The ISC Int'l Journal of Information Security

July 2019, Volume 11, Number 2 (pp. 159–171)

http://www.isecure-journal.org

Anomaly Detection Using SVM as Classifier and Decision Tree for Optimizing Feature Vectors

Elham Serkani [1], Hossein Gharaee [2,*], and Naser Mohammadzadeh [1]

[1] *Department of Computer Engineering, Shahed University, Tehran, Iran*

[2] *Iran Telecom Research Center, Tehran, Iran*

ARTICLE INFO.

Article history:
Received: 30 December 2018
Revised: 1 July 2019
Accepted: 20 July 2019
Published Online: 31 July 2019

Keywords:
Intrusion Detection, Feature Selection, Support Vector Machines, Decision Tree.

ABSTRACT

With the advancement and development of computer network technologies, the way for intruders has become smoother; therefore, to detect threats and attacks, the importance of intrusion detection systems (IDS) as one of the key elements of security is increasing. One of the challenges of intrusion detection systems is managing a large amount of network traffic features. Removing unnecessary features is a solution to this problem. Using machine learning methods is one of the best ways to design an intrusion detection system. Focusing on this issue, in this paper, we propose a hybrid intrusion detection system using the decision tree and support vector machine (SVM) approaches. In our method, the feature selection is initially done by the C5.0 decision tree pruning, and then the features with the least predictor importance value are removed. After removing each feature, the least square support vector machine (LS-SVM) is applied. The set of features having the highest surface area under the Receiver Operating Characteristic (ROC) curve for LS-SVM are considered as final features. The experimental results on two KDD Cup 99 and UNSW-NB15 data sets show that the proposed approach improves true positive and false positive criteria and accuracy compared to the best prior work.

1 Introduction

Nowadays, network security is becoming an increasingly important demand. As networks can become vulnerable to attacks from both internal and external intruders, IDSs are becoming more important to manage threats and attacks. An IDS helps to detect unauthorized use, alteration, and destruction of information systems [1]. IDSs are classified into two groups: host-based and network-based; Host-based IDSs monitor the data and processes of a particular host's software environment. On the other hand, network-based IDSs detect attacks through network traffic monitoring [1, 2].

The detection methods used by IDSs can be categorized into two main categories: misuse-based and anomaly-based [1, 2]. Misused-based IDSs detect attacks by comparing new traffic data with the signature of known attacks. They have a high true positive rate for known attacks. However, these systems are not able to detect new attacks. Furthermore, it is necessary to update the database of attack signatures continuously [3]. Anomaly-based IDSs model the normal network behavior. A deviation from the normal behavior model indicates the occurrence of an

* Corresponding author.
Email addresses: e.serkani@itrc.ac.ir (E. Serkani), gharaee@itrc.ac.ir (H. Gharaee), mohammadzadeh@shahed.ac.ir (N. Mohammadzadeh)
ISSN: 2008-2045

anomaly. The main advantage of this category is its ability to detect zero-day attacks [3]. Anomaly detection is an estimation technique; Despite the difficulty of predicting behavior, machine learning algorithms can help to behavioral modeling, effectively.

A Support Vector Machine (SVM) is an outstanding and well-known machine learning classification method that can be used to design an IDS. The key advantage of the SVM is its mathematical tractability and geometric interpretation basic. SVM can be considered as an empirical risk minimization method and consequently, defines the theoretical bounds on its performance. Least Squares SVM (LS-SVM) is a reformulation version of SVM that solves the SVM convex quadratic programming problem, faster and easier, with the higher performance [4]. Also, feature selection is a useful step before the development of many machine learning-based systems, such as IDSs that can improve classifier efficiency.

The feature selection eliminates the irrelevant or redundant features in each class; It reduces the computational complexity and can optimize the classification results. The most well-known classification for feature selection approaches classifies these methods into the filter-based and wrapper-based methods. In the wrapper-based methods, a classification algorithm is used to assess various feature subsets. However, in the filter-based approaches, the best feature subsets are identified using statistical techniques [5, 6]. Some machine learning methods can be used to select features in the form of the filter or wrapper techniques.

In the literature, a decision tree is one of the most popular and influential classifications and predictive machine learning methods. Decision trees use the divide and conquer method; in each step, they split the instances according to specific feature values that are selected using a criterion; Various decision tree algorithms used different criteria to this progress. The decision tree has a high predictive performance in spite of its small computational effort. It can easily handle the large datasets and datasets with missing values. Various researches use the decision tree in different manners to develop or enhance an IDS. In some of them, it acts as a classifier [7–15]. Another group of researchers used the decision tree building criteria, such as information gain, gain ratio to prioritize and select the best features [8, 16–20]. In a decision tree expansion process, after dividing the dataset using different values of the feature with the highest criterion, the criterion will be computed again because of relation exists between the features. It helps to determine the maximum and the minimum number of features in the regular form, but the mentioned approach outlines the number of features manually. However, the main flaw of this viewpoint is using the criteria in one step, without building the decision tree. In such a case, they are forced to determine the number of features contractually. On the other hand, some researchers used the decision tree for extracting the rules for a misused-based IDS [21].

The decision tree algorithms prefer a tree that is not complex, with the minimum number of nodes and depth. Reducing tree complexity increases its accuracy [22]. A decision tree uses some methods like pruning to getting simply. This paper develops an anomaly-based NIDS using C5.0 decision tree algorithm to optimizing feature vectors and LS-SVM as the classifier. We named our proposed method DT-LSSVM. The decision tree's desire to reduce its complexity and simplicity is the foundation of proposed feature selection [23]. In this paper, we extend the previous method in a new structured way by combining the filter-based and wrapper-based feature selection methods [24]. Also, the values of the true negative rate and false positive rate are added to the normal class detection results. On the other hand, we assess our proposed on two well-known data sets named KDD CUP 99 and UNSW-NB15. Whereas, the previous research has used the UNSW-NB15 data set. Our new feature selection method is a hybrid model where both filter-based and wrapper-based approaches are used to find the best minimal feature set. Accordingly, DT-LSSVM has three components that first and second ones include the hybrid feature selection, and the last component intends our NIDS detection phase. We use multiple criteria together to select the features. Whereas, in the first component, at first, features without any effect on the demarcation of the training instances, are eliminated during extending decision tree using the gain ratio criteria. Then the other novelty of our feature selection is downsizing the remainder feature set by applying the error-based pruning algorithm with the pessimistic upper bound error criterion to prune the constructed tree, without losing detection performance [25, 26]. Here, both decision tree and error-based pruning algorithm try to find the best feature set relied on the general characteristics of the data. Therefore, this phase is considered as the filter-based feature selection approach. Decision tree and error-based pruning algorithm may tend to select redundant features; therefore, in the second component, a wrapper-based method is utilized to achieve the minimum number of features. The second phase uses the predictor importance criterion, presented in the C5.0 algorithm, to select feature subsets and then evaluate them using LS-SVM algorithm. To best of our knowledge, this is the first attempt of usage predictor importance criterion to rate the features and choose the feature set in the intrusion detection do-

main. The output of the second component is the final feature set which will be used in the third component to design the final NIDS with LS-SVM classifier.

