The ISC Int'l Journal of Information Security

July 2022, Volume 14, Number 2 (pp. 115–130)

http://www.isecure-journal.org

A Graph-based Online Feature Selection to Improve Detection of New Attacks

Hajar Dastanpour [1], and Ali Fanian [2,*]

[1] *Isfahan University of Technology, Isfahan, Iran.*

[2] *Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran.*

ARTICLE INFO.

Article history:
Received: December 17, 2020
Revised: May 23, 2021
Accepted: January 8, 2022
Published Online: January 10, 2022

Keywords:
Classification, Clustering, Ensemble Clustering, Intrusion Detection System, Online Feature Selection

Type: Research Article

doi: 10.22042/ISECURE.2022.262485.590

dor: 20.1001.1.20082045.2022.14.2.1.3

ABSTRACT

Today, intrusion detection systems are used in the networks as one of the essential methods to detect new attacks. Usually, these systems deal with a broad set of data and many features. Therefore, selecting proper features and benefitting from previously learned knowledge is suitable for efficiently detecting new attacks. A new graph-based method for online feature selection is proposed in this article to increase the accuracy in detecting attacks. In the proposed method, irrelevant features are first removed by inputting a limited number of instances. Then, features are clustered based on graph theory to reduce the search space. After the arrival of new instances at each stage, new clusters of features are created that may differ from the clusters created in the previous step. Therefore, to find the appropriate clusters, these two clusters are combined to select some relevant features with minimum redundancy. The evaluation results show that the proposed method has better performance, for instance classification with a lesser run time than similar online feature selection methods. The proposed method is also faster with a suitable accuracy in instances classification compared to some offline methods

1 Introduction

The rapid growth of the Internet and computer networks has made security a significant issue. The security of computer systems includes confidentiality, integrity, and availability. An attempt to intrusion into a computer network jeopardizes at least one of the security parameters. Today, to ensure security in computer networks, different security systems and devices are used, including intrusion detection systems (IDS) [1]. The IDS is a software or hardware system or a combination of the two, and it is responsible for distinguishing authorized users from the infiltrators. An IDS aims not to prevent an attack, but its only task is to detect attacks and security bugs in the system and notify the system administrator.

Intrusion detection systems are classified as either signature-based or anomaly-based [2–5]. Signature-based schemes (also denoted as misuse-based) seek defined patterns, or signatures, within the analyzed data including network packets, log messages, system calls, registry access traces and so on. For this purpose, a signature database corresponding to known attacks is specified a priori [5].

On the other hand, anomaly-based detectors attempt to estimate the "normal" behavior of the system to be protected and generate an anomaly alarm

* Corresponding author.
Email addresses: dastanpoor_h@yahoo.com, a.fanian@iut.ac.ir

ISSN: 2008-2045

whenever the deviation between a given observation at an instant and the normal behavior exceeds a predefined threshold. Another possibility is to model the "abnormal" behavior of the system and to raise the alarm when the difference between the observed behavior and the expected one falls below a given limit. Signature and anomaly-based systems are similar in terms of conceptual operation and composition. The main differences between these methodologies are inherent in the concepts of attack and anomaly. An attack can be defined as a sequence of operations that puts the security of a system at risk. An anomaly is just an event that is suspicious from the perspective of security. Signature-based schemes provide very good detection results for specified, well-known attacks. However, they cannot detect new, unfamiliar intrusions, even if they are built as minimum variants of already known attacks [6–8]. On the contrary, the main benefit of anomaly-based detection techniques is their potential to detect previously unseen intrusion events [6].

One of the vital solutions to detect anomalies in the network is based on machine learning techniques [9–11]. These schemes are based on establishing an explicit or implicit model that enables the patterns analyzed to be categorized. In many cases, the applicability of machine learning principles coincides with that for the statistical techniques. However, the former is focused on building a model that improves its performance based on previous results. Hence, a machine learning-based anomaly detection system can change its execution strategy as it acquires new information. Although this feature could make it desirable to use such schemes for all situations, the major drawback is their expensive resource nature. Machine learning techniques are divided into two categories, incremental and non-incremental.

An incremental method's main task is to construct or model the pattern classes and use them to correct the input pattern as either belonging to a specific previously observed class or not belonging to any of the preferred classes. However, for network anomaly detection, the incremental approach updates normal profiles dynamically based on changes in network traffic without fresh training using all data for attack detection. It can decide to raise an attack based on existing profiles if abrupt changes happen in the normal system or network traffic. Thus, it is useful to detect anomalies from a normal system or network traffic from existing signatures or profiles, using an incremental approach. An incremental approach that updates profiles or signatures dynamically, incorporating new profiles as it encounters them. It does not need to load the whole database each time into the memory and learn fresh from the beginning [12].

We observe that the main importance of incremental network anomaly detection lies in dynamic profile updation for both normal and attack, reduced memory utilization, faster and higher detection rate, and improved real-time performance.

The number of features extracted from raw network data, which an IDS needs to examine, is usually large even for a small network [13]. Many researchers have tried to improve the detection rate of IDs by proposing new classifiers, but improving classifier's effectiveness is not an easy task, though feature selection can be used to optimize the existing classifiers. The purpose of feature selection is to acquire a subset of features with the least number of features with higher accuracy than other subset ones [14].

Feature selection plays a vital role in detecting network anomalies. Feature selection methods are used in the intrusion detection domain for eliminating redundant or irrelevant features. Feature selection reduces computational complexity, removes information redundancy, increases the detection algorithm's accuracy, facilitates data understanding, and improves generalization. The feature selection is a Machine Learning technique that reduces the amount of data to be analyzed. Therefore, the feature selection is accomplished by identifying the most important features (or attributes) of a data set and discarding the less important ones. Machine Learning algorithms can make classification predictions more efficient by reducing the dimensionality of a data set to contain only the most essential features. This efficiency is especially relevant to Intrusion Detection, which has demanded real-time performance.

One reason why fewer features can sometimes improve classification performance because some features may contain meaningless noise, so that does not contribute any value to predictive accuracy. The presence of these noisy features can sometimes even degrade classification performance. Another reason why reducing the number of features can improve classification performance is that different features from within the same data set can be highly correlated. The presence of these extraneous features can sometimes cause more confusion than predictive value to classification models. Feature selection can be a valuable technique for Intrusion Detection where real-time computational demands can benefit from improved efficiency while striving to maintain similar or even improved classification performance [13–15].

According to the definitions [14], the features are generally categorized as follows:

Relevant: The features that affect the recognition of instances' class and the ignoring them cause reduced recognition accuracy.

Irrelevant: Features that do not affect in-class recognition of instances.

Redundant: Redundancy is created whenever a feature can have the role of another one. In other words, a feature is highly correlated with one or more other features.

In general, feature selection methods are divided into offline and online. The former is called the traditional feature selection method [16–19]. In this method, all features and instances are available before triggering the feature selection process. Offline methods are not appropriate in many areas, such as image processing and online learning, since the entrance of features or instances is dynamic. Therefore, it is necessary to use an online feature selection in such scenarios. The latter one is called online feature selection, which is defined in two ways: in the first, the number of instances is fixed, features are entered dynamically, and selection is made at the time of the feature entry, and in the second, the number of features is fixed, instances are entered dynamically, and feature selection is performed with the entry of each instance.

Many offline methods have been proposed in the field of intrusion detection, though the use of offline feature selection methods in intrusion detection systems has some limitations. As previously stated, offline feature selection methods require the entire training set, necessitating a vast memory capacity. In computer networks, network packet flows (instances) are entered dynamically and usually are used from an incremental method for intrusion detection; hence, not all network flows are available at the time of feature selection. Also, the cost of waiting for the entrance of all flows will be high; thus, offline methods are not suitable for IDSs. As new attacks occur continuously and dynamically in computer networks, IDSs should adapt themselves to new conditions and use past experiences.

In a limited number of online methods, feature selection is performed, at an instance, arrival time. Such methods, however, do not examine redundancy between features. Also, these methods are suitable for a dataset with two classes only. Even so, most datasets, including network traffic, has more than two classes. In most online methods, feature selection is performed by the dynamic entry of features; thus, these methods are not suitable for IDSs because instances are dynamically entered in these systems, and the features are fixed. According to the literature, online selection methods have not been used in IDSs so far. In order to overcome the mentioned constraints, an online approach can be opted to select features. Thereby, the entrance of new flows into the system can lead to selecting appropriate features besides using previously gained knowledge. Here, a graph-based new method with these properties is presented for an online feature selection. Our proposed method assumes that the number of features is fixed, and feature selection occurs when network flows arrive.

