Hindawi
Security and Communication Networks
Volume 2020, Article ID 8875404, 17 pages
https://doi.org/10.1155/2020/8875404

Research Article

Feature Selection Based on Cross-Correlation for the Intrusion Detection System

Gholamreza Farahani

*Department of Electrical Engineering and Information Technology,
Iranian Research Organization for Science and Technology (IROST), Sh. Ehsani Rad St., Enqelab St., Parsa Sq.,
Ahmadabad Mostoufi Rd., Azadegan Highway, Tehran 3353136846, Iran*

Correspondence should be addressed to Gholamreza Farahani; farahani.gh@irost.org

Received 2 March 2020; Revised 12 August 2020; Accepted 7 September 2020; Published 22 September 2020

Academic Editor: Leonardo Mostarda

One of the important issues in the computer networks is security. Therefore, trusted communication of information in computer networks is a critical point. To have a safe communication, it is necessary that, in addition to the prevention mechanisms, intrusion detection systems (IDSs) are used. There are various approaches to utilize intrusion detection, but any of these systems is not complete. In this paper, a new cross-correlation-based feature selection (CCFS) method is proposed and compared with the cuttlefish algorithm (CFA) and mutual information-based feature selection (MIFS) features with use of four different classifiers: support vector machine (SVM), naive Bayes (NB), decision tree (DT), and K-nearest neighbor (KNN). The experimental results on the KDD Cup 99, NSL-KDD, AWID, and CIC-IDS2017 datasets show that the proposed method has a better performance in accuracy, precision, recall, and *F*1-score criteria in comparison with the other two methods in different classifiers. Also, the results on different classifiers show that the usage of the DT classifier for the proposed method is the best.

1. Introduction

Networks and intelligence devices are one of the most important parts of modern life, which without them, the present life will not be possible. Today, all communications, economics, and entertainment are based on computer networks. The disruption of these networks will result in incalculable costs. Despite criticality and ongoing research studies, computer networks remain an open issue due to the complexity and nature of interconnection.

In order to provide complete security in a computer system, in addition to firewalls and intrusion prevention devices, other systems called IDSs are required. IDSs are able to recognize and deal with attacks, if the attacker crosses the firewall, antivirus, and other security devices [1].

In today's world, computers and computer networks which are connected to the internet play a major role in communications and information transmission. In the meanwhile, the profitable people who have access to the important information of specific centers or information of other people have violated the computer systems. Their intention is inflicting influence or pressure or even disrupting the organization of systems.

Hacker, cracker, and intruder are words that are nowadays more commonly found in computer circles and affect other systems and compromise their security. Therefore, the need to maintain information security in computer networks that are connected with the outside world is quite tangible. Because of the technical impossibility of creating computer systems (hardware and software) without weaknesses and failures of security, therefore, intrusion detection in the research of computer systems has particular importance. IDSs have been used to help system security administrators to detect intrusions and attacks. The purpose of an IDS is not to prevent an attack; it is only detecting attacks and security issues in a system or computer network and announcing them to the system administrator. In general, IDSs are used alongside firewalls and complementary security systems [2].

One of the problems in the implementation of intrusion detection devices is the vast information and high number of features of network data. The existence of a large number of these unrelated and redundant features in the dataset negatively impacts the performance of the machine learning algorithms and increases computational complexity. Therefore, reducing the size of the dataset is a major task in data mining and machine learning applications. High-dimensional datasets from two sides will reduce the classifier performance. From one side, by increasing the dimensions of the data, the volume of computing increases. On the other side, a model based on high-dimensional data has a lower generalization capability. A major solution to deal with this problem is to reduce the dimensions of the feature selection problem by selecting important features. These important features have a greater impact on classifications, and based on them, system will design, and other features that are unrelated or unnecessary will ignore. By reducing the dimensions of the problem by selecting the important features, the computational complexity of the IDS decreases, while the performance [3] and accuracy of the IDS [4] increase.

Cho and Park presented a model that uses fuzzy logic and the Markov model to detect intrusion. In this method, the Markov model is used to reduce the size of features [5].

Chebrolu et al. determined the important features that were effective in the development of the IDS by using the Markov algorithm and DT [6]. Also, they used the Bayesian network (BN) and the classification and regression tree (CART) to construct an IDS.

In recent years, two general solutions have been proposed to reduce the dimension which are feature selection [7] and feature extraction [8]. The feature selection, also known as variable selection, chooses a subset of the initial features by searching among existing subsets. However, in extracting features, the primary features are transmitted to a new smaller-sized space.

Also, to obtain better feature sets to detect attacks in the IDS, Le et al. proposed a framework including two main parts: the first part is the feature selection model with use of the sequence forward selection (SFS) algorithm and DT model. The second part is to build various IDS models to train the best-selected feature subset [9].

Pandey proposed new features to obtain low false rate and high accuracy [10]. In the first step, data are filtered by the vote algorithm; therefore, the information gain will get associated with a base learner to choose the necessary features. Then, different classifiers are used. In the other research, filter method which is a method used in feature selection to identify important features and to eliminate less effective features was developed by Gül and Adalı [11].

Ren et al. proposed an effective IDS by using hybrid data optimization which consists of data sampling and feature selection [12]. In data sampling, the isolation forest (iForest) is used to eliminate outliers, genetic algorithm (GA) to optimize the sampling ratio, and the random forest (RF) classifier as the evaluation criteria to obtain the optimal training dataset. In feature selection, GA and RF are used again to obtain the optimal feature subset. Finally, an intrusion detection system based on RF is built using the optimal training dataset obtained by data sampling, and the features are selected by feature selection.

Reviewing different methods in the literature, as mentioned, can conclude that different methods have been proposed to reduce the number of features to detect the attacks quickly with lowest error in the IDS. In this paper, a new method with use of cross-correlation has been proposed that not only reduces the number of features to detect attacks, but also it has proper performance metrics in the detection of attacks.

The paper has been organized into the following sections. Section 2 introduces the IDS and its challenges, different classifiers, and cross-correlation methods. Section 3 describes the proposed method. In Section 4, simulation results are explained. Section 5 includes discussion of the results. Finally, Section 6 concludes the paper.

2. Intrusion Detection System (IDS)

IDS in the network sends relevant reports to the management department by monitoring the network activities to detect malicious actions or defects in the security policies. In short, this system, with seeing anomaly data on the network, warns the system administrator to take steps to prevent the attack [13].

There are several types of intrusion detection architectures that can generally be classified into three: host-based intrusion detection system (HIDS), network-based intrusion detection system (NIDS), and distributed intrusion detection system (DIDS) [14].

Intrusions can be divided into internal and external divisions. External intrusions are those that are inflicted by authorized or unauthorized people from outside the network into the inside of it. Internal intrusions are made by authorized people within the system and the internal network from within the network itself. Intruders generally use software deficiencies, password cracking, network traffic crashes, and design weaknesses in networks, services, or network computers to infiltrate computer systems and networks [15].

2.1. Challenges of the IDS. One of the most important challenges in evolutionary algorithms for the IDS is the loss of diversity as well as getting caught up in the local optimal. This section will review the pros and cons of these methods.

In the last decade, many intrusion detection devices have been introduced to detect anomalies [16]. In general, intrusion detection devices are divided into three general categories [17]:

(1) Abuse or signature-based

(2) Anomalies

(3) Specification-based

Basically, the signature-based intrusion detection method compares the pattern of behaviors in the network with the observed samples before. This method is very effective against the range of known common attacks to determine the possibility of attacks. However, due to various types of attacks and behavioral patterns that can be exploited by the attackers, its performance is limited. The signature-based diagnostic method is the simplest method for detecting attacks on the computer networks because only ongoing activities are reviewed. Like the last packet process, in this method, the last activity reported in this operation is compared with a list of available patterns. This comparison is carried out using comparative methods of spellchecking [18]. The advantage of these methods is precise detection of intrusions whose patterns are exactly given to the system [19].

In an anomaly-based diagnostic method, a view of normal behavior is created. An anomaly may indicate an intrusion. Approaches such as neural networks (NNs) [20], machine learning techniques [21], and even biological safety devices [22] are used to create normal behavioral views. To detect anomaly behavior, you must identify normal behaviors and find special patterns and rules for them. Behaviors that follow these patterns are normal, and events that have statistical deviation from these patterns are recognized as anomaly behavior.

The main advantage of the anomaly-based diagnostic methods is that they can detect, at a minimum cost, different and unknown types of attacks that have not previously been detected in their pattern. The history of the used system is recorded and reviewed in a training phase that may take for days or weeks.

The problem of this method is that due to the complexity and variety of different behaviors that may occur on a network, creating this record needs a lot of time. In addition, accurate diagnosis of the cause of anomalies is not possible.

Specification-based approaches are based on a comparison of events occurring in different communications with instances of a series of events. These instances are related to appropriate and noninvasive protocols, in order that the system can detect anonymous or suspicious events. Unlike an anomaly detection method that relied on the history of network behaviors, here the examined profiles are well-defined, comprehensive, and specific protocols. These protocols are well known, and their implementation process is clear. Therefore, any violation of their proper use can be a suspicious event of the network violation.

Generally, the use of specification-based approaches means that intrusion detection tools can understand, detect, and track the implementation of transmission and application layer protocols. The most important objection to specification-based approaches is that these methods require hardware and software resources. Also, the complexity of analyzing multiple protocols and maintaining the status and tracking the execution process associated with each of the current protocols imposes a lot of overhead on the system. Another serious problem is that these methods are difficult to detect attacks that are based on the standard protocol patterns. Table 1 summarizes the advantages and disadvantages of each method.

