributes or classes. It directly gives the appropriate classes present in the dataset. It can deal with structured as well as unstructured dataset. The most important part of SVM is the kernel function that allows the SVM to deal with the non-separable data easily. It exhibits low risk of overfitting of data and can handle high dimensional data easily (Dogan et al. 2016).

4.6.2 Disadvantages

The computation time for training the data is very high. Understanding the model is difficult if small calibrations are made to the model. Applying the right “kernel function” is a tedious job. Data analysis of SVM is based on the convex quadratic programming which makes it computationally expensive (Dogan et al. 2016).

4.7 Artificial neural network (ANN) classifier

Neural Networks are classification algorithms inspired from the understanding and emulating human brain and have been applied to diversified applications (van Gerven and Bohte 2018). They are structured in the form of interconnected nodes with an input layer, an output layer and one or more hidden layers. The model has various layers where input to each node is produced by applying some non-linear function on the data. On the basis of this structure neural networks have been applied to application for deriving decisions from the patterns formed (van Gerven and Bohte 2018). For instance, neural networks have been used in an IDS to classify different attack categories based on the features provided to the model for training (Da Silva et al. 2017).

The working of ANNs resembles to that of the neurons in human brain. Analogous to neurons, ANN establishes the connections between its components (Da Silva et al. 2017). Every component is built and examined by the system and input signals having arbitrary weights is assigned to each node in the network. Every input signal in one layer produces an output signal that acts as an input for other nodes in the next layer. The architecture of ANN is in the form of strongly connected layers where the input layer receives the input and the output layer gives the final output. An ANN architecture consists of one or more hidden layers.

ANNs are typically initialized with random weights for all the nodes present in the layers. This implies that each of the neuron is initialized to address a specific problem for which they are configured (van Gerven and Bohte 2018). For instance, a back propagation algorithm is trained for performing a specific task. While training the neurons in the back propagation ANN, the final output obtained is compared with the actual output by analysing the pattern. If the

final output obtained matches the expected output then the weights on the neurons are reinforced whereas, if the output do not match with expected output then the error is calculated between the layers and weights are adjusted to obtain the actual output (Da Silva et al. 2017).

4.7.1 Advantages

ANN have the capability of learning and analyzing the non-linear and complex connection between the input and output nodes. After having studied the connections of the input nodes, ANN can generalize and derive the unknown connections or patterns between the nodes thus making the model more flexible towards the unknown data (van Gerven and Bohte 2018). In contradiction with the other state of art approaches, ANN does not define any constraints on the input variables. ANN also has the capability of dealing with heteroskedasticity which means data with high volatility and high variance (Russell and Norvig 2016). The ability to learn hidden patterns proves to be useful in applications such time series forecasting that has high volatility in the data.

4.7.2 Disadvantages

The ANNs requires very long training time to learn the data and derive patterns. This is mainly a problem when the training is being carried out on a CPU in place of a specialized GPU machine (van Gerven and Bohte 2018). ANN requires a large amount of data to learn if the architecture consists of many hidden layers. This is because, every layer in ANN consists of nodes with randomized weights and connections. To achieve better performance with ANN, the architecture of ANN should be fine-tuned. The performance of ANN depends on attributes such as number of layers considered, number of nodes in each layer, and activation functions (Russell and Norvig 2016).

5 Dataset Description

The NSL-KDD dataset (Tavallaee et al. 2009) is developed to address the shortcomings of KDD CUP 99 dataset. It is the refined version of KDD CUP 99 dataset and is still being used as the benchmark dataset by the researchers for comparative analysis of intrusion detection techniques. The data samples in the training set and test set of NSL-KDD dataset are reduced compared to KDD CUP 99 dataset. The reduced size of the data set enhances the execution of experiments considering the entire set of data samples.

The KDD CUP 99 dataset is developed by considering the data captured in DARPA dataset, and therefore, both the datasets were condemned because of the presence of

synthetic traffic scenarios. The problems scrutinized in KDD CUP 99 dataset are listed as follows.

- While developing the dataset the experiments were conducted to synthesize background as well as the attacks data. The data collected did not undergo analytical and experimental validation to measure the false alarm characteristics of the data (Thakkar and Lohiya 2020a). The dataset was also not analogous to the real network scenarios (McHugh 2000).
- Tcpcdump network tool was used to collect the network traffic. This tool under-performs in heavy traffic scenarios and is easily overloaded and starts dropping packets in heavy traffic load. However, there was no scrutiny considered for performing the analysis and checking the possibility of dropped and lost packets (Tavallae et al. 2009).
- The attack categories are not properly defined. For instance, Probe is not particularly an attack type unless the number of iterations surpass a specific threshold value (McHugh 2000).