The evaluation results show that the proposed method has a better performance than other methods of online feature selection. In other words, the proposed method with a low runtime selects features that increase the accuracy of instance categorization compared to other online methods. Also, the proposed method is faster than offline methods rendering an acceptable accuracy in the classification of instances.

This study first reviews previous related work. Subsequently, the proposed method of feature selection is explained in the third section. Then, the proposed method results are presented and compared with some of the studies carried out in Section IV, followed by a conclusion.

2 Related Work

Most existing feature selection methods and the proposed method here use the correlation criterion to select features. Therefore, a brief description is first offered on the crucial criterion of mutual information (MI) [15, 20]. MI is the most popular nonlinear correlation criterion based on information theory. This criterion is shown as $I(X;Y)$ for the two random variables of X and Y, which has a symmetry property ($I(X;Y) = I(Y;X)$). This criterion represents the amount of information in a random variable concerning another random variable [15, 20]. In other words, this criterion indicates the amount of information shared between two variables. MI is zero if and only if X and Y are independent, meaning that X contains no information about Y. The larger the MI is, the more dependent two variables will be. This means that X gives a lot of information about Y; thus, there is a high similarity between X and Y. MI is obtained using the concept of entropy. As mentioned in the first section, feature selection methods are divided into offline and online categories. In this section, the most critical research is reviewed in both online and offline methods.

Since 1970, many offline feature selection methods have been proposed, divided into three categories: supervised, unsupervised, and semi-supervised [16, 21]. In a supervised feature selection method, features are selected based on the labeled training instance. Based on different evaluation criteria, supervised selection methods fall into three groups: filter, wrapper, and embedded [14]. No method of learning is used in filtering methods to remove irrelevant and redundant

features. Instead, these methods use the characteristics of training instances to evaluate the features and identify relevant features. In this regard, the criteria of distance, correlation, consistency, and information theory are used. Filtering methods are also called ranking methods because they are an appropriate measure for ranking the features.

Filtering methods are suited for high-dimensional data sets and are computationally fast and straightforward. These methods are also independent of the classification method. Therefore, selected features can be appropriate for any learning algorithm [14, 22].

Graph theory has recently been used in some of the filter feature selection methods [23, 24] in which a fully-weighted and undirected graph is constructed using features. Each node in the graph corresponds to a feature, and the weight of the graph edges shows the similarity of features.

In the Wrapper method, the features are selected according to the performance of the learning algorithms [25, 26]. In other words, a subset of features is chosen if they reduce the error of the learning algorithm. Thus, Wrapper methods often yield better results than the filter method, though, these methods need more computational time and resources.

Unlike filter and wrapper methods, the embedded method performs learning and feature selection together [27]. The learning of the decision tree is an example of an embedded method.

Unsupervised methods are used when no labeling instances are available. The variance score method is one of the most natural unsupervised feature selection methods selecting features with the highest variance [28]. The semi-supervised feature selection method has been presented in recent years, in which both labeled and unlabeled instances are available [29, 30].

In most online methods, the selection of features is performed by the dynamic entry of the features [28]. As mentioned above, the instances are constant in some applications, and the features are dynamically entered into the learning model. Therefore, it is necessary to carry out an online feature selection. Other approaches, including grafting [30], alpha-investing [33], OSFS [34], and SAOLA [35], are the most important methods suggested for this purpose. As a result, these methods are not suitable for IDSs because, in these systems, the instances are entered dynamically, and the features are fixed. Recently, a limited number of online methods have been introduced, namely online AdaBoost [33], OFS [37], and RFOFS [38] online feature selection for IDS [39, 40] in which feature selection is performed at the entry time of instances. The most important limitation of these methods is that they are only suitable for a dataset with two classes. At the same time, some datasets, including network traffic, have more than two classes, and attack classes are specified separately. Besides, redundancy between features is not explored in these methods.

The OFS online feature selection method is suggested for the binary classification, in which feature selection is conducted at the instance entrance. Suppose that at t stage, the learner receives input (x_t, y_t). Each instance $x_t \in \mathbb{R}^d$ is a d-dimensional vector with the label $y_t \in \{-1, 1\}$. The learner at t stage presents a classifier $w_t \in \mathbb{R}^d$ which will be used to classify instance x_t by a linear function $sgn(w_t^T x_t)$. To select a feature B, the classifierw_t needs to have the highest B non-zero elements, that is, $B > 0$ specified by the user.

The main idea in the OFS method is that if the instance x_t is not correctly classified ($y_t w_t^T x_t \leq 1$), the classifier is updated using the online gradient descent according to Relation (1). Subsequently, to ensure that the values of the deleted features are small enough, the classifier is updated using Relation (2). Finally, B number of the largest elements in the classifier is selected, and the rest of the elements are removed (zero). In other words, B numbers are selected from the relevant features [37].

$$\widetilde{w}_{t+1} = (1 - \lambda\eta)w_t + \eta x_t y_t \tag{1}$$

$$\widehat{w}_{t+1} = min\{1, \frac{\frac{1}{\sqrt{\lambda}}}{||\widehat{w}_{t+1}||_2}\}\widetilde{w}_{t+1} \tag{2}$$

In the OFS method, some features are not selected during the feature selection process, while features to be deleted may be selected after the entrance of a new instance. In this case, only a new instance is effective in choosing this feature, and it may result in a classifier error. The RFOFS method is, therefore, proposed to eliminate the objection expressed in [38]. The RFOFS method is the same as the OFS method, with the difference that the former uses two classifiers w_t and w_all_t . If the x_t instance is not properly classified, w_all_t will be used to update the classifiers. It also stores deleted features. The w_t classifier is used to select features and to predict the class of instances.

Although OFS and RFOFS methods are less costly than the online AdaBoost method, the redundancy between the features is not checked in both procedures. Also, both are not very efficient for a dataset with more classes.

The method presented in [39] first introduces a semi-supervised anomaly detection system that uses both clustering and decision tree methods to identify anomalies. In order to classify new instances, they

firstly clustered using an online clustering method. Each cluster is assigned a normal or anomaly label using first an unsupervised and then a supervised method. The supervised method does not need the labeling of all instances in the dataset. It requires the label of the clusters, which drastically reduces the number of instances to be labeled. The provided labels are used to learn a decision tree based on the cluster labels. Cluster-based features that depend on the original features and the relationships between clusters are derived for the decision tree training. This paper includes an online mutual information-based feature selection method that selects the most relevant cluster-based features.

In order to make online feature selection, compute the counts of features and label values for the past data and update those counts as new data instances arrive. Therefore, in this paper, compute the feature relevance anytime based on the counts that are kept. As it has been the practice in many online methods such as [41], the effect of the old instances can be reduced exponentially by updating the counts. To this end, the old count is multiplied by a parameter between zero and one. It is also possible to give different weights to different classes by updating the counts proportional to the desired class weights. This allows classes with a small number of instances to be represented in feature selection and is therefore much more efficient than sampling-based methods that the aim is to solve the class imbalance problem [39].

In [40], presented a simple technique based on incremental learning of support vector machines to rank the features in real-time within a streaming model for the network data. This method, like the method presented in [39], is only suitable for datasets with two classes, and the redundancy between features is not calculated.

3 The Proposed Method

In this section, a graph-based online feature selection method is proposed to increase the accuracy of detecting attacks in IDSs. The proposed method can resolve the limitations of other methods mentioned earlier. The proposed model is run in four stages: removing irrelevant features, clustering of features, clustering ensemble, and feature selection. In the following, at first, the architecture of the proposed method and then the steps of the method are described.