The rest of this paper is organized as follows. Section 2 presents a review of the related work. The proposed anomaly detection method is described in Section 3. Section 4 includes the experimental results. Finally, Section 5 concludes the paper.

2 Related Work

SVM is one of the most well-known methods of machine learning classification, which has been employed by numerous researchers to design IDSs. Some machine learning methods, such as genetic algorithm [27–30], principal component analysis algorithm [31], decision tree [19, 32], ant colony [33, 34], and modified mutual information [35] methods are used with the SVM to improve its performance. In a proposed hybrid IDS, the k-means was used for clustering the KDD CUP 1999 training dataset and performing the feature selection, and the SVM algorithm with the RBF kernel was employed as the classifier [36]. Xingzhu proposed an IDS using the ant colony method for feature selection and a feature weighting SVM [34]. Amiri et al. studied two feature selection methods, viz. linear correlation coefficient and forward feature selection algorithm (FFSA) and proposed modified-mutual information feature selection (MMIFS). They used the LS-SVM as the classifier. According to the numerical results, MMIFS performed well as regards feature selection of the Probe and R2L attacks while FFSA performed well in U2R, DoS, and normal traffic. However, these employed methods are statistical methods that only act by estimating the appropriate results [35]. Sainis et al. reduce the feature numbers using common feature selection techniques correlation-based Feature Selection (CFS). They use different machine learning classifier to compare feature reduction techniques. Dataset KDD cup 99, NSL-KDD and GureKDDcup are used in this paper to evaluate the proposed method They identify attacks Dos, Probe, R2L and U2R in their proposed method [10]. Nskh et al. used the principal component analysis (PCA) method to select the features. They analyzed the performance of different kernel functions for SVM; finally, the SVM with the RBF kernel function was employed as their proposed IDS classifier. The results revealed that feature selection reduced the training and testing times in the IDS. This research just considered the normal and attack class types, and it cannot detect the type of attacks [31]. Nema et al. proposed a multi-layer IDS. They have used the SVM as the classifier and employed the genetic algorithm to perform feature selection. Each layer detects one of the Probe, DoS, R2L, and U2R attacks in NSL KDD dataset [29]. Moreover, in our last work, we used a combination of the genetic algorithm and LS-SVM to select the optimal and appropriate features for the KDD CUP 99 and the UNSW-NB15 datasets. We provided a new fitness function according to the effect of the features in the LS-SVM detection accuracy [28]. The genetic algorithm has high computational complexity, and despite the fact that the genetic algorithm may lead to the optimal feature set, it has no guarantee to achieve the optional result. Therefore, we decided to use the decision tree to take its advantages like simplicity and adaptability.

Peddabachigari et al. proposed a hierarchical hybrid IDS using decision tree and SVM. They used the ensemble model based on the decision tree, SVM and their DT-SVM hybrid model. When none of the basic classifiers can detect a class as well, using the ensemble approach will not be suitable; thus, the final results are not appropriate for some class like U2R. This article uses the accuracy criteria and doesnt mention to others like false positive rate [37]. Mulay et al. proposed tree structured multiclass SVM to increase the accuracy criterion of IDS, but they dont prove their claim using numerical experiences [13]. Similarly, Teng et al. combined the decision tree and binary SVM methods to develop a multi-class SVM based on KDD CUP 1999 different types of attacks. They designed their IDS for TCP, UDP and ICMP network packets. Also, they selected the features for these three groups separately; but they did not mention how they did it. This proposed IDS improved the accuracy compared to using SVM or decision tree singly. The results have shown an average accuracy of 89.02% [12]. In the method proposed by Goeschel, first, the linear SVM approach was used to divide the data into the Normal and Attack classes. Afterward, the attack traffic was processed in phase 2 using the J48 decision tree to specify the attack types. In the next step, nave Bayes classifier and J48 decision tree were employed to estimate the unknown instances classes. The decreased FPR was the main finding from this research but using three classifiers and lack of feature selection, cause a heavy computational detection and wasting the time [15]. These works have used the decision tree as the IDS classifier. In some other studies, decision tree making criteria, such as information gain, gain ratio, and chi-square have been used to determine the features priority for feature selection. Foroushani et al. proposed an anomaly-based IDS using the decision tree and k-NN approaches. They employed the "information gain" criterion of the ID3 decision tree to determine the importance of the features and select 30 more important features [8]. Sangkatsance et al. proposed a real-time IDS named RT-IDS. In this system, after extracting 12 features from the headers of the network packets, the importance of each feature

was determined using the "information gain" criterion. The strength of the proposed RT-IDS was its ability to detect the DoS and Probe methods [20]. Moreover, in another work, the authors used the gain ratio criterion for rating the features. Then they choose the first 35 features with the heist gain ratio value [16]. Similarly, Latha et al. prioritize the features using a combination of gain ratio and chi-square criteria [18]. Janarthanan et al. performed feature selection by combining the methods of feature selection, including CfsSubsetEval (attribute evaluator) + GreedyStepwise method and InfoGainAttibuteEval (attribute evaluator) + Ranker method in weka. They used a random forest classification method to classify the traffic data. Also, they conducted their experiments on the UNSW-NB15 dataset [38]. The significant vacuum of such methods is using decision tree criteria once, without extending a decision tree. Therefore, they can't determine the best number of features for each attack and normal classes separately.

3 Proposed Method

In this paper, we aim to develop a hybrid anomaly detection system using state-of-the-art machine learning techniques. To do so, our proposed system utilizes both filter and wrapper-based feature selection methodologies to find the best subset of features, which outperforms our detector. The proposed system adopts the decision tree as a filter-based feature selection method in addition to the LS-SVM as the state-of-the-art learner which is employed in wrapper-based feature selection and considered as our final IDS. Hence, our proposed method is named as DT-LSSVM.

Figure 1 depicts the overall architecture of DT-LSSVM. It has three main phases, 1) filter-based feature selection, 2) wrapper-based feature selection, and 3) detection. Through the first phase, a subset of features is selected regardless of our detection model. Since filter-based methods rely on general characteristics of the data, they are particularly effective in computation time and robust to over-fitting. However, filter methods tend to select redundant features because they do not evaluate various selected feature sets. Therefore, they are mainly used as a pre-processing method, and here we consider filter-based feature selection as the first phase of our proposed system. On the other hand, in the second phase, wrapper methods evaluate subsets of features which allow detecting the possible interactions between features. In other words, the wrapper model requires one predetermined machine learning algorithm and uses its performance as the evaluation criterion. Since our system uses LS-SVM as its intrusion detector, our proposed wrapper feature selection searches for the features better suited to LS-SVM and improves its performance. In the last

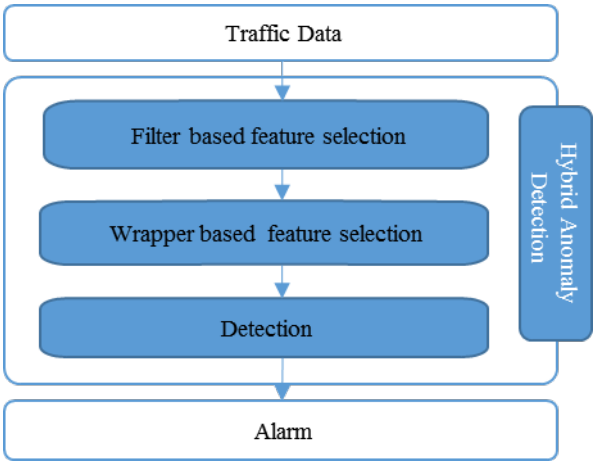

Figure 1. A representation for the proposed DT-LSSVM IDS

phase, named detection, we use the LS-SVM as a machine learning technique to predict the alarm of unseen traffic data. In following, the two main phases of our proposed system, i.e. the filter-based feature selection and the wrapper-based feature selection, are described in detail.