The NSL-KDD dataset consists of 41 features that are used for empirical and statistical analysis of the techniques used for intrusion detection. The NSL-KDD dataset has been developed by removing the duplicate and redundant records from the KDD CUP 99 dataset. We have used 126620 instances of data samples as training set and 22850 instances of data samples as test set. NSL-KDD dataset consists of 41 features. These features are classified into four categories such as basic, content, traffic, and host (Thakkar and Lohiya 2020a). The details regarding NSL-KDD dataset used for experimentation is presented in Table 3.

The dataset has five classes that are named as DoS, Probe, R2L, U2R, and Normal.

- Denial of Service Attack (DoS): It is an attack performed by the intruder by using the memory and keeping the sys-

tem resources busy so that these cannot handle requests of the authenticated users.

- Probing Attack: It is a response attack that is purposely executed for gathering information about the network systems to dodge its security controls.
- Remote to Local Attack (R2L): It is a type of attack by which the intruder gains local access of a machine in the network and has the capability of sending network packets to the other users in the network. It is performed by exploiting the vulnerability of the system and gaining local access.
- User to Root Attack (U2R): It is a type of exploit that is executed by gaining access to a user account. The access is gained by performing dictionary attack, social engineering, or sniffing the password (Richhariya et al. 2017). The execution of these exploits gives root access to the intruder.

The NSL-KDD data set has the following advantages over the KDD CUP 99 data set:

- NSL-KDD dataset does not consist of any redundant instances in the training set. This ensures non-biased classification of the records (Tavallae et al. 2009).
- The dataset does not contain any duplicate records in the test sets. This ensures non-biased detection rates of the frequent records by the classifiers (Tavallae et al. 2009).
- The data samples present at each level group is inversely proportional to the percentage of samples present in the KDD CUP 99 dataset. This results in variation in the classification rates of different algorithms, which implies that the evaluation of the different algorithms can be performed efficiently and accurately (Almseidin et al. 2017).
- The number of instances in the training set and the test set makes it easy to perform the experiments by considering the entire dataset without selecting the random number of instances (Tavallae et al. 2009). This can permit to perform comparative analysis of different research works.

Table 3 Dataset Description

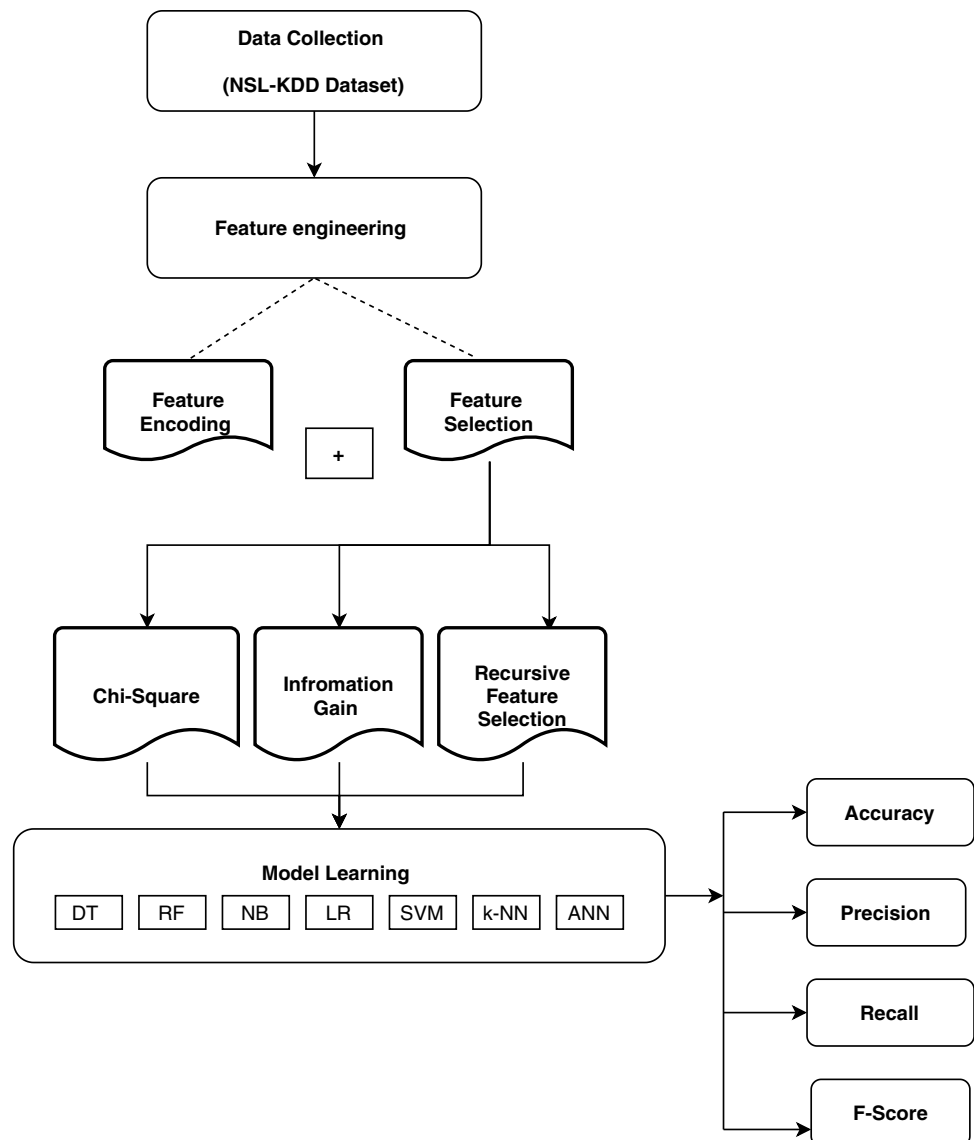
Criteria	Description
Dataset	NSL-KDD
Number of records	1,49,470
Number of records in training set	1,26,620
Number of records in test set	22, 850
Number of attack categories	4 (DoS, Probe, R2L, U2R)
Number of network features	41
Type of network features	Basic features, traffic feature, host features, and content features