3.1 Architecture of the Proposed Method

When the network packet flow is entered in the proposed method, a proper subset of the features is selected online. The proposed method can work in unsupervised or supervised modes. We assume that

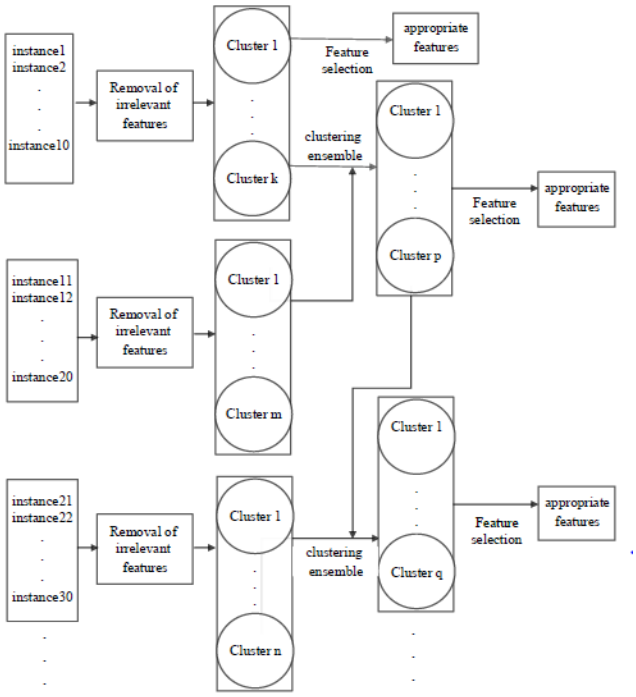

Figure 1. Architecture of proposed method

no classification algorithm is used to select features. Therefore, the proposed method is an online and filter-based feature selection method that can be useful in an anomaly-based intrusion detection system.

Figure 1 shows the proposed method's overall architecture, and the proposed algorithm is presented in Algorithm 1. As shown in Figure 1, the irrelevant features are removed with the entry of a limited number of instances, e.g., N instances (lines 4-12 in Algorithm 1). For simplicity, in Figure 1, the value of N is assumed 10. It should be noted that this step will reduce the runtime in the following steps. The number of features decreases after eliminating irrelevant features. Therefore, the number of possible states to search for the best subset of features also decreases, which results in reduced runtime.

In the next step, as depicted in lines 14-25, the relevant features are clustered according to the graph theory, and a number of features are selected as representatives from each cluster. Since the features of each cluster are similar, there is redundancy among them. The advantage of clustering features is that there is no need to search the space of all features meaning that it is unnecessary to examine all the subsets of features. This leads to reducing in search time as compared to offline methods. In this article, the clustering section is used in addition to the Euclidean distance criterion from the mutual information criteria for weighing graph edges.

As shown in Figure 1, the values of features and

Algorithm 1 Algorithm of the proposed method

Input: q: number of features.
Output: S: selected feature subset

1: Out=∅; //Out is consist of final clusters.
2: $\acute{F}$=∅; //$\acute{F}$ is consist of relevant Features.
3: NumberClustering=0; //number of clustering
4: **for** $t = 1 \rightarrow T$ **do**
5: Receive $D_t = (X_1, X_2, \ldots, X_N, C)$; //$X_i = (F_1, F_2, \ldots, F_m, c), c \in C = \{c_1, c_2, \ldots, c_N\}$, $1 \leq i \leq N$.
6: // Part1: Irrelevant Feature Removal
7: **for** $i = 1 \rightarrow m$ **do**
8: relevant=@rel() //functions that compute the relevance feature F_i
9: **if** $relevant \neq 0$ **then**
10: $\acute{F} = \acute{F} \cup \{F_i\}$;
11: **end if**
12: **end for**
13: k=size ($\acute{F}$);
14: // Part2: Clustering Relevant Feature.
15: // Constructing Compelete Graph G ($\acute{F}$,E).
16: **for** $i = 1 \rightarrow k$ **do**
17: **for** $j = 1 \rightarrow k$ **do**
18: **if** i ≠j **then**
19: $weight(\acute{F}_i, \acute{F}_j) = \frac{1}{I(\acute{F}_i, \acute{F}_j)}$;
20: **else**
21: $weight(\acute{F}_i, \acute{F}_j) = 0$
22: **end if**
23: **end for**
24: **end for**
25: Cluster=Cluster-Graph (G($\acute{F}$, E));
26: NumberClustering=NumberClustering+1;
27: //ClusteringEnsemble
28: **if** NumberClustering==1 **then**
29: Out=Cluster;
30: **else**
31: Out=Cluster-Ensemble (Out, Cluster);
32: NumberClustering=1;
33: **end if**
34: p=size (Out);//Number of final clusters, Out=$\{cl_1, cl_2, \ldots, cl_p\}$.
35: //Part4: Feature Selection
36: $S = \emptyset$;
37: **for** $i = 1 \rightarrow p$ **do**
38: $R = argmax_{\acute{F}_r \in cl_i}(@rel)$;
39: $S = S \cup \{R\}$;
40: **for** $j = 1 \rightarrow size(cl_i)$ **do**
41: $F_{best} = argmax_{\acute{F}_t \in \{cl_i - S\}}(@rel() - \frac{1}{|S|} \sum_{\acute{F}_r \in S} \frac{I(\acute{F}_r, \acute{F}_t)}{H(\acute{F}_r)})$;
42: $S = S \cup \{F_{best}\}$;
43: **if** —S—==q **then**
44: i=p+1;
45: **end if**
46: **end for**
47: **end for**
48: **end for**

then probably the clusters' structures may change with the entry of the next N instances. So, the past experiences are used to find the appropriate clusters, those who properly group the features. To this end, a clustering ensemble is used to combine existing clusters along with the previous clusters. This makes better features are selected during the online feature selection phase. Thus, the selected subset of features in each step may be different from those of the previous stage. In other words, the proposed method has no problems with the nesting effect. At each stage, appropriate clusters are stored to combine with new clusters at a later stage. Therefore, there is no need for each step to store and combine the clusters created in all previous steps. It is worth mentioning that the idea of clustering ensemble is not used in any of the feature selection methods.

To follow the process, a number of features are selected from appropriate clusters with maximum effect in recognition of instances class and minimum redundancy. In this step, the redundancy between the features is examined and calculates the value of relationships (correlations) between features and classes. Besides, it is possible to calculate the relationship of each feature with any number of classes. As a result, unlike the other similar online methods, the proposed method is suitable for datasets with any number of classes.

By the entrance of new instances, the expressed steps are repeated to select appropriate features (in Algorithm 1, for example, time T is repeated). In the following, each part of the proposed model is described in more detail.

3.2 Removal of Irrelevant Features

To remove irrelevant features, it is necessary to calculate the relevance of each feature. We use from function @rel() to compute the relevance of each feature. If we use a supervised feature selection, the @rel() function must then make use of the class labels.

Mutual information is one of the important correlation criteria that can be used in this regard.

Suppose that the D dataset contains the set of the feature $F = \{F_1, F_2, \ldots, F_m\}$ and the category set $C = \{c_1, c_2, \ldots, c_N\}$ (assuming that the number of instances and features is equal to N and m, respectively). To remove the irrelevant features in supervised feature selection, the value of $@rel() = I(F_i, C)$ is calculated for each vector of the feature F_i ($1 \leq i \leq m$) , and the set of category C. For example, for N instances, the values of the vector feature F_i is $F_i = \{f_1, f_2, \ldots, f_N\}$. Now, if $I(F_i, C)$ is zero, then the feature F_i is considered as an irrelevant feature, and thus it is deleted as depicted in lines 7-12 in the algorithm.

Category c can have d different values ($d \geq 2$). Therefore, the proposed method is suitable for datasets with any number of categories. After removing irrelevant features, a set of relevant features will remain, such as $\acute{F} = \{\acute{F}_1, \acute{F}_2, \ldots, \acute{F}_k\}$ for $1 \leq k \leq m$.

Afterward, the subset of features is selected from relevant features with minimal redundancy.

An Efficient relevance measure that can be used for unsupervised Feature selection is the term-variance (TV) method [36]. The term-variance method ranks feature F_i by their sample variance, given by

$$@rel() = TV_i = var_i = \frac{1}{N} = \sum_{j=1}^{N}(F_{ij} - \bar{F}_t)^2, \quad (3)$$

where $\bar{F}_t = (\frac{1}{N})\sum_{j=1}^{N} F_{ij}$ is the average value of feature , and N is the number of instances. Features with higher variance are more informative than features with lower variance.

3.3 Clustering Features

Clustering is one of the most important issues in unsupervised learning widely used in various fields, such as machine learning and data mining. A cluster contains a set of data that is based on their similarity in a group.

Data clustering attempts to divide data into clusters in such a way that maximizes the similarity between data within each cluster and minimize the similarity between data in different clusters. In clustering, it is assumed that primary knowledge, such as knowing the number of clusters, and how to distribute data values, is not available [40, 41].