3.1 Phase 1: filter-based feature selection

As shown in Figure 2, phase 1 aims to identify the features in each dataset that more properly distinguishes each class from the other classes. Here, we consider the binary classification problem. Hence, for the normal category and for each attack category, an independent classifier is built. In each one, the samples of the corresponding category are considered as positive samples, and the other samples would be considered as negative ones. Thus, through the preprocessing component of our architecture, the entire dataset is reshaped in which consists of two classes (i.e. the samples from the desired category are labeled positive and the samples from the other categories are labeled negative).

To find the distinguishable feature set, we proposed a three-layered feature selection mechanism. Firstly, a decision tree is utilized to find features which can more accurately distinguish samples according to general characteristics of the data. Secondly, we apply error-based pruning on the obtained decision tree to eliminate the features which do not affect the classification error, means the elimination of such features does not increase the classification error. Finally, we use a wrapper-based feature selection method using LS-SVM as its classifier to find the minimal effective feature set. Please note that the first and second techniques are filter-based feature selection method and would be done in phase 1.

A decision tree is a classifier that divides the instances with a greedy recursive approach. A decision tree method consists of the following steps.

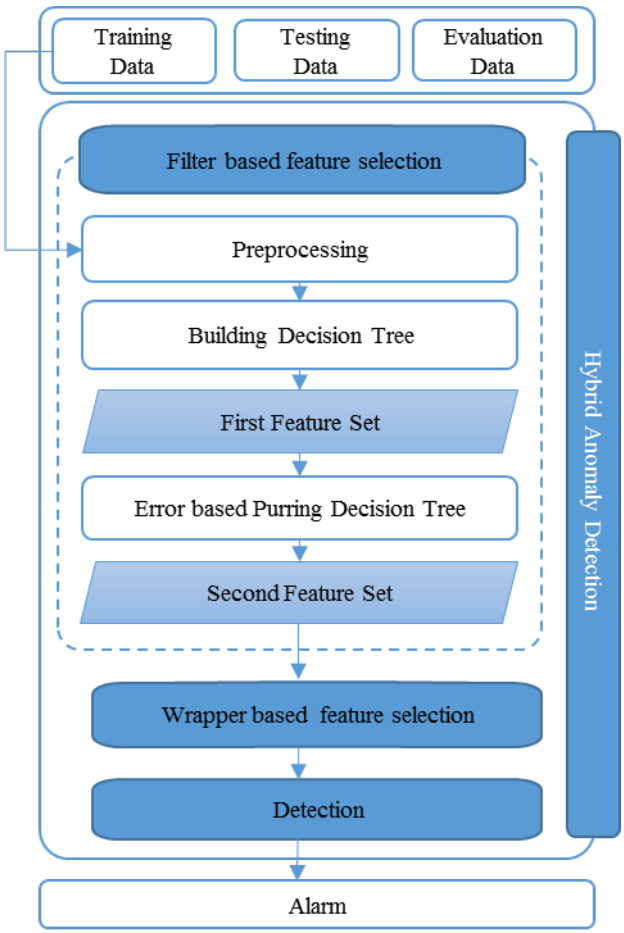

Figure 2. Phase 1: filter-based feature selection

(1) Consider all the training data in the tree root.
(2) Select a feature for expansion based on statistical and specific criteria. Here, since our decision tree is C5.0, the gain ratio is utilized to expand the tree nodes.
(3) Divide the data based on the values of the selected feature in two or more branches.
(4) Repetition of steps 2 and 3 in a recursive manner on each branch of division. This repetition would be ended when each following condition is satisfied:
    (a) When all samples are put into one category after classification
    (b) When there is no other feature for expansion

The tree has grown to a predefined size limit The C5.0 algorithm is an improved version of the well-known and commonly used decision tree algorithm named C4.5 [39]. C5.0 is built with a top-down recursive greedy approach and utilizes the gain ratio to expand the tree nodes; So that, at each stage, the feature that has the highest gain ratio value is chosen for expansion. The Gain ratio is calculated by the following formula:

$$GainRatio\,(S,A) = \frac{IG\,(S,A)}{SplitInformation\,(S,A)} \quad (1)$$

Where S presents instances, A is a specific feature, and IG denotes the information gain criterion [40]. This criterion, which is used to expand a decision tree using the ID3 algorithm, is obtained via the following relation.

$$IG\,(S,A) \equiv Entropy\,(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} log_2 \frac{|S_v|}{|S|} \quad (2)$$

Where S_v is a subset of S in which the value of the feature A is equal to v for all instances. $Entropy(S)$ determines the impurity of the S dataset and is calculated as follows:

$$Entropy\,(S) = -\sum_{x=1}^{c} P_i \times log_2 P_i \bigcup_{i=1}^{n} X_i \bigcup_{i=1}^{n} X_i \quad (3)$$

In formula (1), the $SplitInformation$ cancels the effect of the features that have a large number of values in which have large IG. This value is obtained by formula (4).

$$SplitInformation\,(S,A) \equiv -\sum_{i=1}^{c} \frac{|S_i|}{|S|} log_2 \frac{|S_i|}{|S|} \quad (4)$$

After building our decision tree, an error-based pruning mechanism is carried out to reduce the classification error caused by the noise or excessive details of the training data [26, 41]. Indeed, error-based pruning is a more sophisticated version of pessimistic pruning. In error-based pruning, each node from root to leaves can be considered to be deleted. Each node and its sub-tree would be deleted if the pessimistic upper bound error of classification when this node is not considered is not greater than the upper bound error when this node exists in the decision tree. Considering the confidence level of α, the pessimistic upper bound error is obtained using the normal approximation to the binomial distribution. This error can be calculated via the following formula [41, 42].

$$\varepsilon_{UB}\,(T,S) = \varepsilon\,(T,S) + Z_\alpha \sqrt{\frac{\varepsilon\,(\varepsilon\,(T,S)\,.\,(1-\varepsilon\,(T,S)))}{|S|}} \quad (5)$$

In this formula, $\varepsilon\,(T,S)$ is the false classification rate for tree T in the S training dataset. Moreover, Z is the