2.2. Classifiers. Classification may be defined as a procedure of grouping similar entities with common attributes. It plays a very significant role in the processes of searching and selection. Thus, classification can be used as an efficient tool for various purposes. Previous classifications of attacks for the IDS were performed on the binary (normal/attack) as well as five-class classifications (normal and four classes of attacks). It has been demonstrated that a large number of the features (41 features) are unimportant and may be eliminated without significantly lowering the performance of the IDS [23].

In the five-class classification, Sung and Mukkamala found that, by using 19 of the most important features, instead of the entire 41 features' set, the change in accuracy of intrusion detection was statistically insignificant [23]. They applied the technique of deleting one feature at a time. Each reduced feature set was then tested on SVMs and NNs to rank the importance of the input features. The reduced feature set that yielded the best detection rate in the experiments was considered to be the important feature set. Unlike the research of Sung and Mukkamala, which employed a trial-and-error approach, in this paper, feature reduction is carried out with calculation of cross-correlation (Section 3). Also, recently, SVM has been used for the IDS [24].

Devi and Abualkibash reviewed some supervised learning algorithms for the classification of intrusion detection [25]. They found that KNN has high false rate and detection rate, but AdaBoost algorithm has a very low false rate with high detection rate.

The proposed method is achieved by reducing the data space and then classifying intrusions based on the reduced feature space. To demonstrate the ability to generalize the proposed method in different classifications, in the experiments, four classifiers, SVM, DT, NB, and KNN, have been used. At continuation of this section, these four classifiers are briefly explained.

2.2.1. SVM Classifier. SVM is one of the supervisor learning methods that is used for classification and regression. This method is one of the relatively new methods that has shown good performance in the recent years than the older methods for classification, such as perceptron NNs. The base of the SVM classifier is a linear data classification, and in the linear division of data, it tries to select a line that has a more reliable margin.

For intrusion detection, SVM classifier provides high classification accuracy even if less prior knowledge is available. To find the optimal line for data, it is required to solve the equation by means of quadratic programming (QP) methods that are known methods for solving constrained problems. Various kernel functions, such as Gaussian, exponential, polynomial, and sigmoid can be used. In this paper, Gaussian kernel is used for simulation which is mostly used because of its good features [26].

2.2.2. NB Classifier. This classifier calculates the probability of the input X for each class C_i, $P(X \mid C_i)$, and using the prior probability, $P(C_i)$, and the Bayesian rule, the posterior probability $P(C_i \mid X)$ is obtained as shown in the following equation:

Table 1: Comparison of the performance of the previous methods.

Method	Advantage	Disadvantage
Anomaly-based	(1) Practicability against new attacks (2) Finds network power abuse	(1) Poor accuracy due to the continuous change of events under observation (2) Unavailability of rebuilding profiles (3) The severity of the timely announcement
Signature-based	(1) The easiest and best way to find known problems (2) Detailed analysis of information	(1) Against unspecified attacks, known attacks will change (2) Low understanding of the state and protocols (3) It is hard to keep up-to-date signatures and patterns (4) Time-consuming knowledge preservation
Specification-based	(1) Knowing and tracking the status of protocols (2) Detects the sequence of unexpected commands	(1) Needs to have a lot of resources to follow and test protocols (2) Unreliable attacks that are like as harmless protocols (3) Incompatibility with any operating system

$$P\left(C_i \mid X\right) = \frac{P(C_i)P(X \mid C_i)}{P(X)}. \tag{1}$$

For each class that this probability maximizes, the input is assigned to that class (I^*) which is given in the following equation:

$$I^* = \arg\max_{i} P(Ci \mid X). \tag{2}$$

One of the advantages of this method is that it works well for both numeric data and text data. The main problems of this method are lack of proper estimation, as well as the volume of high and complex computations to estimate the required probabilities. The simple NB method with assuming the independence of the events has solved the problem of estimation. However, in the real world, in many cases, this assumption is not correct. If the dependency of the events is high, this method will not work correctly. However, it has acceptable performance in classifying the text [27].

2.2.3. DT Classifier. A DT is a tree in which the samples are classified in a way that grows from the root to the bottom and eventually reaches the leaf nodes. The characteristics of this tree are as follows:

(1) Each inner or nonleaf node is characterized by an attribute

(2) In each internal node, there are branches as many answers as possible for the question, and each will be specified with the value of that answer

(3) The leaves of this tree are identified by a class or a set of answers

One of the useful algorithms used in the DT is iterative dichotomiser 3 (ID3) algorithm [28]. In this algorithm, the DT is made up to the bottom. This algorithm begins with this question: which feature should be tested at the root of the tree?

To find the answer, a statistical test is used to determine how far each one can individually classify test cases. By choosing this feature for each of its possible values, a branch is created, and trained examples are arranged based on the characteristics of each branch. Following it, the above operation in each branch is to be repeated to select the best feature for the next node. This algorithm is a greedy search in which the previous choices are never reviewed [29].

2.2.4. KNN Classifier. The idea of this method is that points which are in the vicinity of a point are probably in one class. For example, in Figure 1, six neighborhoods are considered. At the *U1* point, four neighbors belong to class B and two neighbors belong to class A. In general, the similarity of this point to class B is greater than class A. Therefore, *U1* is categorized to class B. For the *U2* point, four neighbors belong to class A and two neighbors belong to class B. In general, this point similarity to class A is greater than class B. Therefore, *U2* is categorized to class A.

The advantages of this approach include the simplicity of the design process of classifier implementation, it does not need a learning phase, the low number of parameters, and its high efficiency. This means the KNN method, in most cases, determines the input class well. However, this method has a lot of computation, and therefore, it is very slow. Another major problem is finding the optimal value for the *U* parameter, for which there is no specific method [30].

2.3. Cross-Correlation. Cross-correlation is used when there is a correlation between data and it is possible to use these correlations between data. Therefore, some data will not be used, and thus, the volume of data and calculations will reduce.

The cross-correlation of two sequences $x(n)$ and $y(n)$ is calculated according to the following equation:

$$C(x(n); y(n)) = \sum_{m=-\infty}^{\infty} x(m)y(m+n). \tag{3}$$

If two sequences $x(n)$ and $y(n)$ are the same, the cross-correlation between them will achieve the maximum value.

The cross-correlation is used in various fields, including network intrusion detection. The following is an overview of some of the methods that use cross-correlation to detect network intrusion.

Fleuret made the selection of features based on conditional common data, resulting in a simple and fast algorithm [31]. Amiri et al. proposed a new method for intrusion detection using the mutual information feature selection

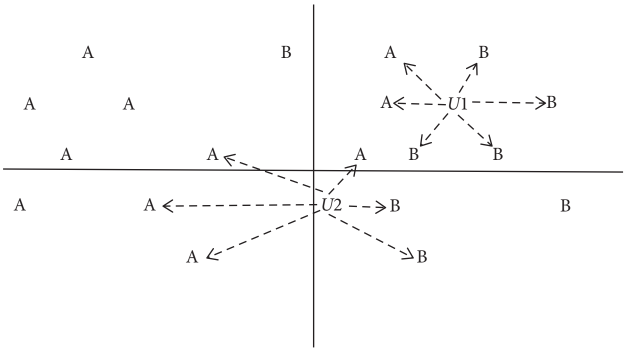


Figure 1: Example for the KNN classifier.

criteria [32]. Also, the value of the features is measured using the mutual information, and the inappropriate features are eliminated.

Zhang et al. [33, 34] proposed an approach based on the cross-correlation of system call sequences for intrusion detection. The main advantage of their method is the low computational load over SVM, hidden Markov model (HMM), or NN methods due to the absence of unnecessary training process. Also, Zhang et al. [35] trained the SVM with Kullback–Leibler (KL) divergence. Then, cross-correlation was calculated by the control and data plane traffic for the detection of anomalies in cyberspace traffic. They proved that usage of cross-correlation can enhance the detection accuracy and detect short-duration intrusions and attacks in the network traffic.

The cross-correlation is used for the detection of shrew distributed denial of service (DDoS) attacks and to distinguish difference between a normal flow and an attack flow by Huang et al. [36].

A meta-heuristic assessment model called the feature correlation analysis and association impact scale were explored by Jyothsna and Rama Prasad [37]. They estimated the degree of intrusion scope threshold from the optimal features of network transaction data available for training. In their strategy, linear canonical correlation for feature optimization was used, and feature association impact scale was explored from the selected optimal features. The results indicated that the feature correlation has a significant impact to minimize the computational and time complexity of measuring the feature association impact scale.

3. Proposed Method

In the classification domains, features may contain false correlations, which hinder the process of detecting intrusions. Also, some features may be redundant since the information they add is contained in other features. Usage of extra features can increase computation time and impact the accuracy of the IDS. Optimal feature selection will improve classification by searching for the subset of features, which best classifies the training data. The feature selection depends on the type of the IDS. It is not known which features are redundant or irrelevant for the IDS and which ones are relevant or essential for the IDS. Therefore, there does not exist any model or function that captures the relationship between different features or between different attacks and features.

With respect to the redundant information in the extracted features, a novel aspect of this paper is usage of cross-correlation to calculate the similarity of features, which reduces the number of features and does not use the irrelevant information that may hinder the IDS. After usage of the proposed method, the accuracy of feature classification for the IDS will increase. Figure 2 demonstrates the detection framework of the proposed method, which consists of three stages including feature selection and dimension reduction with cross-correlation, training of the classifier, and attack recognition. In order to overcome the problem of class imbalance, the proposed feature selection based on cross-correlation is used to eliminate irrelevant features.

In general, the proposed algorithm can be performed as follows:

Step 1: initialization: determine the initial set of all features.

Step 2: compute a set of attributes that are highly correlated with the class but with low intercorrelation. Each element of the feature-class and feature-feature correlation matrices from the training data will be calculated according to equation (3).

In equation (3), $x\ (n)$ is a feature vector and $y\ (n)$ is a class output or other feature vector.