6 Experimental methodology

The experimental methodology is divided into four steps as shown in the Fig. 6.

- **Pre-processing** Data pre-processing techniques are applied for better understanding and visualization of attributes of dataset used for experiment. In our study, we have used NSL-KDD dataset for performance evaluation of ML techniques. The NSL-KDD dataset con-

Fig. 6 Flow of experimentation



sists of binary, numeric, and categorical features. The dataset consists of three categorical features namely protocol type, type of service, and flag. These features are encoded using one-hot encoding. All the features are standardized to have a unit variance using the standard scalar technique.

- **Feature Engineering** After applying pre-processing techniques on features of the dataset, feature engineering is performed. In feature engineering, various feature selection technique such as IG, Chi-Square, and RFE methods are used to select features from the dataset for classification. Relevant features are selected by removing the redundant and irrelevant features from the dataset. To execute FS process, ANOVA F-test is

performed in which every feature is examined separately to defined the connection between the feature and labels. After analyzing each feature, features with highest percentile value are selected.

- **Build the model** After selecting the features from the dataset, seven classifier models are built. The classifier considered for experimentation of IDS dataset are DT, RF, NB, SVM, LR, k-NN, and ANN.
- **Prediction and Evaluation** Test dataset is used for making predictions for the model built and 10-fold cross validation is applied. Performance metrics such as accuracy, precision, recall, and f -score are considered to demonstrate the results, and for comparative analysis of the classifiers.

6.1 Experimental details and results

In this paper, the training set KDDTain+.TXT and the test set KDDTest+.TXT is used that contains 126,620 and 22,850 instances, respectively. The ML techniques adopted for the experimentation along with their parameter settings are listed in Table 4. The FS methods applied on the dataset are IG, Chi-Square, and RFE. The results of the experiments are represented using accuracy, precision, recall, and f -score that are evaluated using Eqs. 3–6, respectively.

$$\text{Accuracy} = \frac{T_p + T_n}{T_p + F_p + F_n + T_n} \quad (3)$$

$$\text{Precision} = \frac{T_p}{T_p + F_p} \quad (4)$$

$$\text{Recall} = \frac{T_p}{T_p + F_n} \quad (5)$$

$$f\text{-score} = \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6)$$

where, T_p , T_n , F_p , and F_n represents true positive, true negative, false positive, and false negative, respectively.

The results for all the seven classifiers without using FS technique and by applying FS technique is presented in Table 5. The accuracy, precision, recall, and f -score value is calculated for each of the attack class of the dataset. As depicted from the results SVM and ANN perform the best among the other classifiers in terms of accuracy, precision,

recall, and f -score. To increase the efficiency and effectiveness for intrusion detection, we have applied FS methods to select relevant features from the 41 features available in the dataset. Results demonstrate that RFE feature selection method performs better compared to other feature selection techniques and gives high accuracy rate for each of the attack classes.

7 Result analysis and discussion

The aim of our study is to incorporate feature engineering to enhance the performance of ML techniques for attack classification. We have used NSL-KDD dataset for evaluating the performance of ML techniques. We have implemented feature selection using three different feature selection techniques namely Chi-Square, IG, and RFE. These techniques use different metrics for the selection of optimal feature subset. For instance, Chi-Square calculates the chi-scores for ranking the features, IG ranks the features based on their gain value, whereas RFE determines the feature subset by performing greedy search and recursively derives the optimal set of features. These feature selection techniques can obtain optimal features for each of the target variables (DoS, Probe, R2L, U2R) present in the dataset.