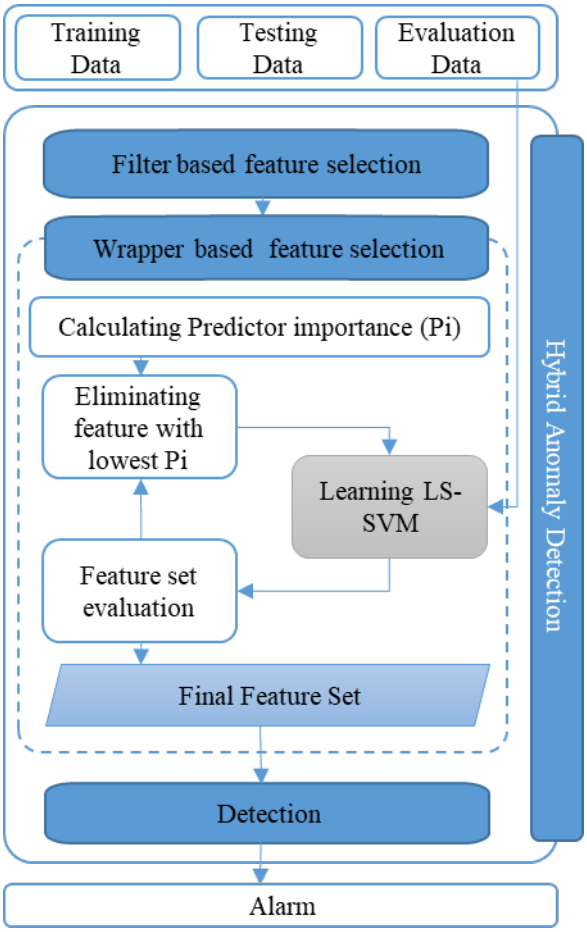

Figure 3. Phase 2: wrapper-based feature selection

inverse of the standard normal cumulative distribution and α shows the confidence level. The following values are calculated through top-down navigation of the nodes.

3.2 Phase 2: wrapper-based feature selection

Through the previous phase, a feature set is firstly selected by a decision tree based on the gain ratio, and then this feature set is modified according to the pessimistic upper bound error. Therefore, the input samples of this phase are presented in the selected feature space. Here, we aim to develop a wrapper-based feature selection method to select the minimal best feature set among the possible feature set outputted from the previous phase (Figure 3). To do so, firstly the predictor importance (PI) of each feature in the pruned decision tree is calculated. Then, the features are sorted by this criterion.

PI is a feature evaluation criterion which is calculated for each node of the C5.0 decision tree. The PI of each node specifies the percentage of the training instances that are divided by the values of the corresponding feature in this node. For example, the PI of the root node is 100% because all samples are in the root and consequently, the feature presented in the root node influences the classification of all samples. Similarly, this ratio is calculated for each feature in the nodes [26].

The third level of our proposed feature selection is a wrapper-based method which is done as following steps:

(1) Considering training samples using the current feature set.
(2) Learning LS-SVM using the above training samples.
(3) Evaluating the current feature set by measuring the performance of LS-SVM on the test samples.
(4) Updating the current feature set by eliminating the feature with the lowest PI from the current feature set.
(5) Repeating phases 1 to 4 until the ending condition is satisfied.

The ROC AUC criterion of the LS-SVM is calculated to measure the performance of classification in step 2. ROC AUC is a performance measure for machine learning algorithms. ROC is a graph of True positive rate against false positive rate. The area under the ROC curve provides a scalar criterion for comparing the performance of different classifiers. This criterion is between 0 and 1. The closeness of its value to 1 shows a better performance, and if this criterion is less than 0.5, the classifier is unacceptable. ROC AUC is not sensitive to changing the data distribution as well as it considers TPR and FPR in proportion to each other.

Algorithm 1 Phase 2: Wrapper-based feature selection

Require: X_{train} is the training sample set, X_{test} is the test sample set, FS is the second feature set selected in phase 1, m is the minimum number of feature which must be in final features set, N is the size of the second features set
Ensure: Final selected feature set
1: Calculating PI of the features in FS
2: Sorting FS according to the PI of each feature increasing
3: $FS_0 = \{f_1, f_2, \cdots, f_n\}$ where $PI(f_i) < PI(f_j)$ and $i < j$
4: **for** $k = 0$ to $(n - m) - 1$ **do**
5: Considering training samples $(X_{train}(FS_k))$ and test samples $(X_{test}(FS_k))$ using FS_k
6: Training LS-SVM on $(X_{train}(FS_k))$
7: Evaluating FS_k by calculating the ROC AUC of learned LS-SVM using $(X_{test}(FS_k))$
8: $FS_{k+1} = FS_k - \{f_{k+1}\}$
9: **end for**
10: Find the feature set that has the highest ROC AUC as the final selected feature set.

According to these advantages we consider this criterion to evaluate our wrapper classifier LS-SVM. Therefore, in this phase, those features yielding the highest ROC AUC values are selected as the final feature set (Figure 4).

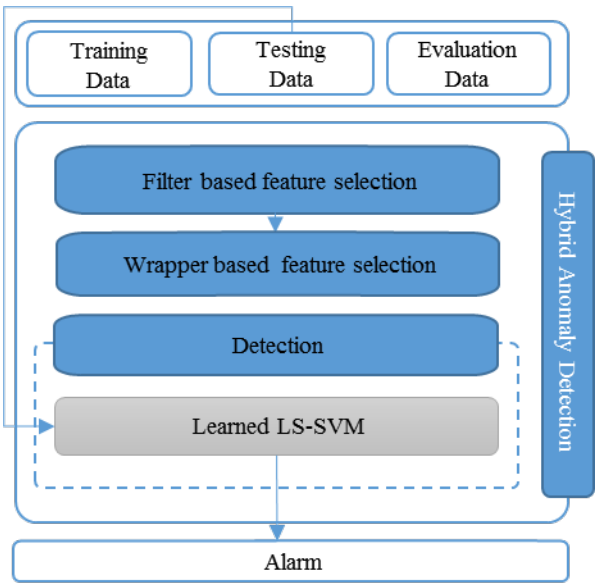

Figure 4. Final LS-SVM classifier

Algorithm 1 presents the Pseudo code of our proposed wrapper-based feature selection method.

In each iteration of the repeating loop, the feature with the lowest PI is eliminated. Therefore, the feature set in each iteration (from 0 to $(n-m)-1$) can be modeled as follows:

$$
\begin{aligned}
&FS_0 = \{f_1, f_2, ..., f_n\} \text{ where} PI\,(f_i) < PI\,(f_j) \text{ and } i < j \\
&FS_1 = FS_0 - \{f_1\} = \{f_2, ..., f_n\} \\
&FS_2 = FS_1 - \{f_2\} = \{f_1, f_2\} = \{f_3, ..., f_n\} \qquad (6) \\
&\dots \\
&FS_{n-m} = FS_{(n-m)-1} - \{f_{n-m}\} = \{f_{n-(m-1)}, ..., f_n\}
\end{aligned}
$$

In the above expressions, FS_0 is corresponding to the output feature set from phase 1. FS_1 to FS_{n-m} are feature sets that obtained through eliminating feature with lowest predictor importance criterion in each loop repetition. It should be noted that m presents the minimum number of features which is allowed.

We choose LS-SVM classifier for evaluating the feature sets. Generally, SVM is a well-known machine learning method used for classification and regression problems. It is a linear binary classifier that can classify the data by finding hyperplane. SVM can also classify the data which are not linearly separable. It uses the kernel to map the data into higher dimensional space in where data can be separable through a hyperplane [44]. One of the distinctions between SVM and other classification methods is that it considers the maximum distance between the separators and the marginal data (Figure 5). In fact, unlike the other neural networks, SVM reduces the operational risk of the target function instead of reducing the classification error. Hence, it can select the separator parameters precisely and optimally, and it does not obtain the separator via trial and error.