The difference in data distribution makes the created clusters have arbitrary shapes, different densities, and unbalanced sizes. A number of methods consider the parameter of cluster numbers and data distribution as the basic information. In this article, the graph-based clustering technique is used based on the minimum spanning tree (MST) [42]. An advantage of this method is that it employed automatically specifies the number of categories with a correct clustering. It is also useful for datasets with different distributions.

The clustering algorithm serves to divide the obtained graph from the relevant features into a number of sub-graphs based on a separation criterion. The separation criterion determines which of the edges in the graph are the inconsistent edges and have to be deleted to cluster the features properly.

The inconsistent edges in the graph are those that should be removed to make clusters (sub-graphs) with the correct shape. In other words, such edges are those that connect the dataset of different categories and are the best choice to be removed during the clustering procedure. Inconsistent edges are also known as outside edges.

This article reports clustering of relevant features using a set F' (relevant features) to construct a weighted, undirected and complete graph such as $G = (V, E)$ in which $F' = \{F'_1, F'_2, \ldots, F'_k\}$ represents the set of vertices of the graph, and $E = \left\{(\acute{F}_i, \acute{F}_j) | \acute{F}_i, \acute{F}_j \in \acute{F} \wedge i, j \in [1, k] \wedge i \neq j\right\}$. Each edge, $e = (\acute{F}_i, \acute{F}_j)$, in the graph G has a length (weight) of $weight(\acute{F}_i, \acute{F}_j)$. After creating the graph from the relevant set of features, this graph's first and second minimum spanning tree is created. Then the graph is constructed based on the combination of the first and second minimum spanning trees. Removing inconsistent edges in this graph leads to a proper clustering of features. Figure 2, sections a and b, respectively, show the first and second minimum spanning trees. The graph based on the combination of the first and second minimum spanning tree is shown in Figure 2, section c, d. In these two figures, removal of the edges of ab and ac yields a valid separation. Therefore, these edges are known as inconsistent edges.

In the proposed method, two Euclidean distance and mutual information criteria are used for edge weights. The lesser the distance between two features indicates the more similarity of the two features, and they would perch in the same cluster. However, two features have more mutual information; they will be more similar to each other. The advantage of clustering features is that there is no need to search the space of all features. In other words, it is not necessary to examine all the subsets of features. This leads to reducing in search time as compared to offline methods. Given that the minimum spanning tree is used in the clustering, smaller weights represent more similarity. Therefore, inverse mutual information is used for weighing the edges (lines 15-24 of Algorithm 1) in the proposed method. Calculating the weights of the edges based on Euclidean distance and mutual information are shown in (4) and (5), respectively, where nd are two features vectors with length N (assuming that at each step the number of instances is equal to N):

$$weight1(\acute{F}_i, \acute{F}_j) = \sqrt{\sum_{u=1}^{N}(\acute{F}_{iu} - \acute{F}_{ju})^2}, \quad (4)$$

$$weight2(\acute{F}_i, \acute{F}_j) = \frac{1}{I(\acute{F}_i, \acute{F}_j)}. \quad (5)$$

3.4 Clustering Ensemble

One of the critical challenges in clustering algorithms is to find a powerful solution for different clustering types of datasets. So far, there is no algorithm that can generate the best cluster for all datasets. On the other hand, finding an appropriate cluster-

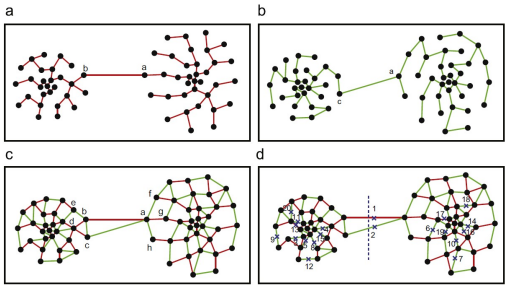

Figure 2. Features clustering: a) first minimum spanning tree, b) second minimum spanning tree, c) graph based on the combination of the first and second spanning minimum, and d) graph cutting using the line curve [41]

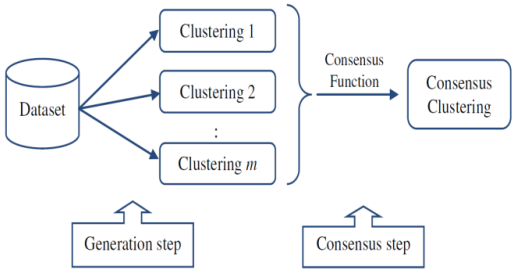

Figure 3. The overall process of clustering ensemble

ing algorithm for each dataset requires specialty and experience [43]. In other words, finding the best similarity criterion, the best clustering method, and the best input parameters of the clustering algorithm is impossible in most cases.

Clustering ensemble is one of the most important solutions to improve clustering performance, which combines multi-partition (clustering) results from data to produce a better performance partition. In this article, the clustering ensemble is used to improve the clustering functionality of features. Assume that F is a set of data, then a partition of F is defined as $\pi_i(F) = \{C_1, C_2, \ldots, C_{|\pi(F)|}\}$, where C_i is a cluster of $\pi(F)$ with the following conditions:

$$C_i \cap C_j = \phi,\ F = \bigcup C_i, 1 \leq i, j \leq |\pi(F)|,\ i \neq j.$$

For a data set F, a set of partitions in the form $\Pi(F) = \{\pi_1(F), \pi_2(F), \ldots, \pi_{|\Pi(F)|}(F)\}$ is defined as $\pi_i(F) = \{C_{i1}, C_{i2}, \ldots, C_{i|\pi_i(F)|}\},\ 1 \leq i \leq |\Pi(F)|$ Besides, a set containing clusters of all partitions is defined as $C(F) = \{C_{ik} | C_{ik} \in \pi_i(F),\ \forall \pi_i(F) \in \Pi(F)\}$.

Clustering ensemble problem is defined as finding a new final clustering

$$\pi^*(F) = \{C_1^*(F), C_2^*(F), \ldots, C_{|\pi^*(F)|}^*\}.$$

By using the information obtained from $\Pi(F)$ such that $\forall i\ \phi(\pi^*(F)) \geq \phi(\pi_i(F)),\ 1 \leq i \leq |\Pi(F)|$.

where the function ϕ is a criterion for evaluating the quality of partitions [47].

As shown in Figure 3, the clustering ensemble is a two-step process, which initially generates different partitions from the dataset. In the second step, an agreement function is used according to definition 1 to combine partitions, which results is a suitable new partition. It should be noted that the clustering ensemble has the following benefits as compared to single clustering algorithms.

Robustness: Better performance in data clustering.

Novelty: Finding clusters that cannot be achieved by any single clustering algorithm.

Stability: Less sensitivity to noise, outlier data, and variability of data.

In this article, the COMUSACL-DEW clustering method is used to find suitable clusters, which is a graph-based method working at the cluster level (graph vertices are the clusters) [44].

If F is a set of data and C is a set of input clusters, then the number of graph vertices in the COMUSACL-DEW method decreases from $|F|$ to $|C|$ with $|C|$ being smaller than $|F|$ (compared to data-level methods in which data are the vertices of the graph).

As a result, the number of edges in the worst case (if the graph is complete) decreases from $\frac{|F|^2-|F|}{2}$ to $\frac{|C|^2-|C|}{2}$, leading to a significant reduction of time.

Besides, the COMUSACL-DEW method decides to expand a cluster just by examining the neighbors of one vertex. Therefore, no additional operations are performed along, and this causes to improve runtime. It is also resistant to noise and outlier data and is suitable for large datasets with different shapes. Another advantage of this method is that it automatically determines the number of final clusters [44]. Lines 31-34 in Algorithm 1 perform clustering ensemble.

3.5 Feature Selection

In the proposed method, as depicted in lines 37-42, a subset of the relevant features with minimum redundancy is selected by entering the network packets flow after creating the appropriate clusters. Given that the features within each cluster are similar (there is redundancy among them), it is sufficient to select a number of features as a representative from each cluster. Therefore, in the proposed method, instead of searching for the entire feature space, the search is performed within each cluster, leading to a reduction in the runtime compared with the offline methods.

In the proposed method, a subset of features such

as S ($|S| = q < k$) is selected to maximize the corresponding evaluation function. It should be noted that k is the number of relevant features, and q is the number of selected features specified by the user.