After obtaining optimal features sets, ML techniques such as DT, RF, NB, k-NN, LR, SVM, and ANN are used to built intrusion detection models for attack classification. The intrusion detection models are built on the training samples using all features of NSL-KDD dataset as well as selected features obtained from Chi-Square, IG, and RFE. Further, the performance of ML techniques is compared in terms of

Table 4 Parameter settings for ML Techniques

ML-technique	Parameter settings
Decision tree classifier	DecisionTreeClassifier(class_weight=None, criterion='gini', min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, presort=False, random_state=0, splitter='best')
Random forest classifier	RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini', max_features='auto', min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, n_estimators=10, n_jobs=2, random_state=None, verbose=0, warm_start=False)
k-nearest neighbours	NearestNeighbors(n_neighbors=5, radius=1.0, algorithm='auto', leaf_size=30, metric='minkowski', p=2, metric_params=None, n_jobs=None)
Logistic regression	LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1, penalty='l2', random_state=42, solver='liblinear', tol=0.0001, verbose=0, warm_start=False)
Naïve Bayes	GaussianNB(priors=None, var_smoothing=1e-09)
Support vector machine	SVC(C=100, cache_size=200, class_weight=None, coef0=0.0, decision_function_shape='ovr', degree=3, gamma=0.001, kernel='rbf', max_iter=-1, probability=False, random_state=None, shrinking=True, tol=0.001, verbose=False)
Artificial neural network	Input layer neurons = 122, Number of hidden layer = 2, dropout probability = 0.2, regularization = l2, activation = sigmoid, optimizer = adam, loss = binary_crossentropy, number of neurons in first hidden layer = 366, number of neurons in second hidden layer = 60

Table 5 Experimental results

K=10	Experimental Results															
	DoS				Probe				R2L				U2R			
	Acc	Pre	Recall	<i>f</i> -score	Acc	Pre	Recall	<i>f</i> -score	Acc	Pre	Recall	<i>f</i> -score	Acc	Pre	Recall	<i>f</i> -score
DT-all	92.8	96.6	91.2	95.3	95.7	93.1	92.2	93.1	94.2	91.7	91.2	93	92.2	86.4	91.6	88.6
DT-Chi	94.3	95	91.7	93.3	97.9	96.9	96.8	96.8	97.3	96.1	96.3	96.2	99.6	90.8	84.2	86.1
DT-RFE	98.6	99.7	99.5	99.6	99.3	99.1	98.8	99	97.7	97.3	96.4	96.8	99.7	96.6	84.1	86.8
DT-IG	99.6	99.5	99.6	99.5	99.5	99.3	99.2	99.3	97.9	97.2	96.9	97	99.6	86.4	91.6	88.6
RF-all	92.7	96.6	91	95.2	92.1	93.2	90.1	94.2	91.2	93.6	91.5	92.3	92.5	90.7	86.1	88.7
RF-Chi	94.5	94.8	92.4	93.6	98.1	97.3	97.2	97.2	97.6	96.5	96.1	96.5	99.6	91.5	82.6	85.8
RF-RFE	98.7	99.6	99.7	99.6	99	98.6	98.4	98.5	97.4	96.6	96	96.3	99.6	87.7	89.1	88.4
RF-IG	99.1	98.9	99	98.8	99.3	98.9	99	99	97.6	96.9	96	96.4	99.6	96.3	80.4	85.6
KNN-all	84.7	90.1	83.3	89.6	87.1	92.2	89.3	86.03	90.5	92.5	87.1	89.3	84.5	92.1	94.2	88.7
KNN-Chi	96.1	96.6	96.9	96.8	91.6	89.6	90.7	94.8	96.9	92.2	94.6	94.8	97.4	94.8	97.4	98.6
KNN-RFE	95.3	95	97.3	96.1	94.4	94.6	99	96.8	96.6	99	97.5	98.2	99.4	94.6	96.8	96.1
KNN-IG	99.9	99.4	99.2	99.3	99.2	99.5	99.3	98.2	98.6	99.6	99.1	99.9	97.4	94.8	97.4	98.6
LR-all	84.8	98.7	82.1	89.7	85.1	95.1	83.3	90.2	88.3	92.5	86.3	87.4	89.3	91	86.1	88.6
LR-Chi	97.6	96.9	95.2	97.1	94.7	93.6	93.6	94.1	91.7	84.8	94.9	96.2	98.2	92.8	97.2	95.6
LR-RFE	96.8	95.7	99	97.3	89.1	88.8	99.7	94	98.2	99	99.2	99.1	99.1	95.7	96.7	97.1
LR-IG	96.1	94.3	99.3	96.8	94.7	93.6	93.6	94.1	99.9	84.8	94.9	96.2	99.9	97.8	99.2	98.3
NB-all	92.9	97.8	92.3	95.5	93.1	96.3	93.2	92.2	93.4	92.3	91.7	93.4	89.7	88.6	90.1	87.3
NB-Chi	96.1	96.6	96.9	96.8	91.6	90.1	90.6	94.8	96.9	91.9	97.6	94.8	97.4	99.8	97.4	98.6
NB-RFE	95.3	95	97.3	96.1	94.4	94.6	99	96.8	96.6	99	97.5	98.2	84	94.1	84	91.3
NB-IG	96.4	94.6	97.8	95.7	91.6	95.1	95.6	94.8	92.9	98.9	98.6	99.6	99.1	97.9	96.1	98.4
SVM-all	93.5	94.5	89.3	92.2	92.2	93.2	91.2	95.1	94.3	92.2	92.7	93.4	92.2	91.7	93.6	91
SVM-Chi	97.4	96.7	95.3	96.8	97.8	97.1	94.2	95.6	93.8	96.8	93.7	98.6	91.9	97.5	92.9	93.9
SVM-RFE	99.6	95.6	99.7	97.6	98.9	99.1	99.6	99.3	98.2	99	99.2	99.1	99.1	97.2	98.5	97.6
SVM-IG	99.6	99.4	99.2	99.3	99.2	99.5	96.3	98.2	98.6	98.6	99.1	99.9	97.4	94.8	97.4	93.6
ANN-all	91.2	86.97	87.5	86.03	91.1	88.7	90.2	88.2	89.5	92.2	93.7	92.1	86.5	90.7	91.2	90.94
ANN-Chi	97.1	89.8	90.2	92.2	83.5	90.2	93.3	92.2	96.5	87.8	90.2	93.3	99.9	97.5	98.9	97.2
ANN-RFE	98.32	97.7	99.7	97.2	94.4	99.2	99.7	97.5	96.8	99.2	97.2	98.2	99.9	97.2	98.5	91.3
ANN-IG	90.11	93.9	95.5	94.3	93.9	92.9	94.7	93.8	96.8	96.6	96.8	96.3	99.9	98.6	97.8	96.2