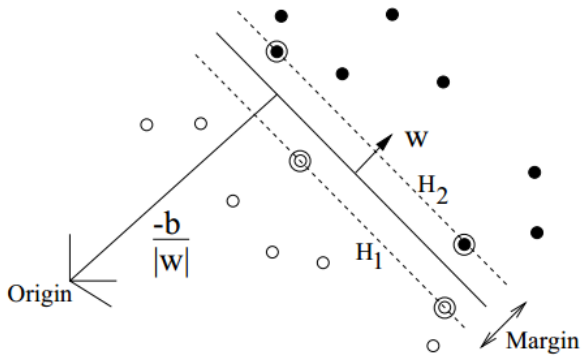

Figure 5. Find the maximum margin between support vectors and find the optimal separator [43]

LS-SVM is a reformulation of the original SVM which using the least square error. The higher speed and accuracy, and the solution to the local optimal problem are the main advantages of the LS-SVM compare to the original SVM [45].

In our proposed architecture, the network traffic data are the input to the IDS, and the LS-SVM classifier is trained after the feature selection is carried out by the decision tree. We use LS-SVM as our model in the second phase named wrapper-based feature selection. Afterward, we use the learned SVM as our detector in the detection phase. Hence, our system can detect different traffic classes to produce alarms when detecting an attack.

4 Experimental Results

In this section, we evaluate our proposed IDS using two well-known datasets, named KDD CUP 99 and UNSW-NB15. Firstly, these datasets and their characteristics are presented. Secondly, the evaluation criteria are introduced. Finally, the experiments and results are presented.

4.1 Datasets

In this paper, the KDD CUP 99 and UNSW-NB15 datasets are used to evaluate the proposed IDS. The KDD CUP 99 dataset was developed by MIT Lincoln laboratories based on the DARPA 1998 intrusion detection evaluation program [46]. This dataset was developed to study and evaluate the researches on intrusion detection. Lincoln laboratory collected the raw TCP traffic data of a local area network (LAN) within nine weeks. The collected data included approximately

five million connection records, and each record represented a TCP/IP connection with 41 features. The features of the KDD CUP 99 dataset were grouped into the following four groups: substantial features, content features, time-based traffic features, and host-based traffic features. The attacks contained in this dataset were also put into five categories, namely DoS, probe, R2L (remote to local), and U2R (user to root). Each group included a specific type of attacks.

Moustafa and Slay stated that the KDD CUP 99 and NSL-KDD datasets were unable to support new attacks by the IDSs. Accordingly, they introduced a dataset based on UNSW-NB15 network [47, 48]. This dataset consisted of 9 different attacks (reconnaissance, shellcode, exploit, fuzzers, worm, DoS, backdoor, analysis, generic, and normal traffic), and each sample has 49 features in UNSW-NB15. This dataset consists of a total of 2540044 records [47]. These 49 features are classified into the following 6 groups: flow features, basic features, content features, time features, additional generated features, and labeled features. The additional generated feature category is divided into two subgroups named general purpose features (features no. 36 to 40) and connection features (features no. 41 to 47) [38].

4.2 Evaluation Criteria

The proposed IDS is evaluated using the three criteria: 1) True Positive Rate (TPR), which is also known as the detection rate and is obtained via the formula (7). 2) False Positive Rate (FPR), which is calculated via formula (8). 3) Accuracy, which is the detection accuracy and is obtained using the formula (9).

$$TPR = TP/(TP + FN) \tag{7}$$
$$FPR = FP/(TN + FP) \tag{8}$$
$$Accuracy = (TN + TP)/(TN + TP + FN + FP) \tag{9}$$

Where TP denotes the number of attacks records detected correctly, and FP stands for the number of attack records which was detected as normal attacks incorrectly. TN also represents the number of normal samples assessed normal, while FN shows the number of abnormal samples considered normal.

4.3 Experiments and Results

The experiments were carried out using a system running Windows 10 with an Intel Core i7 2.80 GHz CPU and 16GB of RAM. We used the LS-SVM Lab toolbox to train the LS-SVM [49]. Given that LS-SVM is a binary classifier, a separate LS-SVM was trained and tested to distinguish each class from the other classes. For instance, in the KDD CUP 99 dataset, an LS-SVM was trained to distinguish between the DoS and non-DoS classes (normal, probe, U2R, and R2L), while another LS-SVM was trained separately for each of the 5 classes in the KDD CUP99 dataset and the 9 classes in the UNSW-NB15 dataset.

Table 1. UNSW-NB15 Selected Features

Class Type	Features No.	Selected Features
Normal	6	36 - 6 - 32 - 33 - 2 - 10
DoS	10	12 - 2 - 30 - 23 - 29 - 41 - 40 - 8 - 5 - 4
Probe	9	2 - 36 - 32 - 30 - 5 - 33 - 37 - 4 - 40
R2L	8	4 - 35 - 40 - 30 - 24 - 2 - 23 - 3
U2R	5	6 - 5 - 32 - 33 - 14

Table 2. UNSW-NB15 Selected Features

Class Type	Features No.	Selected Features
ShellCode	7	4 - 24 - 29 - 14 - 10 - 9 - 23
Reconnaissance	8	24 - 4 - 10 - 12 - 23 - 29 - 14 - 8
Generic	4	8 - 41 - 10 - 2
Fuzzer	15	4 - 10 - 24 - 41 - 13 - 26 - 14 - 11 - 25 16 - 8 - 18 - 29 - 37 - 46
Exploit	15	10 - 41 - 9 - 13 - 16 - 4 - 23 - 12 - 8 - 25 - 28 - 34 - 26 - 14 - 45
Analysis	14	4 - 37 - 11 - 41 - 42 - 29 - 26 - 33 - 16 - 34 - 23 - 38 - 10 - 14
Backdoor	12	2 - 23 - 25 - 13 - 8 - 28 - 14 - 41 - 10 - 45 - 29 - 11
Normal	4	10 - 37 - 4 - 41
DoS	16	2 - 29 - 41 - 8 - 37 - 39 - 25 - 9 - 12 - 15 - 10 - 16 - 7 - 18 - 31 - 4
worm	9	10 - 8 - 23 - 4 - 7 - 34 - 26 - 2 - 18

The KDD Cup 99 and the UNSW-NB15 data sets have more than four and two million records, respectively. These large data sets are not suitable for SVM training. Therefore, we randomly selected 6025 records of KDD Cup 99 as the training set and 7025 records as the testing set as well as 86736 records of UNSW-NB15 as the training set and 122067 instances as testing dataset. Please note that the training data set is used for feature selection in C5. 0 and the LS-SVM training.

The machine learning methods used in our model, namely C5.0 and LS-SVM, are trained using training dataset. The evaluation dataset is used in the second phase of our model, namely wrapper-based feature selection. The final feature set is selected in such a way that the performance of LS-SVM is the best on the test dataset. Finally, the evaluation dataset is used in experiments to evaluate the accomplishment of our proposed IDS.