In the proposed method as depicted in line 41 of Algorithm 1, Equation (6) is used to evaluate the features, where the variable S represents the set of selected features, and $\acute{F}_r$, $\acute{F}_t$ are vectors with length N, representing the selected feature, the evaluated feature, respectively.

$$m = (@rel)\frac{1}{S}\sum_{\acute{F}_r \in S}\frac{I(\acute{F}_r;\acute{F}_t)}{H(\acute{F}_r)} \tag{6}$$

As discussed above, the $@rel()$ function does not take into consideration the redundancy among the selected features. To address this issue, a scheme is proposed to eliminate redundancy among the selected features based on mutual information and appended to this function that is shown in (6), which, for a vertex such as $\acute{F}_t$, the degree of redundancy of each feature is defined as $\frac{1}{S}\sum_{\acute{F}_r \in S}\frac{I(\acute{F}_r;\acute{F}_t)}{H(\acute{F}_r)}$.

To ensure the values of $@rel()$ function and mutual information do not vary greatly, both values are adapted to the range [0,1] [36]. in supervised feature selection $@rel() = \frac{I(\acute{F}_t;C)}{H(\acute{F}_t)+H(C)}$ and in supervised feature selection $@rel() = Var(\acute{F}_t)$.

An advantage of Equation (6) is used in unsupervised or supervised feature selection.

As stated above, a subset of features with maximum relevance and the minimum redundancy is chosen from each cluster in the proposed method. First, a feature such as $\acute{F}_1$ is selected that has more impact on detection categories than other features. In other words, feature $\acute{F}_1$ is chosen if the value of $@rel()$ function is maximized. It should be noted that each selected feature perches in the S set. In the second step, feature $\acute{F}_2$ is chosen among the remaining features if Equation (6) is minimized. Therefore, the feature $\acute{F}_2$ is relevant and has a little redundancy with feature $\acute{F}_1$. This process is repeated for each cluster until q feature is selected.

4 Experimental Results

The proposed method is implemented in the MATLAB environment of the 2012 version using a computer with operating system specifications Microsoft Windows 7 Ultimate 64-bit, 4 GB RAM, and Intel (R) Core (TM) i5- 2430M 2.40GHz processor. To evaluate the proposed method, we run it on the Moore [44, 45] and KDD Cup 99 [20, 45, 46] datasets and datasets from the UCI Machine Learning Repository. The results are compared with the reported results in some existing online and offline feature selection methods. These datasets have been used to evaluate the performance of various algorithms in other related papers as well.

In this section, based on the results, it is shown that the proposed method is faster than the two most popular online feature selection methods, RFOFS and OFS, and also it has better performance. In other words, the proposed method increases the accuracy of instances classification by choosing the appropriate features. Besides, the proposed method is faster with suitable accuracy, in instance classification, in comparison with some offline methods.

Afterward, used datasets, evaluation criteria, and the experimental results will be fully described.

4.1 Used Datasets

To Evaluate the proposed method, the following datasets are used in the simulation.

- The Moore dataset

The Moore dataset includes a variety of network traffics and attacks. Andrew Moore collected this dataset in 10 files of 5 to 17 megabytes in 2005. These collections include streams of traffic categories such as WWW, FTP, Mail, Database, Multimedia, Interactive, Attack, P2P, Games, and Services. In this dataset, each instance is described by 248 features.

The KDD Cup 99 dataset is one of the most common datasets to evaluate intrusion detection systems. This dataset was collected from the MIT lab in 1999 within a few weeks of network traffic. This dataset contains standard instances, including a set of simulated attacks and infiltrations in a network. Each instance in this data set describes a connection in the form of 41 features, which has 38 numerical features and three non-numeric features.

The KDD Cup 99 dataset contains 24 types of attacks, divided into four main categories, including DoS, R2L, U2R, and probe attacks [17, 42, 44].

Table 1 shows general information about the data set used in the experiment.

Table 1. The data set used in experiments

Name of data set	Number of instances	Number of features	Number of class
Moore	24863	248	10
Kdd cup 99	63966	41	5
Spambase	4601	57	2
Svmguide	1284	23	2

4.2 Evaluation Criteria

As mentioned in the first section, after selecting features, the feature selection method's performance should be evaluated. For this purpose, classifier performance can be assessed using a new dataset (data set with selected features). There are different criteria to assess classifier performance, which we will continue to explain.

For simplicity, at first, evaluation criteria related to issues with two categories, including positive and negative, are introduced. For this purpose, the symbols used in the evaluation criteria for two category issues are briefly described.

True Negative (TN) stands for the number of instances whose real category is negative, and also, the classification algorithm detected them negative correctly.

True Positive (TP) stands for the number of instances whose real category is positive, and also the classification algorithm has recognized them positively correctly.

False Positive (FP) stands for the number of instances whose real category is negative, and the classification algorithm has recognized them positively improperly.

False Negative (FN) stands for the number of instances whose real category is positive, and the classification algorithm has recognized them negatively improperly.

In this article, we have used Recall criteria (detection rate) and FPR (false positive rate) to evaluate the classifier's performance, which is obtained from Relations (7) and (8), respectively.

$$Recall = \frac{TP}{FN + TP} \tag{7}$$

$$FPR = \frac{FP}{FP + TN} \tag{8}$$

4.3 Experiment Results

In this section, the results of experiments performed on different datasets, based on evaluation criteria, are presented. Initially, the proposed method is compared with the two most popular online methods, including RFOFS and OFS. In these comparisons, the two Spambase and Svmguide from UCI datasets have been used. The Moore and KDD cup 99 data sets are not suitable for OFS and RFOFS methods because they have more than two categories, while OFS and RFOFS are only suitable for datasets with two categories. Afterward, the proposed method's performance is evaluated using the two Euclidean distance and mutual information criteria. Later, for the proposed method, the results of the experiments performed on the Moore and KDD cup 99 datasets are shown. At the end of this section, the proposed method is compared with the two most popular offline feature selection methods, including mRMR and FCFS methods. Three datasets, i.e., Spambase, Moore, and KDD cup 99 datasets, have been used to this end. In all experiments, we have used a decision tree, and the learning and testing of the classifier have been done based on Weka software.

In all experiments, it was assumed that 10% of the total instances were selected at each step.

4.3.1 Comparison of the Proposed Method with Online Methods

To evaluate the proposed method, we compare it with the two most popular online methods called OFS and RFOFS. In all experiments, after removing unrelated features, 30% of the relevant features were selected. It is necessary to mention that in the proposed method, the number of selected features is specified by the user. After selecting features using the proposed methods, OFS, and RFOFS, the decision tree classifier's accuracy on a new dataset based on selected features are calculated. In this way, two-thirds of the new dataset instance is selected as a training instance, and one-third of them are randomly selected as a test instance. After learning of classifier using training instances, the classifier is tested on the test instances. The classifier functionality is also evaluated using all the features and instances and to comparable it with the proposed method at various stages; it is shown as a smooth line. It should be noted that in the accomplished experiments in this section, the Euclidean distance criterion is used to cluster the features.

In Figure 4 and Figure 5, the average false positive rate is compared for the proposed methods, OFS and RFOFS, and all the features are presented. These comparisons are performed on the Spambase and Svmguide datasets. As depicted in these figures, the false positive rate decreases when the number of input instances increases in the proposed method. In other words, those features are selected that reduce the classifier error in the category detection of the instances. The value of the proposed method's false positive rate with the number of different instances is lower than the two other online methods. It should be noted that the false-positive rate for the two OFS and RFOFS online methods is higher than the state when all the features are considered. This is while, after entering a number of instances, the value of this criterion for the proposed method is lower than all features. As a result, using the proposed method, we

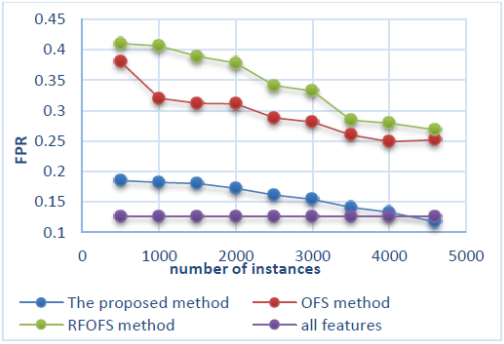

Figure 4. Comparison of the proposed method with online feature selection methods on the Spambase dataset with all the features based on FPR evaluation criteria

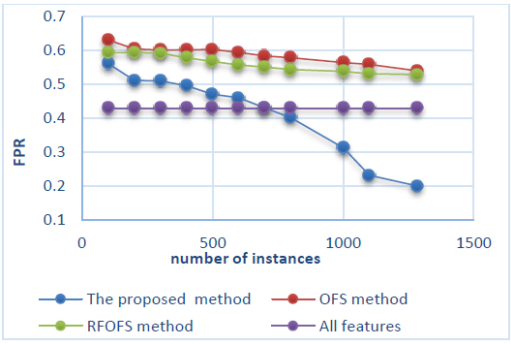

Figure 5. Comparison of the proposed method with online feature selection methods on Svmguide dataset with all features based on FPR evaluation criteria

can select those features that reduce the classification error rather than all features.