Step 3: select the first feature that maximizes M_s. Equation (4) is used to calculate M_s [38]:

$$M_s = \frac{k.\overline{C_{cf}}}{\sqrt{k + k(k-1).\overline{C_{ff}}}}, \tag{4}$$

where M_s is the correlation between the summed feature subset S, k is the number of subset features, $\overline{C_{cf}}$ is the average of the correlation between subset features and the class variable, and $\overline{C_{ff}}$ is the average intercorrelation between subset features.

Step 4: repeat Steps 2 and 3 until the desired number of features is selected.

Step 5: output set that includes selected features from Step 3.

4. Simulation

This section is intended to simulate the proposed methodology presented in the previous section. First, the datasets that are used in the experiments are explained. Then, the simulation results in comparison with other methods are presented.

System specifications for implementing the proposed method and other compared methods include 2.5 GHz processor, 4 GB physical RAM, Windows 10 operating system, and MATLAB 2019b software implementation tool.

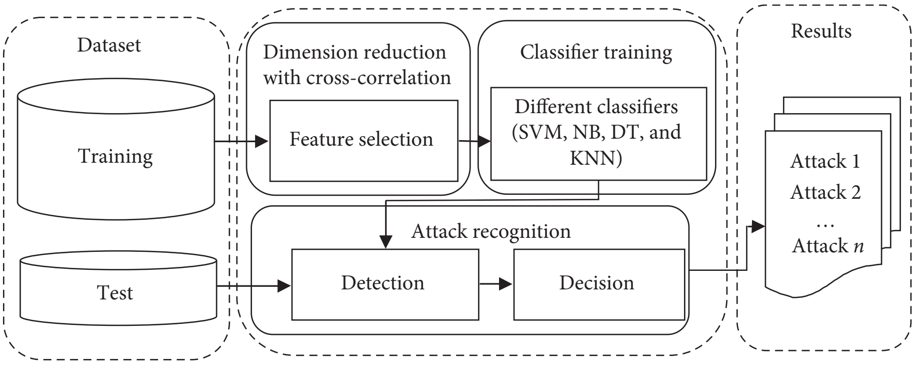


Figure 2: The framework of the proposed feature selection model.

4.1. Datasets. In experiments, four different datasets are used to measure the performance of the proposed method. These datasets are KDD Cup 99, NSL-KDD, AWID, and CIC-IDS2017. At continuation, these datasets are explained briefly.

4.1.1. KDD Cup 99. KDD Cup 99 [39] is a well-known set of intrusion evaluation data. The raw training data were processed into about five million connection records. A connection is a sequence of TCP packets starting and ending at some well-defined times. Each record is unique in the dataset with 41 continuous and nominal features plus one class label.

Connection records in KDD Cup 99 contain 41 features [40]; 10% KDD Cup 99 training and test datasets contain about 494,020 and 31,108 connection records, respectively. Because this dataset is too large, two subsets, training and test datasets, are extracted randomly such that the proportion of each attack both in the training and test datasets is preserved, and each attack is divided by 100. The number of the training data is 4947 and test data is 3117 that are selected randomly for the experiment [41].

The KDD Cup 99 dataset contains 24 attack types that have been categorized into four groups: probe, denial of service (DOS), user to root (U2R), and remote to user (R2L) [42].

4.1.2. NSL-KDD. The NSL-KDD dataset is an effort made by Tavallaee et al. [43] in 2009 as a new revised version of the original dataset KDD Cup 99.

On the one hand, NSL-KDD retained the advantageous and challenging characteristics of KDD Cup 99. On the other hand, it addressed some drawbacks inherited from the original dataset. The benefits of using the NSL-KDD dataset are as follows:

(1) No duplicate records in the test set which have better reduction rates.

(2) The number of selected records from each difficult level group is inversely proportional to the percentage of records in the original KDD Cup 99 dataset.

(3) It has fewer data points than KDD Cup 99, all of which are unique. It is thus less computationally expensive to use for training.

In this study, KDDTrain+, KDDTest+, and KDDTest-21 sets of the NSL-KDD dataset are used. The KDDTrain+ set contains totally 125,973 instances comprising of 58,630 instances of attack traffic and 67,343 instances of normal traffic. However, the KDDTest+ set contains totally 22,544 instances, and as a subset of the KDDTest+ set, the KDDTest-21 set includes totally 11,850 instances [44].

4.1.3. Aegean WiFi Intrusion Dataset (AWID). AWID is a collection of sets of WiFi network data, which contain real traces of both normal and intrusive data collected from real network environments [45]. Each record in the dataset is represented as a vector with 155 attributes, and each attribute has numeric or nominal values. Based on the number of target classes, the dataset can be classified into AWID-CLS and AWIDATK datasets. AWID-CLS dataset groups the instances into 4 main classes including normal, flooding, impersonation, and injection, while the AWID-ATK dataset has 17 target classes that belong to the 4 main classes. On the contrary, based on the number of instances, all the datasets have two different versions: full set and reduced set. In this paper, experiments have been carried out on the reduced four-class dataset (AWID-CLS-R-Tst). In general, AWID-CLS-R-Tst dataset includes totally 575,643 instances [46].

4.1.4. CIC-IDS2017. The CIC-IDS2017 dataset was published by Canadian Institute for Cybersecurity (CIC) in 2017; it contains benign and the most up-to-date common attacks [47]. It also includes the results of the network traffic analysis using CICFlowMeter with labeled flows based on the time stamp, source and destination IPs, source and destination ports, protocols, and attacks. This dataset covers necessary criteria with updated attacks such as DDoS, brute force, XSS, SQL injection, infiltration, port scan, and botnet. In detail, this dataset contains 2,830,743 records devised on 8 files, and each record includes 78 different features with its label.

In order to maintain the same order of magnitude of each dataset while taking into account the requirements of multiclassification, the Wednesday-workingHours set has been chosen for experiments. This set includes totally 691,406 instances belonging to 6 categories, and the detail information could be found in [48].

4.2. Dataset Preprocessing. Data preprocessing is an important step in data mining. Realistic data typically come from heterogeneous platforms and can be noisy, redundant, incomplete, and inconsistent [49]. Thus, it is essential to transform raw data into a suitable format for analysis. In this section, the preprocessing steps including data filtration, data transformation, and normalization are explained.

4.2.1. Data Filtration. Due to the heterogeneity of the platforms, the raw data contain abnormal and redundant instances, which can have a negative influence on the classification accuracy. Therefore, these records need to be removed from the dataset before starting the experiments. For instance, the feature "Fwd Header Length" appears twice in the CIC-IDS2017 dataset, and "Flow Packets/s" includes abnormal values such as "Infinity" and "NaN." Moreover, missing values could be replaced with zeroes, and the features with constant values are dropped out as they do not contribute to the class distinction. Therefore, in the AWID-CLS-R-Tst dataset, 84 features remain from the 155 original features after data filtration.

4.2.2. Data Transformation and Normalization. The datasets contain symbolic, continuous, and binary values. For instance, the feature "protocol type" in the NSL-KDD datasets includes symbolic values such as "tcp," "udp," and "icmp." As many classifiers accept only numerical values, every single value is replaced with an integer in order to handle the symbolic features. Also, different scales of features will degrade the classification performance. For example, features that take on large numeric values such as "Flow Duration" in the CICIDS2017 dataset can dominate the classifier's model relative to features with relatively small numeric values such as "Total Fwd Packets." Therefore, normalization maps features onto a normalized range. One of the simple and fast approaches for normalization is the minimum-maximum method [50] that is used in the experiments which can be defined as the following equation:

$$\overline{x} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \tag{5}$$

where $x_{\min}$ and $x_{\max}$ represent the minimum and maximum values of feature x.

At continuation, the evaluation criteria are expressed. Then, the proposed method named cross-correlation-based feature selection (CCFS) is compared with the mutual information-based feature selection (MIFS) method [32] and cuttlefish algorithm (CFA) [41] on the KDD Cup 99, NSL-KDD, AWID, and CIC-IDS2017 datasets.

4.3. Evaluation Criteria. The proposed method is evaluated based on different performance criteria including accuracy, precision, recall, and $F1$-score [51]. The definitions of these parameters are based on true positive (T_p), false positive (F_p), false negative (F_n), and true negative (T_n).

Accuracy criterion is used to measure how many instances are correctly classified as normal and attack as defined in the following equation:

$$\text{accuracy} = \frac{T_p + T_n}{T_p + F_p + F_n + T_n}. \tag{6}$$

Precision is used to evaluate the true-positive instances in relation to the false-positive instances as shown in the following equation:

$$\text{precision} = \frac{T_p}{T_p + F_p}. \tag{7}$$

The purpose of recall is to evaluate true-positive instances in relation to false-negative instances. The mathematical form of recall is expressed in the following equation:

$$\text{recall} = \frac{T_p}{T_p + F_n}. \tag{8}$$

This $F1$-score criterion is the average of recall and precision. It can be calculated as given in the following equation:

$$F1\text{-score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \tag{9}$$

Sometimes, performance assessment may not be proper with accuracy and recall. If one algorithm has low recall but high precision, then another criterion is needed. Therefore, $F1$-score can solve this problem.