The number of final selected features in our proposed feature selection method is not the optimal number, but we indicated the best number of features using their importance. We begin with pruning output features and incrementally eliminate the least significant features in phase 2. In each step of phase 2, TPR and FPR criterion are considered, and ROC AUT is

Table 3. Training and Testing runtime per training and testing UNSW-NB15 instance

	Building time(ms)	Test time(ms)
Shellcode	0.3094	0.0633
Reconnaissance	0.0650	0.3520
Generic	0.0577	0.2506
Fuzzers	0.6202	0.0990
Exploit	0.7218	0.1218
Analysis	0.6034	0.1216
Backdoor	0.9101	0.1359
Normal	0.8073	0.1303
DoS	0.7715	0.1627
Worm	0.3853	0.0572

Table 4. Training and Testing runtime per training and testing KDD CUP 1999 instance

	Building time(ms)	Test time(ms)
Normal	0.0904	0.3304
DoS	0.0993	0.1447
Probe	0.1674	0.0517
R2L	0.0683	0.2858
U2R	0.0552	0.3299

obtained according to the distance between the TPR and FPR criterions in LS-SVM with the training data. The ultimate features are those which have maximal ROC AUC value. Table 1 and 2 show the final features, selected for detecting each class of the datasets. A significant reduction in the number of features is clear in these tables. The training and testing runtime of applying our proposed method on UNSW-NB15 and KDD-CUP99 dataset are shown in Table 3 and 4, respectively. Figure 6 and 7 depict the ROC curve for detecting each class in the UNSW-NB15 and KDD-CUP99 datasets.

The KDD CUP 1999 or UNSW-NB15 has been selected as the experimental data set in huge body of literature. Therefore, we compare our proposed IDS with some other works that have used machine learning methods on the KDD CUP 1999 or UNSW-NB15 data sets. Table 5 and 6 present the Accuracy, TPR, FPR and ROC AUCs values for final selected features in LS-SVM using evaluation datasets; also, the results of other studies are shown in these tables to compare with our method based on these criterions. Each row in these tables is considered for evaluation of detecting a particular class type. Please note that the numerical results of the GF-SVM model and proposed feature selection method in [10] were obtained after implementing the model and using their suggested features, and the numerical results of other resources

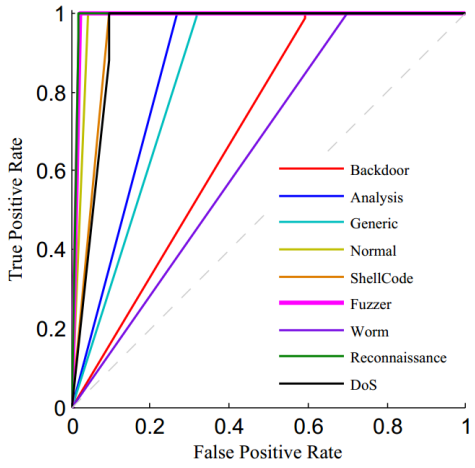

Figure 6. UNSW-NB15 Classes ROC

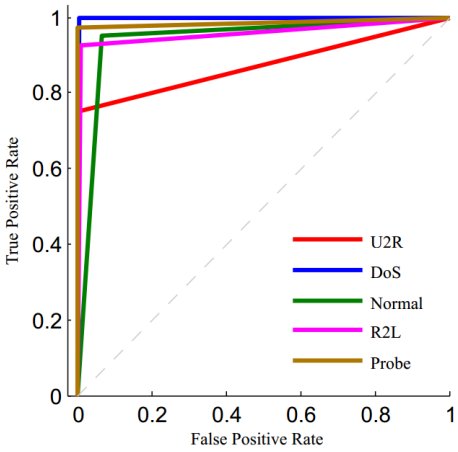

Figure 7. KDD CUP 99 Classes ROC

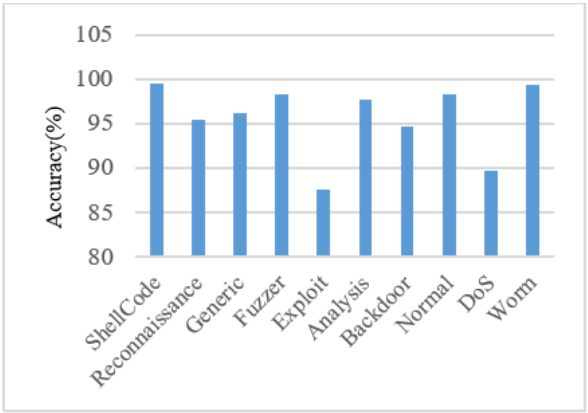

Figure 8. UNSW-NB15 Accuracy

were extracted from the presented tables in each article. Also, in [12], the results are presented for the detection of TCP, UDP and ICMP traffic instances, separately. In order to compare it with our work, we consider the average value of these results. Numerical results show that our method has been successful in identifying all

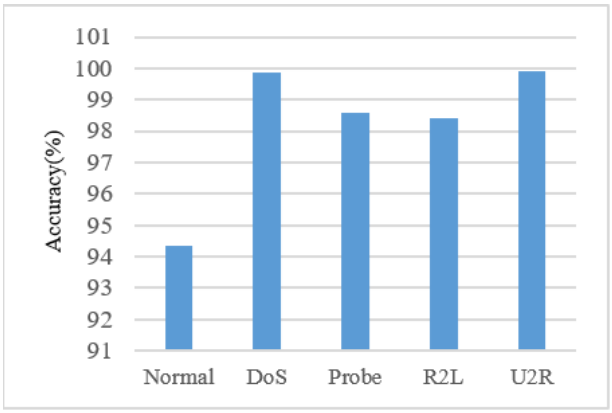

Figure 9. KDD CUP 99 Accuracy

Table 5. KDD CUP 99 Compared Evaluation Values

Class	Model	Accuracy(%)	TPR(%)	FPR(%)	AUC
DoS	DT-LSSVM	99.88	99.8	0.099	**0.9985**
	[28]	99.86	99.8	0.068	**0.9997**
	[35]	99.81	99.8	0.23	**0.9950**
	[10]	99.88	100	23.10	-
	[12]	92.2	-	-	-
	[29]	92.95	-	-	-
	[50]	92.55	-	-	-
Probe	DT-LSSVM	98.60	97.24	0.039	**0.9860**
	[28]	95.16	93.80	3.86	**0.9516**
	[35]	97.83	96.08	0.44	**0.9782**
	[10]	93.57	90.92	3.80	-
	[12]	82.98	-	-	-
	[29]	97.825	-	-	-
	[50]	96.6	-	-	-
R2L	DT-LSSVM	98.42	92.56	0.74	**0.9595**
	[28]	95.75	84.35	2.59	**0.9087**
	[35]	95.47	79.07	2.15	**0.8846**
	[10]	96.95	84.19	1.20	-
	[12]	100 (test with 1 record)	-	-	-
	[29]	90.975	-	-	-
	[50]	90.2	-	-	-
U2R	DT-LSSVM	99.891	75	0	**0.8750**
	[28]	99.891	75	0	**0.8750**
	[35]	99.891	75	0	**0.8750**
	[10]	99.78	65.00	0.07	-
	[12]	75.0	-	-	-
	[29]	98.425	-	-	-
	[50]	84.98	-	-	-