It is also clear that the speed of classifying instances is enhanced by feature selection because unrelated and additional features are eliminated.

The proposed method is evaluated in addition to the false positive rate using the rate of detection. As shown in Figure 6 and Figure 7, with increasing the number of input instances, the categories' average detection rate increases. In other words, the classifier determines the class of the instances with higher accuracy. It also has better performance than other online methods and all the features.

4.3.2 Performance evaluation of the proposed method based on Euclidean distance and mutual information criteria

In the proposed method, the similarity of features is calculated based on Euclidean distance and mutual information criteria. To compare these two criteria in Figure 8 and Figure 9, the detection rate is shown based on both criteria. As can be seen, the proposed method's detection rate is higher by using mutual information than the Euclidean distance. Because the Euclidean distance determines the amount of similarity (correlation) in the geometric space, while the mutual information criterion is a nonlinear correlation criterion based on information theory, which well determines the amount of correlation of the features. As a result, when the mutual information criterion to compute the graph's edges' weight, a better clustering of the features is performed, more appropriate features are selected.

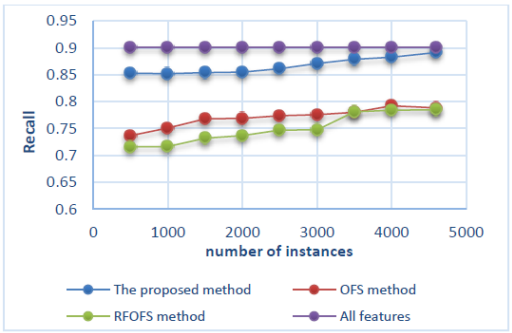

Figure 6. Comparison of the proposed method with online feature selection methods on the Spambase dataset with all features based on recall evaluation criteria

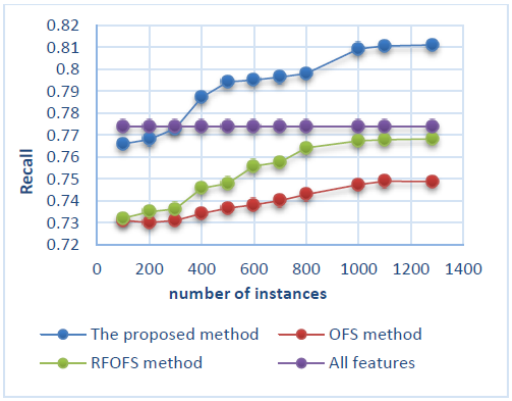

Figure 7. Comparison of the proposed method with online feature selection methods on the svmguide dataset with all features based on recall evaluation criteria

In Table 2, the proposed method is compared with the methods of the OFS and RFOFS in terms of time. In all data sets, the average time of the feature selection is calculated by entering each instance in seconds. The Moore and Kdd cup 99 data sets are not suitable for OFS and RFOFS methods because these data sets have more than two categories, while OFS and RFOFS are only suitable for datasets with two categories. The proposed method using both standard Euclidean distance and mutual information online has a better running time compared to other methods. It should be noted that the proposed method using the Euclidean distance criterion has less time to implement than mutual information criterion because

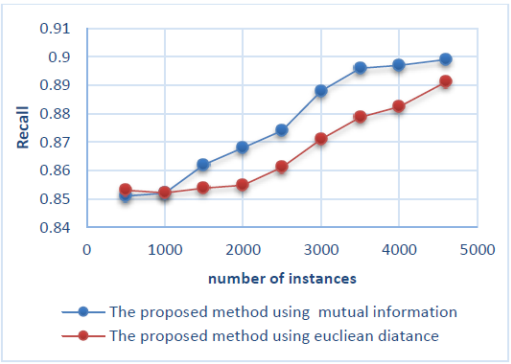

Figure 8. comparison Performance of the proposed method using two criteria of mutual information and Euclidean distance based on the Spambase dataset

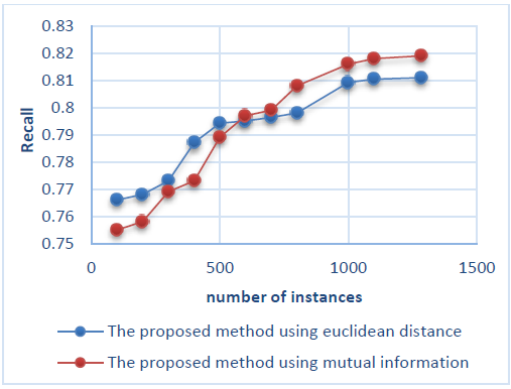

Figure 9. Comparison performance of the proposed method using two criteria of mutual information and Euclidean distance on the svmguide dataset

it requires less computation.

Table 2. Average feature selection time for each instance (in seconds)

Dataset name	The proposed method with Euclidean distance criterion	The proposed method with mutual information criterion	OFS	RFOFS
Moore	0.028	0.17	—	—
Kdd cup 99	0.002	0.005	—	—
Spambase	0.002	0.011	0.006	0.01
Svmguide	0.001	0.004	0.007	0.005

4.3.3 Performance Evaluating of the Proposed Method in Detecting Attacks

In Figure 10 and Figure 11, the results of the experiments are shown on the KDD Cup 99 dataset. In these experiments, the accuracy of the classifier is computed using the selected features by the proposed method and also by using all the features. As shown in Figure 10, by increasing the number of instances, the categories' average detection rate also increases. In addition, in the later stages, the precision of the classifier is higher than the case when all the features are considered employing the proposed method. Because the proposed method at each stage gets more experience to find the proper features, in other words, in the early stages, when the number of input instances is low, the proposed method cannot create appropriate clusters of features; thus, it is not able to find the proper subset of the features. Improving classifier performance using the proposed method instead of using all features show that many features of the KDD cup 99 dataset are additional and irrelevant. As previously stated, the OFS and RFOFS feature selection methods are only suitable for a dataset with two categories.

Given that the KDD cup 99 dataset has five categories; therefore, these methods cannot be used for this type of dataset. In Figure 9, the proposed method performance for the KDD cup 99 dataset is evaluated using the mean false positive rate. The results in this figure show that by increasing the number of instances, the percentage of instances that they determine the wrong category is zero. Also, by increasing the number of instances, the proposed method chooses the appropriate features. Therefore, the classifier function improves with the use of these features rather than considering all the features.

As shown in Figure 12 and Figure 13, the proposed method's performance in the Moore dataset improves with an increasing number of instances. Also, after entering a percentage of the instances, the proposed method, with experience (finding the appropriate clusters), chooses the relevant features, so that the classifier function is better than when all the features are taken into consideration.

The proposed method is efficient for datasets with more than two classes, such as KDD cup 99, while other online methods [39, 40] provided for intrusion detection systems can only use effectively for datasets with two classes of attack and normal.

Table 3 and Table 4 show the performance of the proposed method for the four attack and normal classes in dataset KDD cup 99, for the first 10% of the instances and the last 10% of the instances, respectively. Comparing the results in these two tables shows that with increasing instances and gaining experience, the false positive rate for all categories decreases. Increasing the values of recall, precision, and F-measure for all categories is proof of the efficiency of the proposed method.