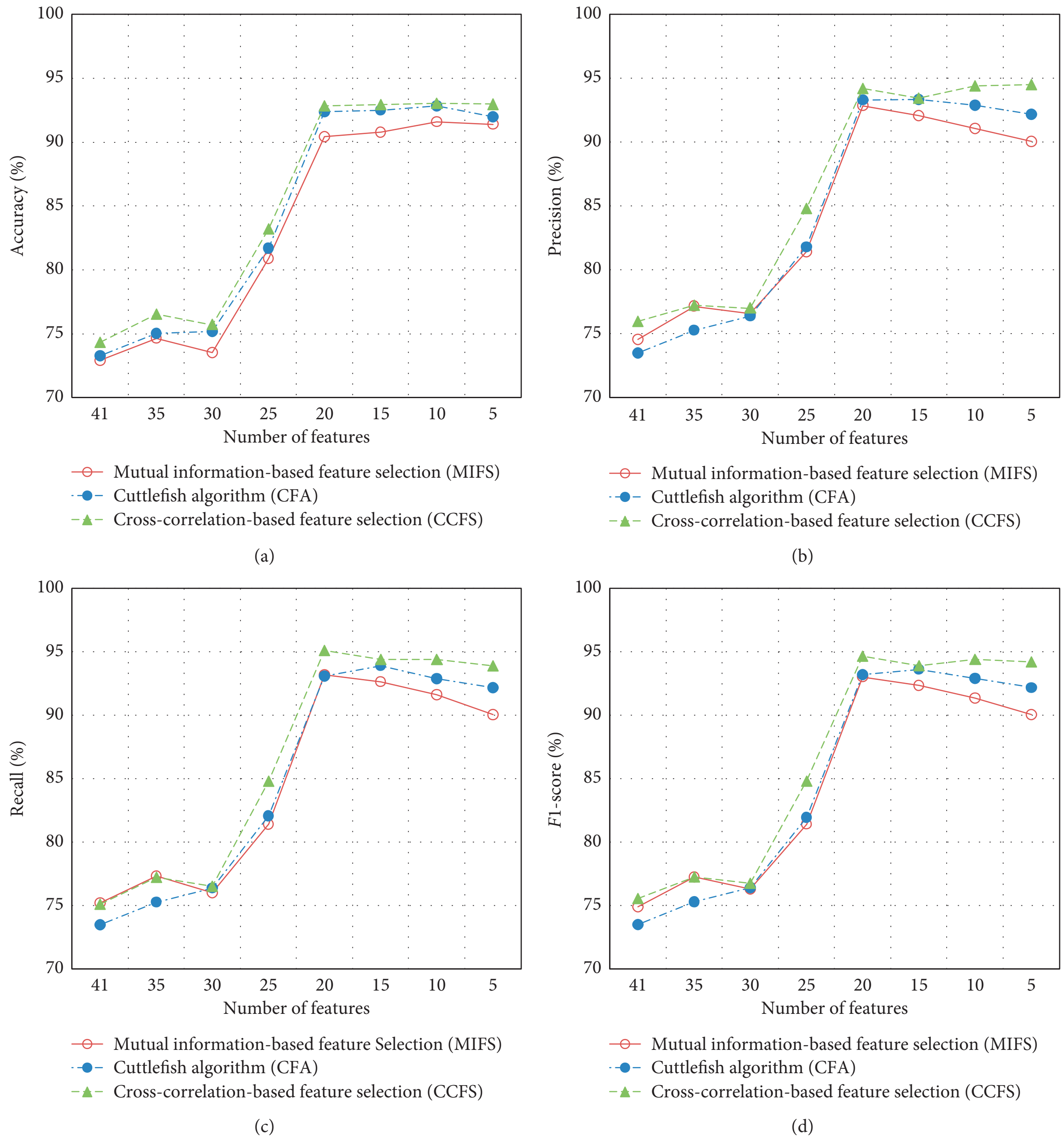
In the next section, the simulation results of the models for 4 different datasets are shown.

4.4. Assessment of the Proposed Method. In this section, a different classification has been used to investigate the intrusion detection performance of the proposed method, and the results are compared with other methods. First, with use of the KDD Cup 99 dataset, 5, 10, 15, 20, 25, 30, 35, and all 41 features of it are used to calculate the accuracy, precision, recall, and $F1$-score criterion in addition to the receiver operating characteristic (ROC) curves with DT, SVM, KNN, and NB classifiers. ROC helps in visualizing a classifier performance by plotting the true-positive rate (TPR) against the false-positive rate (FPR) of the classifier.

The results of the proposed CCFS method are compared with CFA and MIFS methods on the KDD Cup 99 dataset. Then, according to the results of KDD Cup 99, the best number of features is selected, and it has been tested on the NSL-KDD dataset.

At continuation, the AWID and CIC-IDS2017 datasets are used to evaluate the proposed CCFS method.

4.4.1. KDD Cup 99 Dataset Results. The obtained results on the KDD Cup 99 dataset are shown in Figures 3–10. As shown in these figures, the results of three MIFS, CFA, and CCFS methods with four classifiers DT, SVM, KNN, and NB are calculated. In the simulation, for the KNN classifier, $K = 3$, and then the object is simply assigned to the class of its nearest neighbor. The numerical values of accuracy,



(a)

(b)

(c)

(d)

Figure 3: Performance of the CCFS, MIFS, and CFA methods with the DT classifier for the KDD Cup 99 dataset: (a) accuracy, (b) precision, (c) recall, and (d) $F1$-score.

precision, recall, and $F1$-score criteria are average of ten runs for each classifier.

As it is clear from Figures 3, 5, 7, and 9, when the number of features increases, all accuracy, precision, recall, and $F1$-score criteria are decreased. In addition, the proposed method has a better performance in all classifiers when compared with the results using MIFS and CFA methods. Also, the usage of 20 features has best results than other number of features. Furthermore, CFA method has better results than the MIFS method. Between four classifiers, the DT classifier has the best results. In Table 2, the percentage of improvement of the proposed CCFS method in comparison with the CFA method and DT classifier is presented.

According to the ROC of MIFS, CFA, and CCFS methods (Figures 4, 6, 8, and 10), it is clear that the CCFS method has better TPR and less FPR for all classifiers in comparison with MIFS and CFA methods.

4.4.2. NSL-KDD Dataset Results. According to the results obtained from Section 4.4.1, 20 features on KDD Cup 99 have the best performance than other number of features. Therefore, in this section, this number of features is used

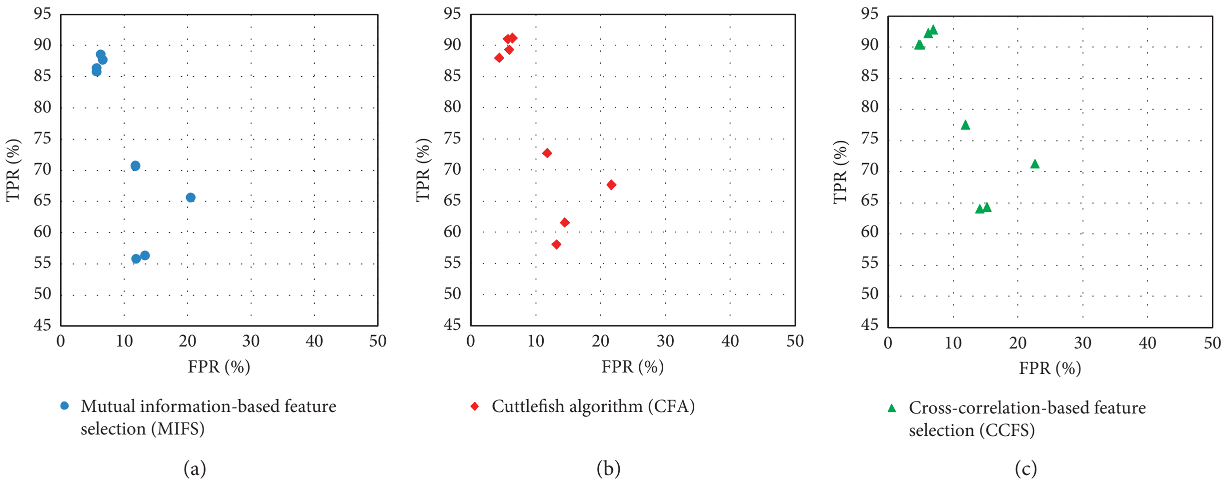


Figure 4: The ROC curve for MIFS, CFA, and CCFS methods with the DT classifier for the KDD Cup 99 dataset.

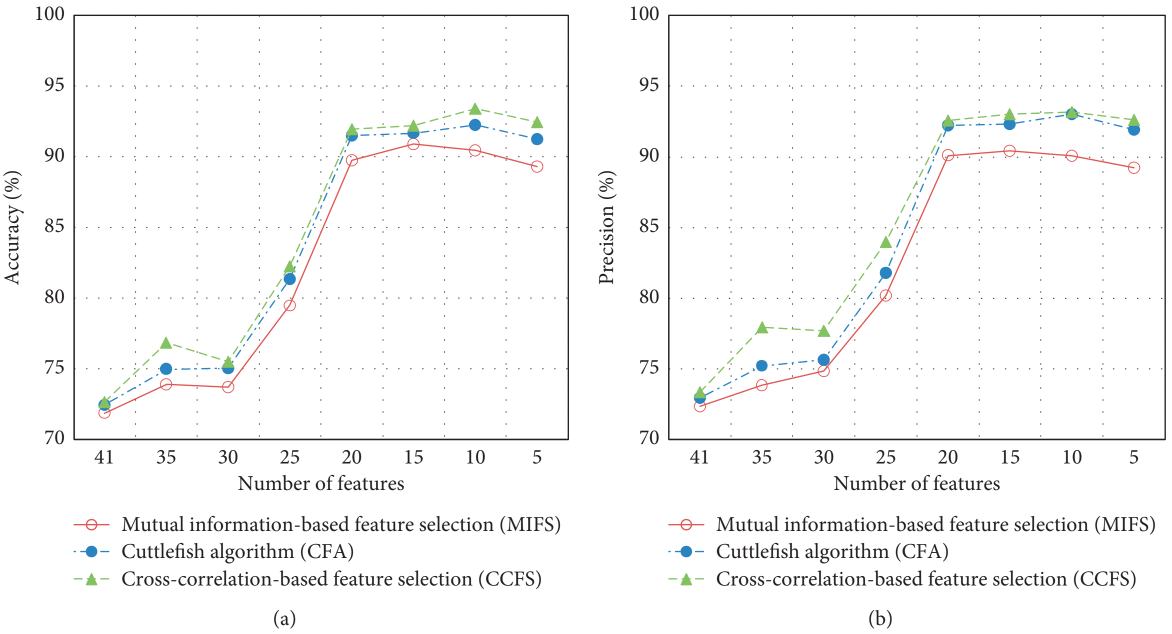


Figure 5: Continued.