Class	Model	Accuracy(%)	TNR(%)	FNR(%)	AUC
Normal	DT-LSSVM	94.35	95.27	6.57	**0.9435**
	[28]	93.19	98.71	12.37	**0.9317**
	[35]	89.529	86.318	7.235	**0.8967**
	[10]	89.09	80.07	1.80	-
	[12]	92.76	-	-	-
	[29]	99.375	-	-	-
	[50]	96.12	-	-	-

Table 6. UNSW-NB15 Compared Evaluation Values

Class	Model	Accuracy(%)	TPR(%)	FPR(%)	AUC
ShellCode	DT-LSSVM	99.459	100	9.69	0.9516
	[38]	94.41	100	100	0.996
	[28]	99.30	100	12.50	0.9375
	[47]	94.66	97.45	52.50	-
Reconnaissance	DT-LSSVM	95.37	93.41	2.03	0.990
	[38]	93.54	90.50	2.45	0.977
	[28]	89.54	88.39	8.93	0.9346
	[47]	90.46	87.98	06.26	-
Generic	DT-LSSVM	96.12	98.96	7.42	0.9911
	[38]	94.01	96.77	9.45	0.996
	[28]	85.51	99.26	30.17	0.8402
	[47]	97.29	96.49	1.70	-
Fuzzer	DT-LSSVM	98.27	98.60	2.04	0.990
	[38]	96.19	96.20	3.80	0.961
	[28]	96.76	97.38	3.84	0.9803
	[47]	96.06	94.34	2.27	-
Exploit	DT-LSSVM	87.47	87.72	12.83	0.959
	[38]	83.52	85.09	18.40	0.950
	[28]	79.19	67.31	6.23	0.782
	[47]	84.29	88.27	20.57	-
Analysis	DT-LSSVM	97.69	99.92	27.00	0.866
	[38]	93.59	99.57	82.29	0.50
	[28]	-	-	-	-
	[47]	93.19	99.53	87.14	-
Backdoor	DT-LSSVM	94.67	98.94	59.43	0.70
	[38]	93.59	99.59	82.29	0.809
	[28]	-	-	-	-
	[47]	93.19	99.53	87.14	-
DoS	DT-LSSVM	89.65	90.23	10.86	0.9470
	[38]	90.10	92.17	11.73	0.911
	[28]	83.45	92.89	24.91	0.9451
	[47]	80.056	92.09	29.46	-
Worm	DT-LSSVM	99.33	100	70	0.6285
	[38]	99.05	100	100	0.50
	[28]	-	-	-	0.996
	[47]	99.21	99.98	80	-

Class	Model	Accuracy(%)	TNR(%)	FNR(%)	AUC
Normal	DT-LSSVM	98.29	100	4.38	0.9781
	[38]	98.23	99.90	4.38	0.987
	[28]	41.13	6.79	4.98	0.9519
	[47]	41.13	6.79	4.98	-

classes and has yielded significant results. The results of the comparisons show that the proposed method has achieved better results rather than compared methods in DoS, Probe, R2L, and U2R classes in KDD CUP 1999 and ShellCode, Reconnaissance, Fuzzer, Exploit, Analysis, Backdoor, Normal, Worm classes in UNSW-NB15 data set.

5 Conclusion and Future Work

In this article, an IDS was proposed based on a combination of the C5.0 decision tree methods. In the DT-LSSVM, a decision tree was used for feature selection, because it offers significant advantages in terms of obtaining the important features through classification. We used C5.0 as the well-known version of the decision trees. We used the gain ratio in the expansion of tree, the error-based pruning algorithm to prune the obtained decision tree, and PI criterion to select the feature set in the first phase of our model, named filter-based feature selection. In addition, we utilized LS-SVM, as one of the most accurate and reliable classifiers in the second phase of our model. Using both filter-based and wrapper-based feature selection methods led to the high TPR and low detection error rate. The experiments conducted on the KDD CUP99 and UNSW-NB15 datasets are indicative of an increase in the detection accuracy and TPR and a decrease in the detection error rate using a combination of the C5.0 and LS-SVM algorithms. The use of the proposed feature selection method along with other classifiers will be explored in future research. In addition, since the decision tree pruning algorithms are set to find the most effective feature in each class, the use of various decision tree pruning algorithms and a combination of the algorithms will be studied with the aim of selecting the best features.

References

[1] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[2] Shraddha Khonde and V Ulagamuthalvi. A machine learning approach for intrusion detection using ensemble technique-a survey. 2018.

[3] Asish Kumar Dalai and Sanjay Kumar Jena. Hybrid network intrusion detection systems: A decades perspective. In *Proceedings of the International Conference on Signal, Networks, Computing, and Systems*, pages 341–349. Springer, 2017.

[4] Lin Li Zhong, Zhang Ya Ming, and Zhang Yu Bin. Network intrusion detection method by least squares support vector machine classifier. In *2010 3rd International Conference on Computer Science and Information Technology*, volume 2, pages 295–297. IEEE, 2010.

[5] Pablo Bermejo, Luis de la Ossa, José A Gámez, and José M Puerta. Fast wrapper feature subset selection in high-dimensional datasets by means of filter re-ranking. *Knowledge-Based Systems*, 25(1):35–44, 2012.

[6] Yinhui Li, Jingbo Xia, Silan Zhang, Jiakai Yan, Xiaochuan Ai, and Kuobin Dai. An efficient intrusion detection system based on support vector machines and gradually feature removal method. *Expert Systems with Applications*, 39(1):424–430, 2012.

[7] Sevcan YILMAZ GÜNDÜZ and Muhammet Nurullah ÇETER. Feature selection and comparison of classification algorithms for intrusion detection. *Anadolu University of Sciences & Technology-A: Applied Sciences & Engineering*, 19(1), 2018.

[8] Zohreh Abtahi Foroushani and Yue Li. Intrusion detection system by using hybrid algorithm of data mining technique. In *Proceedings of the 2018 7th International Conference on Software and Computer Applications*, pages 119–123. ACM, 2018.

[9] Chibuzor John Ugochukwu, EO Bennett, and P Harcourt. An intrusion detection system using machine learning algorithm. *International Journal of Computer Science and Mathematical Theory*, 4(1):2545–5699, 2018.

[10] Nachiket Sainis, Durgesh Srivastava, and Rajeshwar Singh. Feature classification and outlier detection to increased accuracy in intrusion detection system. *International Journal of Applied Engineering Research*, 13(10):7249–7255, 2018.

[11] Chandrashekhar Azad and Vijay Kumar Jha. Decision tree and genetic algorithm based intrusion detection system. In *Proceeding of the Second International Conference on Microelectronics, Computing & Communication Systems (MCCS 2017)*, pages 141–152. Springer, 2019.

[12] Shaohua Teng, Naiqi Wu, Haibin Zhu, Luyao Teng, and Wei Zhang. Svm-dt-based adaptive and collaborative intrusion detection. *IEEE/CAA Journal of Automatica Sinica*, 5(1):108–118, 2017.

[13] Snehal A Mulay, PR Devale, and GV Garje. Intrusion detection system using support vector machine and decision tree. *International Journal of Computer Applications*, 3(3):40–43, 2010.