4.3.4 Comparison of the Proposed Method with Offline Methods

To show the proposed method's computational efficiency, we compared it with two of the important methods of offline feature selection. Experiments were

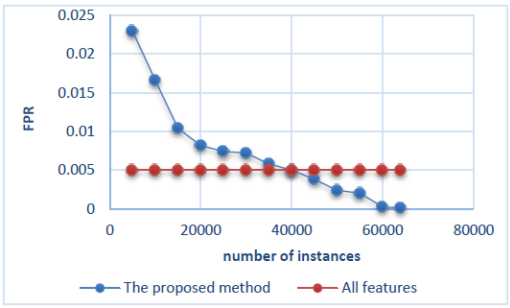

Figure 10. Comparison of the proposed method with all features on the KDD cup 99 dataset using the FPR evaluation criterion

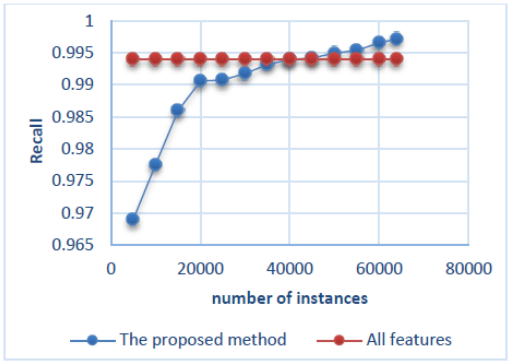

Figure 11. Comparison of the proposed method with all the features on the KDD cup 99 dataset using the recall evaluation criterion

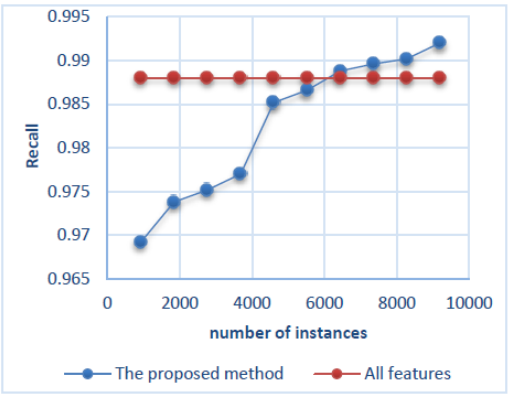

Figure 12. Comparison of the proposed method with all the features based on the Moore dataset using the recall evaluation criterion

performed on KDD cup 99 and Moore datasets. By comparing the average detection rate of offline methods (mRMR and FCFS) with online methods presented in Table 3, it can be concluded that although all of the features and instances are available in offline methods but the proposed method is more efficient. Besides, as shown in Figure 14 and Figure 15, we observe that the proposed algorithm is faster than the offline feature selection algorithms, especially when the number of features to be selected is large because in the proposed method, initially irrelevant features

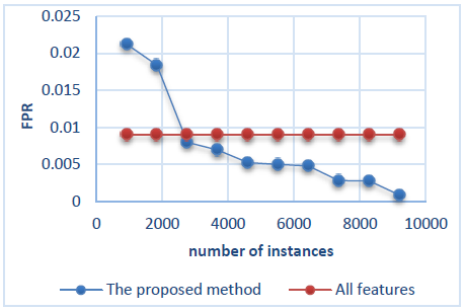

Figure 13. Comparison of the proposed method with all the features based on the Moore dataset using the recall evaluation criterion

Table 3. : Efficiency of the proposed method after entering the first 10% of the data in KDD cup 99.

Class name	F-Measure	Recall	Precision	FPR
Norma	0.982	0.983	0.98	0.03
R2L	0.924	0.929	0.921	0.01
Probe	0.881	0.833	0.852	0.001
Dos	0.972	0.969	0.976	0.001
U2R	0.899	0.905	0.896	0.002

Table 4. Efficiency of the proposed method after entering the last 10% of the data in KDD cup 99.

Class name	F-Measure	Recall	Precision	FPR
Normal	0.998	0.998	0.997	0.0002
R2L	0.993	0.993	0.993	0.0001
Probe	0.994	0.994	0.995	0
Dos	0.998	0.998	0.998	0
U2R	0.991	0.992	0.992	0.00002

are removed; so, the runtime for constructing the graph is not high in the clustering of features and composition of the clusters. Also, regarding the features selection, searching is performed within each cluster, and there is no need for searching the entire feature space and check all the subsets of the features.

As shown in Figure 16 and Figure 17, if the decision tree on the entire feature space (without feature selection) will be faster than the decision tree on reduced feature space (with proposed feature selection), especially when the number of features to be selected is large.

5 Conclusion

In this article, to increase the accuracy in detecting attacks, a new graph-based method was proposed for online feature selection. The primary approach in the proposed method was that irrelevant features were firstly removed by entering a limited number of instances. The proposed method was compared with the OFS and RFOFS online feature selection methods and the offline methods of mRMR and FCFS. Our proposed method has better performance than all of them. Besides, the proposed method is not

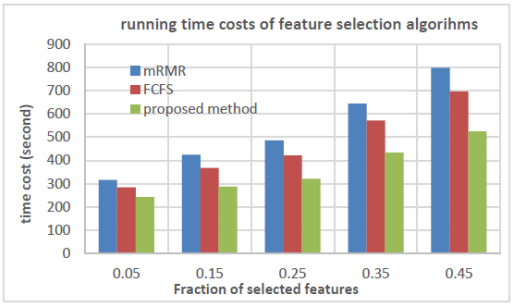

Figure 14. Evaluation of time efficiency: proposed method versus Offline feature selection (mRMR and FCFS) on the Moore dataset.

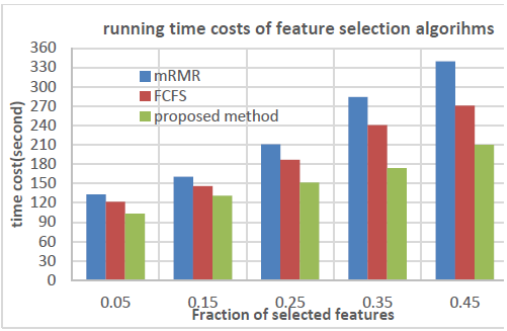

Figure 15. Evaluation of time efficiency: proposed method versus Offline feature selection (mRMR and FCFS) on the kdd cup 99 dataset

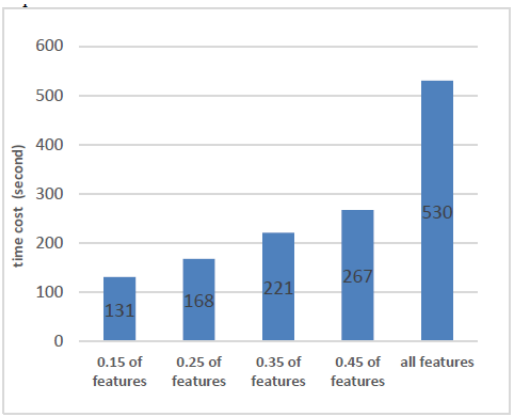

Figure 16. Time is taken to build classifier trees j48 after feature selection(with the proposed algorithm) versus without feature selection(all features) on the KDD cup 99 dataset

limited to only two-class classification, as the other methods mentioned. In this regard, the proposed method was evaluated using Moore and KDD Cup 99 datasets. The experiments' results showed that the proposed method has a better performance than when all features were considered. Also, by selecting the appropriate features, attacks are detected with high accuracy.

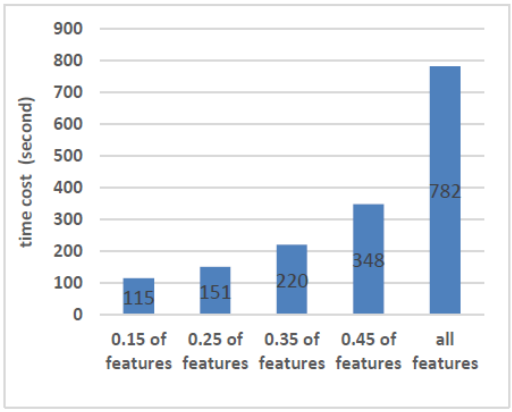

Figure 17. Time is taken to build classifier trees j48 after feature selection (with the proposed algorithm) versus without feature selection (all features) on the Moore dataset

References

[1] J.McHugh, A.Christie, J.Allen, "Defending yourself: The role of intrusion detection systems", *IEEE software*, vol. 17, no. 5, pp. 42-51, 2000.

[2] S.Aljawarneh, M.Aldwairi, M.Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model", *Journal of Computational Science*, vol. 25, pp. 152-160, 2018.

[3] B. Morin and M. Ludovic, "Intrusion detection and virology: an analysis of differences, similarities and complementariness", *Journal in Computational Virology*, vol. 3, no. 1, pp. 39–49, 2007.