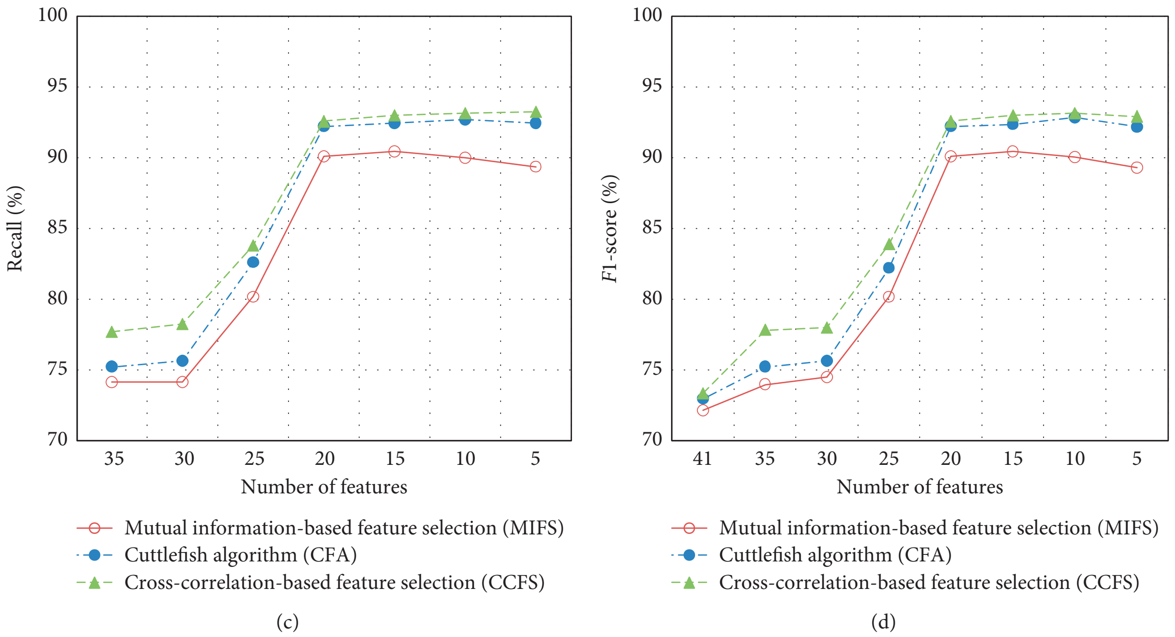


FIGURE 5: Performance of the CCFS, MIFS, and CFA methods with the SVM classifier for the KDD Cup 99 dataset: (a) accuracy, (b) precision, (c) recall, and (d) $F1$-score.

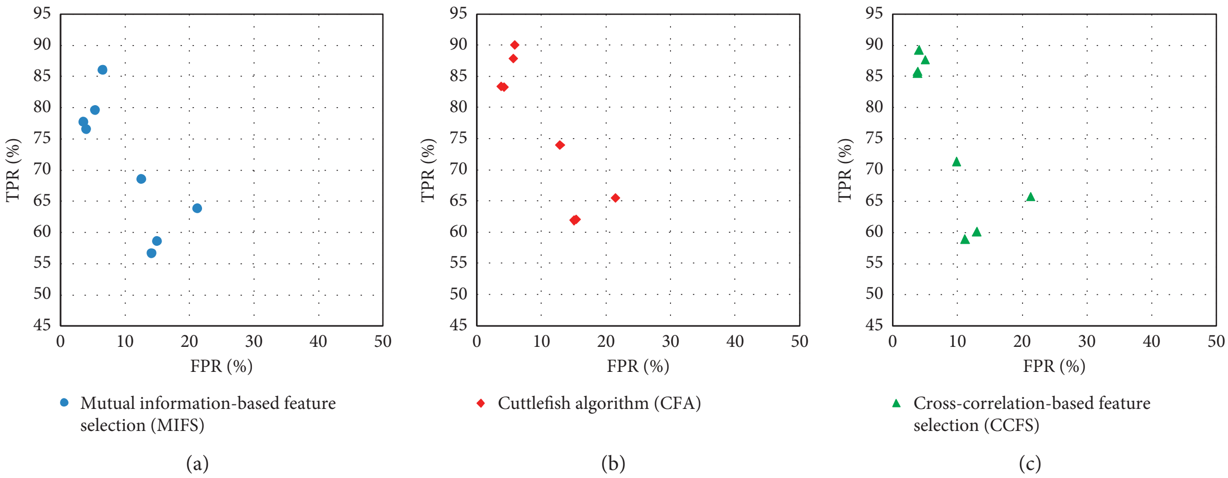


FIGURE 6: The ROC curve for MIFS, CFA, and CCFS methods with the SVM classifier for the KDD Cup 99 dataset.

to compare the results of the MIFS, CFA, and CCFS methods.

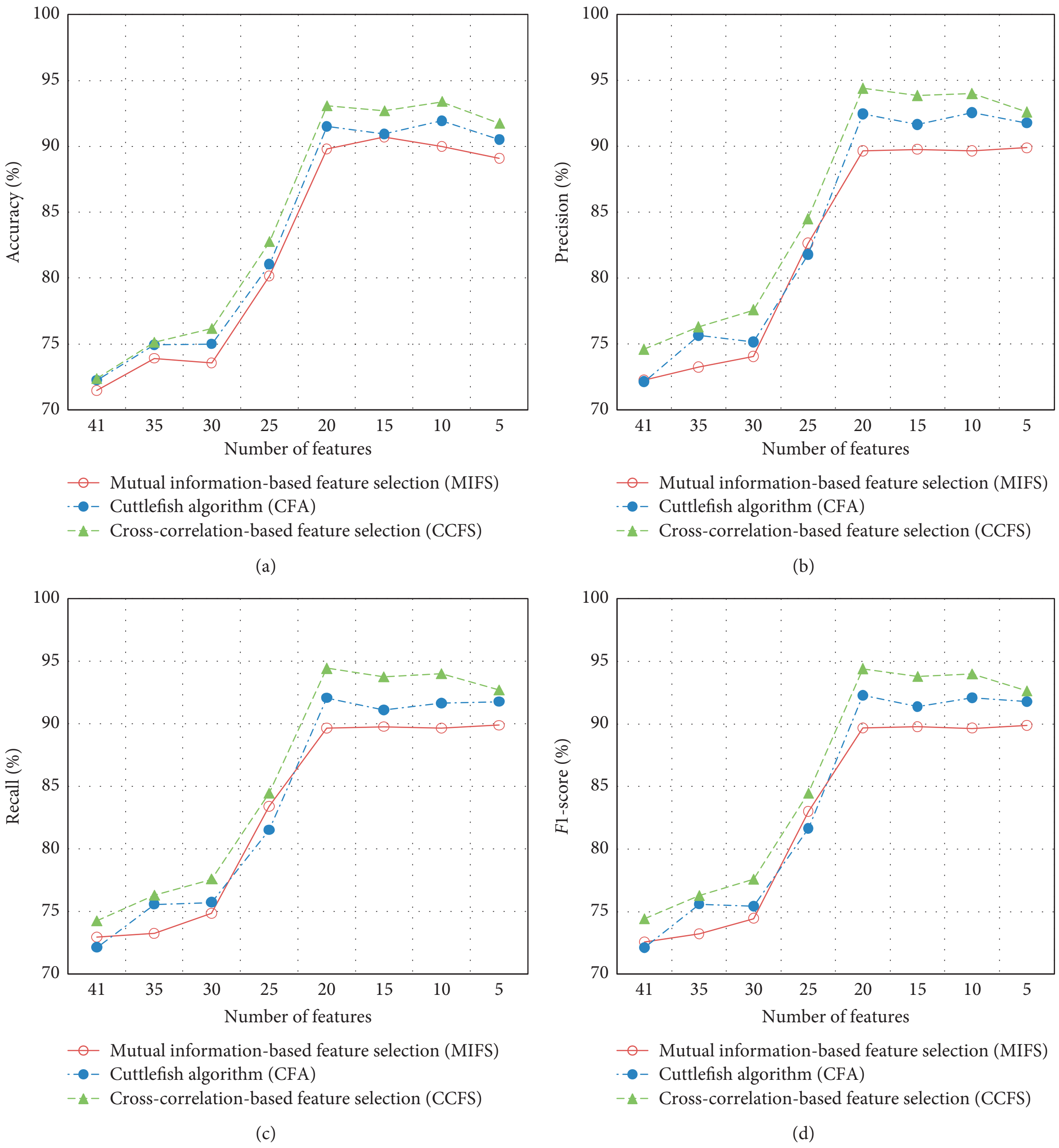
Table 3 shows the accuracy, precision, recall, and $F1$-score criteria for MIFS, CFS, and CCFS methods on the NSL-KDD dataset with 20 features. According to Table 3, CCFS has better results on the accuracy, precision, recall, and $F1$-score than the other two methods. These improvements are shown in Table 4, and they are different for each criterion.

4.4.3. AWID Dataset Results. To verify the performance of the CCFS method on the wireless communication networks, it has been examined on the AWID dataset. The results on the AWID dataset with 84 features are shown in Table 5. The improvement of the CCFS method compared with MIFS and CFS methods is shown in Table 6.

As expected, the results of the proposed CCFS method are also acceptable on the wireless AWID dataset, and better results can be obtained compared with the other two methods.