Bold values in the table represent best results for the experiments performed

accuracy, precision, recall, and *f*-score. The results obtained provide a perception of using the classification algorithms along with FS methods to enhance the efficiency and effectiveness of the system. The FS methods implemented selects relevant and important features from the feature set in order to increase the classification accuracy of the system.

The major goal of the work was to analyze the effect of feature selection techniques on detection rate and accuracy of the system. The accuracy, precision, recall, and *f*-score are measured for attack classes by considering the entire feature set (41 features) and using the feature set obtained by applying the FS algorithms (Chi-Square, IG, and RFE). Here, in our experiments, each of the attack class were evaluated individually as they exhibit different characteristics. Relevant features of each of the attack class were obtained and accuracy, precision, recall, and *f*-score for each of the attack

classes is evaluated. A comparative analysis of various classifiers is presented which shows that the performance of the model is improved if FS methods are incorporated in the model.

The idea behind incorporating feature engineering is that not all network attributes extracted from the network packet are useful in identifying malicious network traffic. This implies that dataset might contain redundant and irrelevant features that might lead to overfitting of the intrusion detection model built using ML technique. Therefore, applying feature selection helps to reduce the over-fitting as well as defines the best search space for classification.

The experimental results show that the combination of SVM classifier with RFE yields comparatively better result with accuracies for DoS, Probe, R2L, and U2R as 99.6%, 98.9%, 98.2%, and 99.1%, respectively. This is because

RFE feature selection technique measures the usefulness of every attribute of the dataset. It considers accuracy metric to derive optimal attributes or set of attributes that contribute better in predicting the target variable. Whereas, Chi-Square and IG feature selection techniques measure the relevance of the attributes by considering collinearity and correlation between them. RFE uses cross validation for evaluating the feature subsets whereas, Chi-Square and IG use statistical techniques for evaluating the feature subsets.

The experimental results of IDS by incorporating FS methods on NSL-KDD dataset is presented in Table 5. Based on the experiments conducted the results analysis can be concluded in the following points:

- FS process enhances the efficiency and effectiveness of the system as compared to using the entire feature set for attack classification.
- Out of the three FS methods implemented, RFE performs comparatively well.
- Out of the seven classifier implemented, SVM performs well for all the attack classes in terms of accuracy, precision, recall, and f -score.