[4] J. Davis and A. Clark, "Data preprocessing for anomaly-based network intrusion detection: A review",*Computers Security*, vol. 30, no. 6, pp. 353-357, 2011.

[5] "High-Speed Security Log Analytics Using Hybrid Outlier Detection". *Doctoral thesis*, Universität Potsdam, 2019.

[6] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fernandez, E. Vazquez, "Anomaly-based network intrusion detection: techniques, systems and challenges", *Computer Security*, vol. 28, no. 1, pp. 18–28, 2009.

[7] H. Liao, C. Lin, Y. Lin, K. Tung, "Intrusion detection system: a comprehensive review", *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16-24, 2013.

[8] A. Patcha, J. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends", *Computer Networks*, vol. 51, no. 12, pp. 3448–3470, 2007.

[9] Kunal and M. Dua, "Machine Learning Approach to IDS: A Comprehensive Review," *3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, pp. 117-121, 2019.

[10] A.Verma, V.Ranga, "Machine Learning Based Intrusion Detection Systems for IoT Applications". *Wireless Personal Communications*, pp.2287–2310, 2020.

[11] A. Amouri, V. T. Alaparthy, and S. D. Morgera, "A Machine Learning Based Intrusion Detection System for Mobile Internet of Things", *Sensors (Basel)*, vol.20, no. 2, 2020.

[12] H. Bhuyan, Monowar, K.Dhruba Bhattacharyya, and K. Jugal Kalita. "Survey on incremental approaches for network anomaly detection", *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 3, no. 3, December 2011.

[13] T. Chou, K. Yen, J. Luo, "Network intrusion detection design using feature selection of soft computing paradigms",*International Journal of Computational Intelligence*, vol. 4, no. 3, pp. 196–208, 2008.

[14] L. Ladha, T. Deepa, "Feature Selection Method and Algorithms", *International Journal on Computer Science and Engineering*, vol.3, no. 5, pp. 178-179, 2011.

[15] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection", *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[16] J. Wang, P. Zhao, C. Hoi, and R. Jin, "Online feature selection and its applications", *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–14, 2013.

[17] U. Xindong, K. Yu, W. Ding, W. Hao, Z. Xingquan, "Online feature selection with streaming features", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 5, pp. 1178-1192, 2013.

[18] R. Collins, Y. Liu, M. Leordeanu, "Online selection of discriminative tracking features", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1631-1643,2005.

[19] K. Glocer, J. Theiler, "Online feature selection for pixel classification", *in Proceedings of the 22nd international conference on Machine learning*, pp. 249–256, 2005.

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

[21] L. Yu, H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution", *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*, vol. 3, pp. 856-863, 2003.

[22] H. Mark, "Correlation-based Feature Selection for Machine Learning", *Ph.D. Thesis*, 1999.

[23] Z. Zhang and E. Hancock,"A graph-based approach to feature selection", *Springer Berlin Heidelberg*, pp. 205-214, 2011.

[24] P. Moradi and M. Rostami, "A graph theoretic approach for unsupervised feature selection"", *Journal of Engineering Applications of Artificial Intelligence*, vol. 44, pp. 33-45, 2015.

[25] F. Zhang, D. Wang, "An effective feature selection approach for network intrusion detection", *IEEE Eighth International Conference on Networking*, pp. 307-311, 2013.

[26] G. Stein, H. Wu, "Decision Tree Classifier for Network Intrusion Detection with GA-based Feature Selection", *Proceeding of the 43rd annual Southeast regional conference*, pp. 136-141, 2000.

[27] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection", *The Journal of Machine Learning Research*, vol. 3, pp. 1157-1182, 2003.

[28] D. Zhang, S. Chen, Z. Zhou, "Constraint score: A new filter", *Pattern Recognition*, vol. 41, no. 5, pp. 1440–1451, 2008.

[29] Z. Zhao and H. Liu, "Semi-supervised feature selection via spectral analysis", *Proceedings of the 2007 SIAM International Conference on Data Mining (SDM)*, 2007.

[30] Z. Xu and R. Jin, "Discriminative semi supervised feature selection via manifold regularization", *IEEE Transactions on Neural Networks*, vol. 21, no. 7, pp. 1033–1047, 2010.

[31] M.Javadi, S.Eskandari, "Online streaming feature selection: a minimum redundancy, maximum significance approach", *Pattern Analysis and Applications*, vol.22, no. 3, pp.949-963, 2019.

[32] S. Perkins and J. Theiler, "Online Feature Selection Using Grafting", *Proceedings of the 20thInternational Conference on Machine Learning*, pp. 592-599, 2003.

[33] J. Zhou, D. Foster, R. Stine, and R. Ungar, "Streaming Feature Selection Using Alpha-Investing", *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pp. 384 -393, 2005.

[34] U. Xindong, K. Yu, W. Ding, W. Hao, Z. Xingquan, "Online feature selection with streaming features", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 5, pp. 1178-1192, 2013.

[35] Y. Kui, W. Xindong, W. Wei, and P. Jian, "Towards Scalable and Accurate Online Feature Selection for Big Data", *IEEE International Conference on data mining*, pp. 660-669, 2014.

[36] H. Grabner, H. Bischof, "Online boosting and vision", *Computer Vision and Pattern Recognition, IEEE Computer Society*, vol. 1, 2006.

[37] J. Wang, P. Zhao, C. Hoi, and R. Jin, "Online fea-

ture selection and its applications", *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–14, 2013.

[38] H. Zheng and H. Zhang, "Online Feature Selection Based on Passive-Aggressive Algorithm with Retaining Features", *Web Technologies and Applications, Springer International Publishing*, pp. 707-719, 2015.

[39] Z.Cataltepe, U.Ekmekci, T.Cataltepe, and I.Kelebek. "Online feature selected semi-supervised decision trees for network intrusion detection." *In NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*, pp. 1085-1088, 2016.

[40] B.Atli, and A.Jung, "Online feature ranking for intrusion detection systems", *arXiv preprint arXiv:1803.00530*, 2018.

[41] M. Hinkka, T. Lehto, K. Heljanko and A. Jung, "Structural feature selection for event logs", *in Business Process Management Workshops. BPM 2017*, 2017.

[42] X. Rui and W. Donald, "Survey of clustering algorithms", *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645-678, 2005.

[43] J. Han, M. Kamber, "Data Mining: Concepts and Techniques (3rd ed.)", *San Francuisco, CA, USA: Morgan Kaufmann Publisher Inc.*, 2011.

[44] C. Zhong, M. Duoqian and W. Ruizhi, "A graph-theoretical clustering method based on two rounds of minimum spanning trees", *Pattern Recognition*, vol. 43, no. 3, pp. 752-766, 2010.

[45] R. Ghaemi, M. Sulaiman, N. Ibrahim, "A survey: clustering ensembles techniques", *World Academy of Science, Engineering and Technology*, vol. 50, pp. 636-645, 2009.

[46] S. Mimaroglu, E. Erdil, "An efficient and scalable family of algorithms for combining clustering", *Engineering Applications of Artificial Intelligence*, vol. 26, no. 10, pp. 2525-2539,2013.

[47] BRAZIL, http://www.cl.cam.ac.uk/.

[48] M. Hosseinzadeh Aghdam and P. Kabiri, "Feature Selection for Intrusion Detection System Using Ant Colony Optimization", *International Journal of Network Security*, vol. 18, no. 3, pp.420-432, 2016.

[49] O. Al-Jarrah and A. Elsalamouny, "Machine-Learning-Based Feature Selection Techniques for Large-Scale Network Intrusion Detection", *IEEE 34th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, 2014.

Hajar Dastanpour received the B.Sc. degree in computer science from Kashan University , Iran, in 2013. She received the M.Sc. degree in computer engineering from Isfahan University of Technology in 2016. Her thesis investigates online feature selection to improve detection of new attacks in intrusion detection systems. Her research interests include artificial intelligence, intrusion detection systems and image processing.

Ali Fanian received the B.Sc., M.Sc. and Ph.D. degree all in computer engineering from Isfahan University of Technology (IUT) in 1999, 2002 and 2011 respectively. He is currently an associate professor at Department of Electrical and Computer Engineering, Isfahan, Iran. He is a member of a research group on security in networks and systems in IUT and he is also manager of an academic CSIRT, IUT. His current research interests include network security, internet of things (IoT), intrusion detection systems, blockchain and cryptocurrency technology.