4.4.4. CIC-IDS2017 Dataset Results. Finally, in this section, the CIC-IDS2017 dataset that includes most up-to-date common attacks has been used to evaluate the proposed CCFS method.

The results of the MIFS, CFA, and CCFS methods for different classifiers are shown in Table 7. The results show the effectiveness of the CCFS for the CIC-IDS2017 dataset.



(a)

(b)

(c)

(d)

Figure 7: Performance of the CCFS, MIFS, and CFA methods with the KNN classifier for the KDD Cup 99 dataset: (a) accuracy, (b) precision, (c) recall, and (d) *F*1-score.

Also, the improvement percentage of the CCFS compared with MIFS and CFA for accuracy, precision, recall, and *F*1-score criteria is shown in Table 8.

5. Discussion

In this paper, the performance of the IDS is evaluated based on its capability of classifying network traffic into a correct type. The proposed CCFS method has been evaluated using the KDD Cup 99, NSL-KDD, AWID, and CIC-IDS2017 training and test subsets. More specifically, for each dataset, the features which are highly correlated with the class but with low intercorrelation are selected.

First, essential features are identified by utilizing the proposed CCFS approach to evaluate the integrity of the reduced feature subset in the feature selection stage. Then, candidate features are selected from the original ones for the next stage.

By implementing CCFS and elimination of the irrelevant features of the dataset and then applying four different classifiers for decision (Figure 2), the performance of the IDS has improved. In order to evaluate the proposed model, the

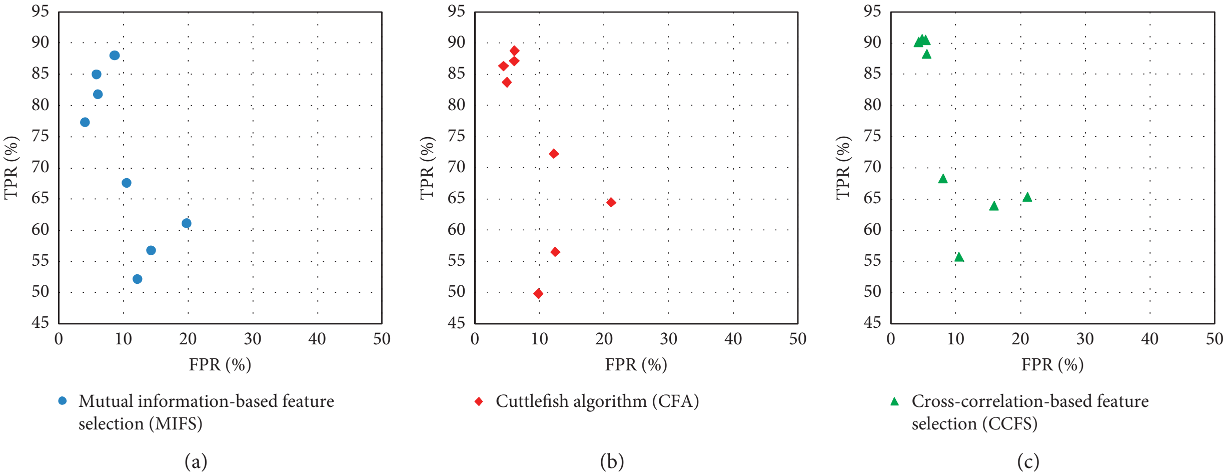


Figure 8: The ROC curve for MIFS, CFA, and CCFS methods with the KNN classifier for the KDD Cup 99 dataset.

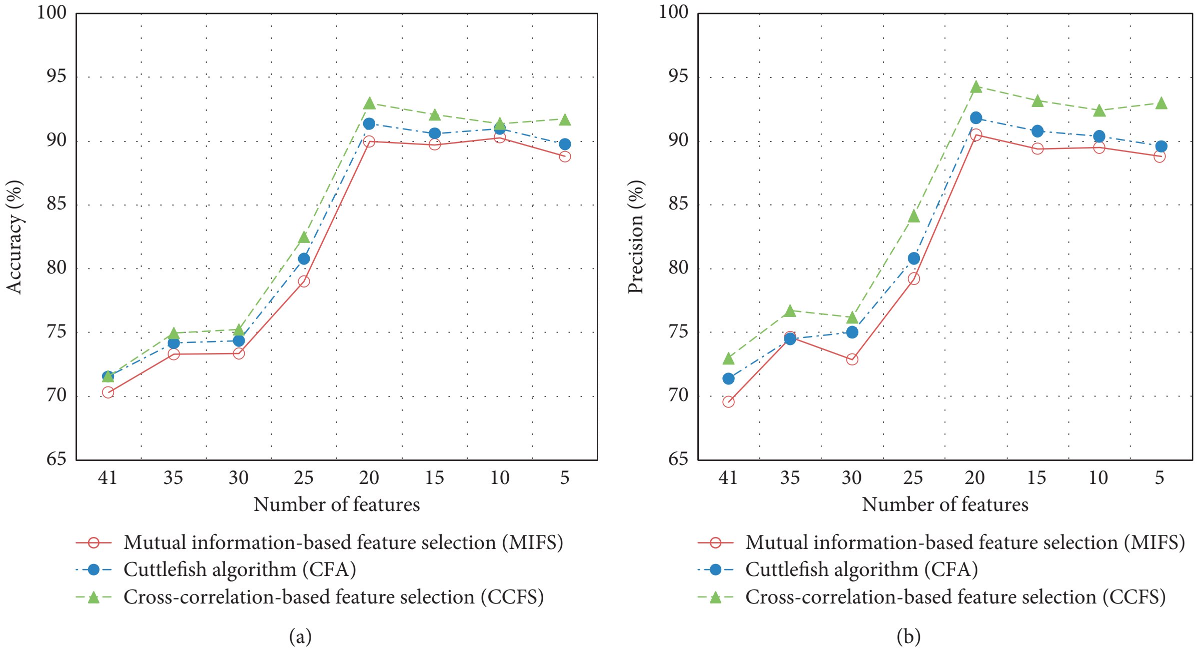
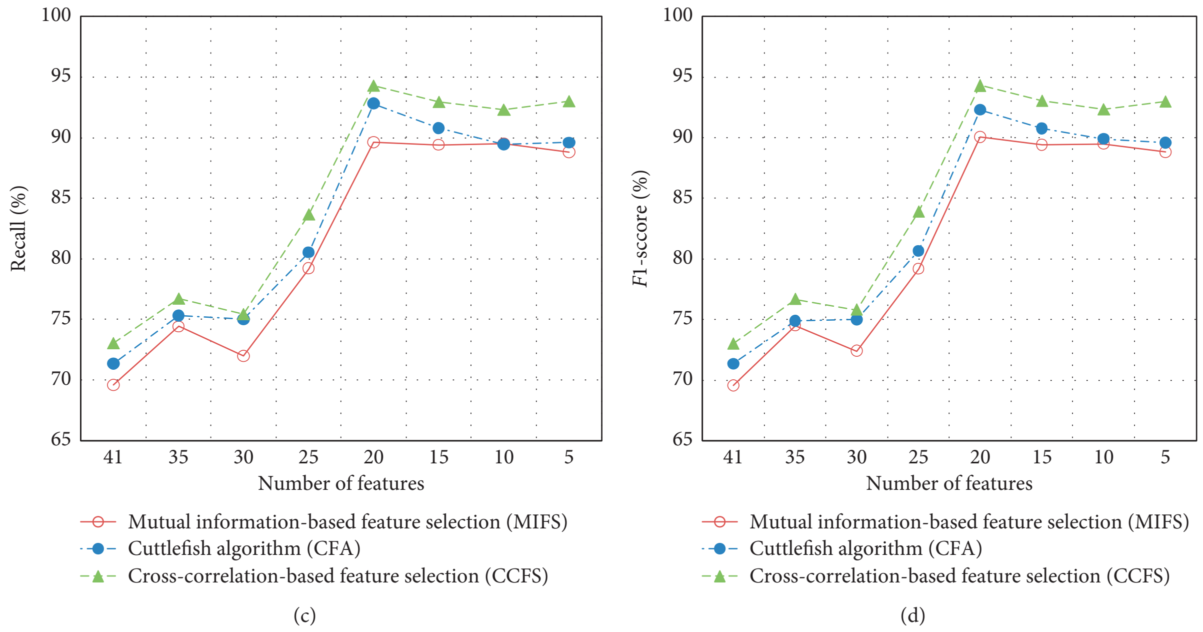


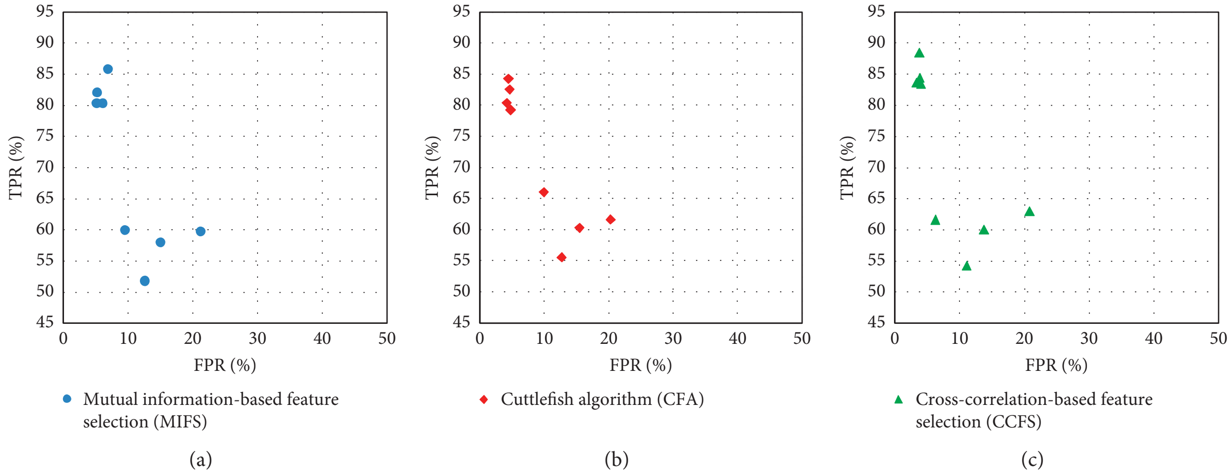
Figure 9: Continued.