A comparison of various studies on IDS using ML techniques by applying different FS techniques is summarized in Table 6. Different performance measure are considered for demonstrating the performance of ML techniques. Comparing the FS techniques used, it can be stated that RFE contributes comparatively better in attack classification.

8 Conclusion

The study in the field of IDS has shown that there two major processes that are important in building an effective IDS model namely, an efficient classification algorithm and a good FS algorithm for attack classification. In this paper, three FS algorithms are used namely, Chi-square, IG, and RFE methods. Chi-Square and IG are filter based FS methods while RFE is wrapper based FS method. These methods are chosen to study which of the three FS algorithms performs well with the ML classifiers such as DT, RF, LR, k-NN, NB, SVM, and ANN. The experiments are conducted on NSL-KDD dataset. The performance of RFE is comparatively well for all the attack classes. The performance has

Table 6 Study on IDS with feature selection using NSL-KDD dataset

Year	References	Feature selection	Classifier	Result analysis
2015	Wahba et al. (2015)	Correlation-based, IG	Adaboost	F-measure: 98%, FPR: 0.041
2015	Deshmukh et al. (2015)	Fast correlation-based	NB, Hidden NB, NBTree	<ul style="list-style-type: none"> · For NB · Accuracy: 88.20% · Error rate: 11.80% · For Hidden NB · Accuracy: 93.40% · Error rate: 6.60% · For NBTree · Accuracy: 94.6% · Error rate: 5.40%
2015	Ingre and Yadav (2015)	Correlation-based	ANN	Accuracy: 81.2%
2016	Kumar and Batth (2016)	Correlation-based, IG, Gain ratio	NB	<ul style="list-style-type: none"> · Accuracy with Correlation Based: 97.5% · Accuracy with IG: 97.1% · Accuracy with Gain ratio: 98.6%
2016	Subba et al. (2016)	PCA	SVM, DT, NB	<ul style="list-style-type: none"> · Accuracy of SVM: 99.13% · Accuracy of DT: 96.85% · Accuracy of NB: 94.56%
2017	Thaseen and Kumar (2017)	Chi-square	SVM	Accuracy: 98%, FAR: 0.13%
2017	Mkuzangwe and Nelwamondo (2017)	IG	DT, Adaboost	Accuracy: 90%
2017	Bitaab and Hashemi (2017)	IG	DT, Gaussian mixture model	Accuracy: 94.28%
2018	Benaddi et al. (2018)	PCA	k-NN	Accuracy: 94%
2020	Our study	IG, Chi-square, RFE	DT, RF, NB, LR, k-NN, SVM, and ANN	<ul style="list-style-type: none"> · Out of the three feature selection technique RFE outperformed and in ML techniques SVM gave better results for all attack categories. · For DoS: Accuracy 99.6% · For Probe: Accuracy 98.9% · For R2L: Accuracy 98.2% · For U2R: Accuracy 99.1%

been measured in terms of accuracy, precision, recall, and *f*-score and results are validated using 10-fold cross validation. Finally, based on the experimental results obtained, it can be stated that performance of the IDS model is enhanced by applying the FS algorithms. Although the selected FS algorithms have given promising results for the attack classification, the FS model could further be improved by optimizing the search process. In fact, the influence of the unbalanced distribution of the data samples on the performance of the IDS has to be taken into consideration in future.