[14] T Augustine, P Vasudeva Reddy, and PVGD Prasad Reddy. A frame work for performance evaluation of classifiers: Case study on nids. *International Journal of Pure and Applied Mathematics*, 118(20):973–984, 2018.

[15] Kathleen Goeschel. Reducing false positives in intrusion detection systems using data-mining techniques utilizing support vector machines, decision trees, and naive bayes for off-line analysis. In *SoutheastCon 2016*, pages 1–6. IEEE, 2016.

[16] Ngoc Tu Pham, Ernest Foo, Suriadi Suriadi, Helen Jeffrey, and Hassan Fareed M Lahza. Improving performance of intrusion detection system using ensemble methods and feature selection. In *Proceedings of the Australasian Computer Science Week Multiconference*, page 2. ACM, 2018.

[17] Anuradha S Varal and SK Wagh. Misuse and

anomaly detection using ensemble learning network traffic model.

[18] S Latha and Sinthu Janita Prakash. Hpfsm-a high pertinent feature selection mechanism for intrusion detection system. *International Journal of Pure and Applied Mathematics*, 118(9):77–83, 2018.

[19] Ping Wang, Kuo-Ming Chao, Hsiao-Chung Lin, Wen-Hui Lin, and Chi-Chun Lo. An efficient flow control approach for sdn-based network threat detection and migration using support vector machine. In *2016 IEEE 13th International Conference on e-Business Engineering (ICEBE)*, pages 56–63. IEEE, 2016.

[20] Phurivit Sangkatsanee, Naruemon Wattanapongsakorn, and Chalermpol Charnsripinyo. Practical real-time intrusion detection using machine learning approaches. *Computer Communications*, 34(18):2227–2235, 2011.

[21] Meesala Shobha Rani and S Basil Xavier. A hybrid intrusion detection system based on c5. 0 decision tree and one-class svm. *International journal of current engineering and technology*, 5(3):2001–2007, 2015.

[22] L Breiman, JH Friedman, RA Olshen, and CJ Stone. Classification and regression trees (monterey, ca: Wadsworth and brooks/cole). *Links*, 1984.

[23] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[24] E Serkani, H Gharaee Garakani, N Mohammadzadeh, and E Vaezpour. Hybrid anomaly detection using decision tree and support vector machine. *International Journal of Electrical, Electronic and Communication Sciences*, page 6, 2018.

[25] Lior Rokach and Oded Z Maimon. *Data mining with decision trees: theory and applications*, volume 69. World scientific, 2008.

[26] Max Kuhn and Kjell Johnson. *Applied predictive modeling*, volume 26. Springer, 2013.

[27] Hossein Gharaee and Maryam Fekri. A new feature selection for intrusion detection system. *International Journal of Academic Research*, 7, 2015.

[28] Hossein Gharaee and Hamid Hosseinvand. A new feature selection ids based on genetic algorithm and svm. In *2016 8th International Symposium on Telecommunications (IST)*, pages 139–144. IEEE, 2016.

[29] Aditi Nema, Basant Tiwari, and Vivek Tiwari. Improving accuracy for intrusion detection through layered approach using support vector machine with feature reduction. In *Proceedings of the ACM Symposium on Women in Research 2016*, pages 26–31. ACM, 2016.

[30] Peiying Tao, Zhe Sun, and Zhixin Sun. An improved intrusion detection algorithm based on ga and svm. *IEEE Access*, 6:13624–13631, 2018.

[31] Praneeth Nskh, M Naveen Varma, and Roshan Ramakrishna Naik. Principle component analysis based intrusion detection system using support vector machine. In *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pages 1344–1350. IEEE, 2016.

[32] Angela Denise Landress. A hybrid approach to reducing the false positive rate in unsupervised machine learning intrusion detection. In *SoutheastCon 2016*, pages 1–6. IEEE, 2016.

[33] Tahir Mehmood and Helmi B Md Rais. Svm for network anomaly detection using aco feature subset. In *2015 International symposium on mathematical sciences and computing research (iSMSC)*, pages 121–126. IEEE, 2015.

[34] Wang Xingzhu. Aco and svm selection feature weighting of network intrusion detection method. *International Journal of Security and its Applications*, 9(4):129–270, 2015.

[35] Fatemeh Amiri, MohammadMahdi Rezaei Yousefi, Caro Lucas, Azadeh Shakery, and Nasser Yazdani. Mutual information-based feature selection for intrusion detection systems. *Journal of Network and Computer Applications*, 34(4):1184–1199, 2011.

[36] Ujwala Ravale, Nilesh Marathe, and Puja Padiya. Feature selection based hybrid anomaly intrusion detection system using k means and rbf kernel function. *Procedia Computer Science*, 45:428–435, 2015.

[37] Sandhya Peddabachigari, Ajith Abraham, Crina Grosan, and Johnson Thomas. Modeling intrusion detection system using hybrid intelligent systems. *Journal of network and computer applications*, 30(1):114–132, 2007.

[38] Tharmini Janarthanan and Shahrzad Zargari. Feature selection in unsw-nb15 and kddcup'99 datasets. In *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, pages 1881–1886. IEEE, 2017.

[39] Taimur Bakhshi and Bogdan Ghita. On internet traffic classification: A two-phased machine learning approach. *Journal of Computer Networks and Communications*, 2016, 2016.

[40] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[41] Rokach Lior et al. *Data mining with decision trees: theory and applications*, volume 81. World scientific, 2014.

[42] Max Kuhn and Kjell Johnson. *Applied predictive modeling*, volume 26. Springer, 2013.

[43] Christopher JC Burges. A tutorial on support

vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[44] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[45] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[46] M Lincoln. Kdd cup 99. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99. 1999.

[47] Nour Moustafa and Jill Slay. The significant features of the unsw-nb15 and the kdd99 data sets for network intrusion detection systems. In *2015 4th international workshop on building analysis datasets and gathering experience returns for security (BADGERS)*, pages 25–31. IEEE, 2015.

[48] J. S Nour Moustafa. The unsw-nb15 data set. https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/, 2015.

[49] K. L University. Ls-svm lab toolbox. https://www.esat.kuleuven.be/sista/lssvmlab/.

[50] Ramandeep Kaur and Meenakshi Bansal. Multidimensional attacks classification based on genetic algorithm and svm. In *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, pages 561–565. IEEE, 2016.

Elham Serkani received B.S. degree in computer engineering from Hamedan University of Technology, in 2012, M.S. degree in information technology engineering from Shahed University, Tehran, Iran, in 2018. Her research interest includes network and information security.

Hossein Gharaee received B.S. degree in electrical engineering from Khaje Nasir Toosi University, in 1998, M.S. and Ph.D. degree in electrical engineering from Tarbiat Modares University, Tehran, Iran, in 2000 and 2009 respectively. Since 2009, he has been with the department of network technology in Iran Telecom Research Center (ITRC). His research interests include general area of VLSI with emphasis on basic logic circuits for low-voltage low-power applications, DSP Algorithm, crypto chip and Intrusion detection and prevention systems.

Naser Mohammadzadeh received the B.S. and M.S. degrees in computer engineering from Sharif University of Technology, Iran. He received the Ph.D. degree in computer engineering from Amirkabir University of Technology, Iran. His research interests include optimization and quantum design automation.