(c)

(d)

Figure 9: Performance of the CCFS, MIFS, and CFA methods with the NB classifier for the KDD Cup 99 dataset: (a) accuracy, (b) precision, (c) recall, and (d) $F1$-score.



(a)

(b)

(c)

Figure 10: The ROC curve for MIFS, CFA, and CCFS methods with the NB classifier for the KDD Cup 99 dataset.

Table 2: Improvement percentage of the proposed method (CCFS) in comparison with the CFA method for the DT classifier.

KDD Cup 99				
Number of features	CCFS improvement versus CFA (%)			
	Accuracy	Precision	Recall	$F1$-score
41	1.42	3.40	2.19	2.79
35	2.05	2.61	2.61	2.61
30	0.72	0.82	0.16	0.49
25	1.83	3.70	3.35	3.52
20	0.52	0.99	2.16	1.57
15	0.48	0.13	0.51	0.32
10	0.20	1.62	1.62	1.62
5	1.06	2.50	1.86	2.18

Table 3: Comparison of the MIFS, CFA, and CCFS methods for DT, SVM, KNN, and NB classifiers on the NSL-KDD dataset with 20 features.

NSL-KDD (20 features)												
Classification method	MIFS				CFS				CCFS			
	DT	SVM	KNN	NB	DT	SVM	KNN	NB	DT	SVM	KNN	NB
Accuracy	91.21	87.80	89.79	87.35	93.12	90.15	91.50	89.24	93.47	90.69	93.09	90.87
Precision	93.52	88.17	89.66	89.27	94.82	91.18	92.46	90.34	94.87	92.39	94.39	92.27
Recall	93.45	87.21	89.61	89.27	94.82	91.18	92.46	91.18	95.79	92.12	94.35	92.27
*F*1-score	93.49	87.69	89.64	89.27	94.82	91.18	92.46	90.76	95.33	92.25	94.37	92.27

Table 4: Improvement percentage of the CCFS method over MIFS and CFS methods for DT, SVM, KNN, and ND classifiers on the NSL-KDD dataset.

NSL-KDD (20 features)								
Classification method	CCFS improvement versus MIFS (%)				CCFS improvement versus CFA (%)			
	Accuracy	Precision	Recall	*F*1-score	Accuracy	Precision	Recall	*F*1-score
DT	2.48	1.44	2.50	1.97	0.38	0.05	1.02	0.53
SVM	3.29	4.78	5.63	5.21	0.60	1.33	1.04	1.18
KNN	3.68	5.27	5.29	5.28	1.74	2.09	2.04	2.07
NB	4.03	3.36	3.36	3.36	1.83	2.14	1.20	1.67

Table 5: Comparison of the MIFS, CFA, and CCFS methods for DT, SVM, KNN, and NB classifiers on the AWID dataset with 84 features.

AWID (84 features)												
Classification method	MIFS				CFS				CCFS			
	DT	SVM	KNN	NB	DT	SVM	KNN	NB	DT	SVM	KNN	NB
Accuracy	95.46	95.84	96.32	94.08	97.21	96.02	96.97	94.79	98.42	96.75	97.84	95.53
Precision	95.70	95.87	96.39	94.15	97.61	96.03	96.99	94.95	98.49	96.51	97.87	95.55
Recall	95.96	95.28	96.35	93.47	96.87	96.03	96.10	95.92	98.35	96.51	97.87	96.08
*F*1-score	95.83	95.57	96.37	93.81	97.23	96.03	96.54	95.43	98.42	96.51	97.87	95.81

Table 6: Improvement percentage of the CCFS method over MIFS and CFS methods for DT, SVM, KNN, and ND classifiers on the AWID dataset.

AWID (84 features)								
Classification method	CCFS improvement versus MIFS (%)				CCFS improvement versus CFA (%)			
	Accuracy	Precision	Recall	*F*1-score	Accuracy	Precision	Recall	*F*1-score
DT	3.10	2.92	2.49	2.71	1.24	0.91	1.53	1.22
SVM	0.95	0.66	1.30	0.98	0.76	0.50	0.50	0.50
KNN	1.58	1.53	1.58	1.56	0.90	0.91	1.84	1.38
NB	1.54	1.49	2.79	2.14	0.78	0.63	0.16	0.40

Table 7: Comparison of the MIFS, CFA, and CCFS methods for DT, SVM, KNN, and NB classifiers on the CIC-IDS2017 dataset with 78 features.

CIC-IDS2017 (78 features)												
Classification method	MIFS				CFS				CCFS			
	DT	SVM	KNN	NB	DT	SVM	KNN	NB	DT	SVM	KNN	NB
Accuracy	95.24	94.43	94.82	94.01	96.87	96.02	96.54	94.59	97.91	97.12	97.24	96.32
Precision	95.53	95.12	95.76	94.61	97.84	96.60	97.68	95.06	99.42	98.23	98.75	96.62
Recall	95.53	94.82	95.76	94.61	97.84	97.28	98.02	95.61	99.07	97.94	98.75	96.61
*F*1-score	95.53	94.97	95.76	94.61	97.84	96.94	97.85	95.33	99.24	98.09	98.75	96.62

Table 8: Improvement percentage of the CCFS method over MIFS and CFS methods for DT, SVM, KNN, and ND classifiers on the CIC-IDS2017 dataset.

CIC-IDS2017 (78 features)								
Classification method	CCFS improvement versus MIFS (%)				CCFS improvement versus CFA (%)			
	Accuracy	Precision	Recall	$F1$-score	Accuracy	Precision	Recall	$F1$-score
DT	2.80	4.07	3.70	3.88	1.07	1.61	1.25	1.43
SVM	2.85	3.27	3.29	3.28	1.15	1.69	0.67	1.18
KNN	2.55	3.12	3.12	3.12	0.73	1.09	0.75	0.92
NB	2.46	2.13	2.12	2.13	1.83	1.65	1.05	1.35

Table 9: Results of the CCFS method on different datasets for accuracy, precision, recall, and $F1$-score criteria.

Classification method	KDD Cup 99				NSL-KDD				AWID				CIC-IDS2017			
	DT	SVM	KNN	NB	DT	SVM	KNN	NB	DT	SVM	KNN	NB	DT	SVM	KNN	NB
Accuracy	85.19	84.66	84.66	84.06	93.47	90.69	93.09	90.87	98.42	96.75	97.84	95.53	97.91	97.12	97.24	96.32
Precision	86.45	85.53	85.97	85.37	94.87	92.39	94.39	92.27	98.49	96.51	97.87	95.55	99.42	98.23	98.75	96.62
Recall	86.43	85.63	85.92	85.16	95.79	92.12	94.35	92.27	98.35	96.51	97.87	96.08	99.07	97.94	98.75	96.61
$F1$-score	86.44	85.58	85.95	85.27	95.33	92.25	94.37	92.27	98.42	96.51	97.87	95.81	99.24	98.09	98.75	96.62

CCFS method has been compared with some well-known feature selection methods, namely, MIFS and CFA.

For comparison, accuracy, precision, recall, and $F1$-score metrics are used. The results have shown that the CCFS method outperforms on the MIFS and CFS methods in all criteria on the datasets.

The main reason for better performance of the proposed CCFS method is consideration of correlation between features which helps to better distinguish the attack from normal instances on datasets. Also, according to Figures 3–10 and Table 3, CCFS has a better performance on NSL-KDD than KDD Cup 99.

Table 9, concludes the results of the CCFS method that are obtained for different datasets with DT, SVM, KNN, and NB classifiers. For the KDD Cup 99 dataset, the average of all of features for each criterion is used in this table. According to Table 9, the DT classifier has best results than other classifiers on all datasets.

These results prove the effectiveness of the proposed method in identification of the correlated features. Also, it is clearly seen from Figures 3–10 that using 20 features has better results when compared with using the whole 41 features for the KDD Cup 99 dataset.

In the KDD Cup 99 dataset, the value of FPR is not performed during the simulations. This is because there are some instances of attacks in the test dataset that are never appeared in the training dataset, such as mscan, saint, apache2, mailbomb, processtable, snmpgetattack, and snmpguess.

6. Conclusion

An anomaly may indicate a penetration. To detect anomaly behavior, you must identify normal behaviors and find patterns and rules for them. Behaviors that follow these patterns are normal, and events that are too commonly offset from these patterns are recognized as anomaly behavior.

In this paper, with use of cross-correlation, the features are found which are more robust than other features to intrusion. For evaluation of the proposed method that named CCFS, four KDD Cup 99, NSL-KDD, AWID, and CIC-IDS2017 datasets are used. The evaluation results of the CCFS method are compared with MIFS and CFA methods with DT, SVM, KNN, and NB classifiers. According to the results of the evaluations, the CCFS method has better results in accuracy, precision, recall, and $F1$-score criteria than the other two methods with DT as the best classifier.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The author declares no conflicts of interest.

Acknowledgments

This work was supported by IROST.

References

[1] U. Kanika, "Security of network using IDS and firewall," *International Journal of Scientific and Research Publication*, vol. 3, no. 6, pp. 1–4, 2013.

[2] A. D. Wankhade and Dr P. N. Chatur, "Comparison of firewall and intrusion detection system," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 1, pp. 674–678, 2014.