Compliance with ethical standards

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

References

- Agarwal N, Hussain SZ (2018) A closer look at intrusion detection system for web applications. *Secur Commun Netw* 2018:1–27. <https://doi.org/10.1155/2018/9601357>
- 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
- Allahyari M, Pouriye S, Assefi M, Safaei S, Trippe ED, Gutierrez JB, Kochut K (2017) A brief survey of text mining: Classification, clustering and extraction techniques. *arXiv preprint arXiv:170702919*
- Almseidin M, Alzubi M, Kovacs S, Alkasassbeh M (2017) Evaluation of machine learning algorithms for intrusion detection system. In: 2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY), IEEE, pp 000277–000282
- 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
- Balasaraswathi VR, Sugumaran M, Hamid Y (2017) Feature selection techniques for intrusion detection using non-bio-inspired and bio-inspired optimization algorithms. *J Commun Inform Netw* 2(4):107–119
- Benaddi H, Ibrahim K, Benslimane A (2018) Improving the intrusion detection system for nsl-kdd dataset based on pca-fuzzy clustering-knn. In: 2018 6th International Conference on Wireless Networks and Mobile Communications (WINCOM), IEEE, pp 1–6
- Besharati E, Naderan M, Namjoo E (2019) Lr-hids: logistic regression host-based intrusion detection system for cloud environments. *J Ambient Intell Human Comput* 10(9):3669–3692
- Biswas SK (2018) Intrusion detection using machine learning: a comparison study. *Int J Pure Appl Math* 118(19):101–114
- Bitaab M, Hashemi S (2017) Hybrid intrusion detection: Combining decision tree and gaussian mixture model. In: 2017 14th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC), IEEE, pp 8–12
- Breiman L (2017) Classification and regression trees. Routledge, Abingdon
- Chomboon K, Chujai P, Teerarassamee P, Kerdprasop K, Kerdprasop N (2015) An empirical study of distance metrics for k-nearest neighbor algorithm. In: Proceedings of the 3rd international conference on industrial application engineering, pp 1–6
- Chou TS, Yen KK, Luo J (2008) Network intrusion detection design using feature selection of soft computing paradigms. *Int J Computat Intell* 4(3):196–208
- Da Silva IN, Spatti DH, Flauzino RA, Liboni LHB, dos Reis Alves SF (2017) Artificial neural networks. Springer International Publishing, Cham
- Dash M, Liu H (1997) Feature selection for classification. *Intelligent data analysis* 1(1–4):131–156
- Denning DE (1987) An intrusion-detection model. *IEEE Trans Softw Eng* 2:222–232
- Deshmukh DH, Ghorpade T, Padiya P (2015) Improving classification using preprocessing and machine learning algorithms on nsl-kdd dataset. In: 2015 International Conference on Communication, Information & Computing Technology (ICCICT), IEEE, pp 1–6
- Dogan Ü, Glasmachers T, Igel C (2016) A unified view on multi-class support vector classification. *J Mach Learn Res* 17(45):1–32
- Ektefa M, Memar S, Sidi F, Affendey LS (2010) Intrusion detection using data mining techniques. In: 2010 International Conference on Information Retrieval & Knowledge Management (CAMP), IEEE, pp 200–203
- Fadlil A, Riadi I, Aji S (2017) Ddos attacks classification using numeric attributebased gaussian naive bayes. *Int J Adv Comput Sci Appl (IJACSA)* 8(8):42–50
- Hackeling G (2017) Mastering Machine Learning with scikit-learn. Packt Publishing Ltd, pp 1–254. <https://www.packtpub.com/in/big-data-and-business-intelligence/mastering-machine-learning-scikit-learn-second-edition>
- Harrell FE Jr (2015) Regression modeling strategies: with applications to linear models, logistic and ordinal regression, and survival analysis. Springer, Berlin
- Heba FE, Darwish A, Hassanien AE, Abraham A (2010) Principle components analysis and support vector machine based intrusion detection system. In: Intelligent Systems Design and Applications (ISDA), 2010 10th International Conference on, IEEE, pp 363–367
- Ingre B, Yadav A (2015) Performance analysis of nsl-kdd dataset using ann. In: 2015 International Conference on Signal Processing and Communication Engineering Systems, IEEE, pp 92–96
- 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
- Kloft M, Brefeld U, Düessel P, Gehl C, Laskov P (2008) Automatic feature selection for anomaly detection. In: Proceedings of the 1st ACM workshop on Workshop on AIsec, ACM, pp 71–76
- Kumar K, Bath JS (2016) Network intrusion detection with feature selection techniques using machine-learning algorithms. *Int J Comput Appl* 150(12):1–13. <https://doi.org/10.5120/ijca2016910764>
- Kumari B, Swarnkar T (2011) Filter versus wrapper feature subset selection in large dimensionality micro array: a review. *Int J Comput Sci Inf Technol* 2(3):1048–1053
- Larson D (2016) Distributed denial of service attacks-holding back the flood. *Netw Secur* 2016(3):5–7
- Li J, Cheng K, Wang S, Morstatter F, Trevino RP, Tang J, Liu H (2017) Feature selection: A data perspective. *ACM Comput Surv* 50:94:1–94:45
- Maillo J, Ramírez S, Triguero I, Herrera F (2017) knn-is: an iterative spark-based design of the k-nearest neighbors classifier for big data. *Knowl Based Syst* 117:3–15
- 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
- Mansournia MA, Geroldinger A, Greenland S, Heinze G (2017) Separation in logistic regression: causes, consequences, and control. *Am J Epidemiol* 187(4):864–870

- Mayuranathan M, Murugan M, Dhanakoti V (2019) Best features based intrusion detection system by rbm model for detecting ddos in cloud environment. *J Ambient Intell Human Comput*: 1–11
- 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 Inform Syst Secur (TISSEC)* 3(4):262–294
- Meira J, Andrade R, Praça I, Carneiro J, Bolón-Canedo V, Alonso-Betanzos A, Marreiros G (2019) Performance evaluation of unsupervised techniques in cyber-attack anomaly detection. *J Ambient Intell Human Comput*. <https://doi.org/10.1007/s12652-019-01417-9>
- Meyer D, Wien FT (2015) Support vector machines. *Interf Libsvm Pack* e1071:28
- Mkuzangwe NN, Nelwamondo F (2017) Ensemble of classifiers based network intrusion detection system performance bound. In: 2017 4th International Conference on Systems and Informatics (ICSAI), IEEE, pp 970–974
- Mousavi SM, Majidnezhad V, Naghipour A (2019) A new intelligent intrusion detector based on ensemble of decision trees. *J Ambient Intell Human Comput*. <https://doi.org/10.1007/s12652-019-01596-5>
- Mukherjee S, Sharma N (2012) Intrusion detection using naive bayes classifier with feature reduction. *Proc Technol* 4:119–128
- 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 H, Franke K, Petrovic S (2010) Improving effectiveness of intrusion detection by correlation feature selection. In: Availability, Reliability, and Security, 2010. ARES'10 International Conference on, IEEE, pp 17–24
- Olusola AA, Oladele AS, Abosede DO (2010) Analysis of kdd'99 intrusion detection dataset for selection of relevance features. *Proc World Cong Eng Comput Sci Citeseer* 1:20–22
- Phutane MT, Pathan A (2015) Intrusion detection system using decision tree and apriori algorithm. *J Comput Eng Technol* 6(7):09–18
- Puga JL, Krzywinski M, Altman N (2015) Points of significance: Bayes' theorem. *Nat Methods* 12:277–278. <https://doi.org/10.1038/nmeth.3335>
- Rajput D, Thakkar A (2019) A survey on different network intrusion detection systems and countermeasure. *Emerging Research in Computing, Information, Communication and Applications*, Springer, pp 497–506
- Richhariya R, Manjhar AK, Makwana RRS (2017) A hybrid approach for user to root and remote to local attack. *Int J Comput Sci Eng* 5(6):73–79
- Russell SJ, Norvig P (2016) *Artificial intelligence: a modern approach*. Pearson Education Limited, Malaysia
- Sahani R, Rout C, Badajena JC, Jena AK, Das H, et al. (2018) Classification of intrusion detection using data mining techniques. In: *Progress in computing, analytics and networking*, Springer, pp 753–764
- Sharafaldin I, Lashkari AH, Ghorbani AA (2018) Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: *ICISSP*, pp 108–116
- Smaha SE (1988) Haystack: An intrusion detection system. In: [Proceedings 1988] Fourth Aerospace Computer Security Applications, IEEE, pp 37–44
- Song YY, Ying L (2015) Decision tree methods: applications for classification and prediction. *Shanghai Arch Psychiatry* 27(2):130
- 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
- Suthaharan S (2016) Support vector machine. In: *Machine learning models and algorithms for big data classification*, vol 36. Springer, pp 207–235
- Tavallae M, Bagheri E, Lu W, Ghorbani AA (2009) A detailed analysis of the kdd cup 99 data set. In: 2009 IEEE symposium on computational intelligence for security and defense applications, IEEE, pp 1–6
- Thakkar A, Lohiya R (2020a) A review of the advancement in intrusion detection datasets. *Procedia Comput Sci* 167:636–645. <https://doi.org/10.1016/j.procs.2020.03.330>
- Thakkar A, Lohiya R (2020b) Role of swarm and evolutionary algorithms for intrusion detection system: a survey. *Swarm Evolut Comput* 53:100631
- Thaseen IS, Kumar CA (2017) Intrusion detection model using fusion of chi-square feature selection and multi class svm. *J King Saud Univ Comput Inform Sci* 29(4):462–472
- van Gerven M, Bohte S (2018) Artificial neural networks as models of neural information processing. *Frontiers Media SA*. <https://www.frontiersin.org/research-topics/4817/artificial-neural-networks-as-models-of-neural-information-processing>
- Wahba Y, ElSalamouny E, ElTaweel G (2015) Improving the performance of multi-class intrusion detection systems using feature reduction. *arXiv preprint arXiv:150706692*
- Witten IH, Frank E, Hall MA, Pal CJ (2016) *Data mining: practical machine learning tools and techniques*, 3rd edn. Morgan Kaufmann, pp 1–629. ISBN 978-0-12-374856-0. <https://doi.org/10.1016/B978-0-12-374856-0.00002-X>
- Zainal A, Maarof MA, Shamsuddin SM et al (2009) Ensemble classifiers for network intrusion detection system. *J Inform Assur Secur* 4(3):217–225
- Zaman S, Karray F (2009) Features selection for intrusion detection systems based on support vector machines. In: *Consumer Communications and Networking Conference, 2009. CCNC 2009*. 6th IEEE, IEEE, pp 1–8

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