[3] S. K. Dash, S. Rawat, and A. K. Pujari, "Use of dimensionality reduction for intrusion detection," in *Proceedings of the International Conference Information Systems Security*, Delhi, India, December 2007.

[4] K. Kumar, G. Kumar, and Y. Kumar, "Feature selection approach for intrusion detection system," *International Journal*

Advanced Tren Computer Science Engineering, vol. 2, no. 5, pp. 47–53, 2013.

[5] S.-B. Cho and H.-J. Park, "Efficient anomaly detection by modeling privilege flows using hidden Markov model," *Computers & Security*, vol. 22, no. 1, pp. 45–55, 2003.

[6] S. Chebrolu, A. Abraham, and J. P. Thomas, "Feature deduction and ensemble design of intrusion detection systems," *Computers & Security*, vol. 24, no. 4, pp. 295–307, 2005.

[7] G. Sivasangari and M. Sathya, "A feature selection for intrusion detection system using a hybrid efficient model," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 3, no. 3, pp. 1917–1929, 2018.

[8] S. Parsazad, E. Saboori, and A. Allahyar, "Fast feature reduction in intrusion detection datasets," in *Proceedings of the International Convention MIPRO*, Opatija, Croatia, May 2012.

[9] T. T. H. Le, Y. Kim, and H. Kim, "Network intrusion detection based on novel feature selection model and various recurrent neural networks," *Applied Science*, vol. 9, no. 1392, pp. 1–29, 2019.

[10] S. K. Pandey, "Design and performance analysis of various feature selection methods for anomaly-based techniques in intrusion detection system," *Security and Privacy*, vol. 2, no. e56, pp. 1–14, 2019.

[11] A. Gül and E. Adalı, "A feature selection algorithm for IDS," in *Proceedings of the International Conference Computer Science and Engineering*, Antalya, Turkey, October 2017.

[12] J. Ren, J. Guo, W. Qian, H. Yuan, X. Hao, and H. Jingjing, "Building an effective intrusion detection system by using hybrid data optimization based on machine learning algorithms," *Security and Communication Networks*, vol. 2019, Article ID 7130868, 11 pages, 2019.

[13] S. O. Al-mamory and F. S. Jassim, "On the designing of two grains levels network intrusion detection system," *Karbala International Journal of Modern Science*, vol. 1, no. 1, pp. 15–25, 2015.

[14] N. Das and T. Sarkar, "Survey on host and network based intrusion detection system," *International Journal of Advanced Networking and Applications*, vol. 6, no. 2, pp. 2266–2269, 2014.

[15] A. Yousaf and O. Yousaf, "Intruders and intrusion detection systems - an overview," in *Proceedings of the International Conference Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, Phuket, Thailand, June 2017.

[16] D. Novikov, R. V. Yampolskiy, and L. Reznik, "Anomaly detection based intrusion detection," in *Proceedings of the International Conference Information Technology: New Generations*, Las Vegas, NV, USA, April 2006.

[17] S. A. Agah, "Investigating identification techniques of attacks in intrusion detection systems using data mining algorithms," *International Journal of Computer Science and Network Security*, vol. 17, no. 5, pp. 174–181, 2017.

[18] J. Mchugh, "Intrusion and intrusion detection," *International Journal of Information Security*, vol. 1, no. 1, pp. 14–35, 2001.

[19] M. Uddin, K. Khowaja, and A. Abdul Rehman, "Dynamic multi-layer signature based intrusion detection system using mobile agents," *International Journal of Network Security & Its Applications*, vol. 2, no. 4, pp. 129–141, 2010.

[20] A. Shenfield, D. Day, and A. Ayesh, "Intelligent intrusion detection systems using artificial neural networks," *ICT Express*, vol. 4, no. 2, pp. 95–99, 2018.

[21] J. U. Chibuzor and E. O Bennett, "An intrusion detection system using machine learning algorithm," *International Journal of Computer Science and Mathematical Theory*, vol. 4, no. 1, pp. 39–47, 2018.

[22] M. S. Hoque, Md.A. Mukit, and Md.A. N. Bikas, "An implementation of intrusion detection system using genetic algorithm," *International Journal of Network Security & Its Applications*, vol. 4, no. 2, pp. 109–120, 2012.

[23] A. H. Sung and S. Mukkamala, "Identifying important features for intrusion detection using support vector machines and neural networks," in *Proceedings of the International Symposium on Applications and the Internet*, Orlando, FL, USA, January 2003.

[24] L. A. A. Almeida and J. C. M. Santos, "Evaluating features selection on NSL-KDD data-set to train a support vector machine-based intrusion detection system," in *Proceedings of the IEEE Colombian Conference Applications in Computational Intelligence*, Barranquilla, Colombia, June 2019.

[25] R. R. Devi and M. Abualkibash, "Intrusion detection system classification using different machine learning algorithms on KDD-99 and NSL-KDD datasets-a review paper," *International Journal of Computer Science and Information Technologies*, vol. 11, no. 3, pp. 65–80, 2019.

[26] N. Kausar, B. B. Samir, A. Abdullah et al., "A review of classification approaches using support vector machine in intrusion detection," in *Proceedings of the International Conference Informatics Engineering and Information Science*, Kuala Lumpur, Malaysia, November 2011.

[27] S. Mukherjee and N. Sharma, "Intrusion detection using naive Bayes classifier with feature reduction," *Procedia Technology*, vol. 4, pp. 119–128, 2012.

[28] J. R. Quinlan, *Induction of Decision Trees', Machine Learning*pp. 81–106, Kluwer Academic Publishers, Boston, MA, USA, 1st edition, 1986.

[29] M. Kumar, M. Hanumanthappa, and T. V. Suresh Kumar, "Intrusion detection system using decision tree algorithm," in *Proceedings of the IEEE International Conference Communication Technology*, Chengdu, China, Nov. 2012.

[30] S. V. Lakshmi and T. E. Prabakaran, "Application of k-nearest neighbour classification method for intrusion detection in network data," *International Journal of Computer Applications*, vol. 97, no. 7, pp. 34–37, 2014.

[31] F. Fleuret, "Fast binary feature selection with conditional mutual information," *J. Mach. Learn. Res.* vol. 5, pp. 1531–1555, 2004.

[32] F. Amiri, M. Rezaei Yousefi, C. Lucas, A. Shakery, and N. Yazdani, "Mutual information-based feature selection for intrusion detection systems," *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1184–1199, 2011.

[33] X. Zhang, Z. Zhu, and P. Fan, "Intrusion detection based on cross-correlation of system call sequences," in *Proceedings of the IEEE International Conference Tools with Artificial Intelligence*, Hong Kong, China, November 2005.

[34] X. Zhang, Z. Zhu, P. Fan, and M. He, "Intrusion detection based on SVM and cross-correlation of sample sequences," *Journal of China Railway*, vol. 29, pp. 113–117, 2007.

[35] Y. Zhang, Q. Yang, S. Lambotharan et al., "Anomaly-based network intrusion detection using SVM," in *Proceedings of the International Conference Wireless Communications and Signal Processing*, Xi'an, China, October 2019.

[36] C. Huang, P. Yi, F. Zou, Y. Yao, W. Wang, and T. Zhu, "CCID: cross-correlation identity distinction method for detecting shrew DDoS," *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 6705347, 2019.

[37] V. Jyothsna and V. V. Rama Prasad, *FCAAIS: Anomaly Based Network Intrusion Detection through Feature Correlation*

Analysis and Association Impact Scale, ICT Express, South Korea, 2016.

[38] M. A. Hall, "Correlation-based feature selection for machine learning," Ph. D. thesis, Deparment of Computer Science, University of Waikato, Hillcrest, New Zealand, 1999.

[39] KDD Cup 99-UCI Knowledge Discovery in Databases Archive', http://kdd.ics.uci.edu, October 2018.

[40] S.-J. Horng, M.-Y. Su, Y.-H. Chen et al., "A novel intrusion detection system based on hierarchical clustering and support vector machines," *Expert Systems with Applications*, vol. 38, no. 1, pp. 306–313, 2011.

[41] A. S. Eesa, Z. Orman, and A. M. A. Brifcani, "A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems," *Expert Systems with Applications*, vol. 42, no. 5, pp. 2670–2679, 2015.

[42] S. Mukkamala, A. H. Sung, and A. Abraham, "Intrusion detection using an ensemble of intelligent paradigms," *Journal of Network and Computer Applications*, vol. 28, no. 2, pp. 167–182, 2005.

[43] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Ottawa, Canada, July 2009.

[44] 'Nsl-kdd Data Set for Network-Based Intrusion Detection Systems', http://nsl.cs.unb.ca/NSL-KDD/, March 2009, accessed June 2020.

[45] C. Kolias, G. Kambourakis, A. Stavrou, and S. Gritzalis, "Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset," *IEEE Communications Surveys & Tutorials*, vol. 18, pp. 184–208, 2015.

[46] 'AWID. Aegean Wireless Intrusion Dataset', http://icsdweb.aegean.gr/awid/download.html, November 2014, accessed June 2020.

[47] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proceedings of the International Conference Information Systems Security and Privacy*, Portugal, January 2018.

[48] 'CIC-IDS. Canadian Institute for Cybersecurity Dataset', https://www.unb.ca/cic/datasets/ids-2017.html, 2017.

[49] J. Li, K. Cheng, S. Wang et al., "Feature selection: a data perspective," *ACM Computing Surveys*, vol. 50, p. 94, 2017.

[50] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Data preprocessing for supervised leaning," *International Journal of Computer Science*, vol. 1, pp. 111–117, 2006.

[51] C. Iwendi, S. Khan, J. H. Anajemba, M. Mittal, M. Alenezi, and M. Alazab, "The use of ensemble models for multiple class and binary class classification for improving intrusion detection systems," *Sensors*, vol. 20, no. 9, p. 2559, 2020.