World Academy of Science, Engineering and Technology
International Journal of Electrical and Computer Engineering
Vol:12, No:6, 2018

Hybrid Anomaly Detection Using Decision Tree and Support Vector Machine

Elham Serkani, Hossein Gharaee Garakani, Naser Mohammadzadeh, Elaheh Vaezpour

Abstract**—**Intrusion detection systems (IDS) are the main components of network security. These systems analyze the network events for intrusion detection. The design of an IDS is through the training of normal traffic data or attack. The methods of machine learning are the best ways to design IDSs. In the method presented in this article, the pruning algorithm of C5.0 decision tree is being used to reduce the features of traffic data used and training IDS by the least square vector algorithm (LS-SVM). Then, the remaining features are arranged according to the predictor importance criterion. The least important features are eliminated in the order. The remaining features of this stage, which have created the highest level of accuracy in LS-SVM, are selected as the final features. The features obtained, compared to other similar articles which have examined the selected features in the least squared support vector machine model, are better in the accuracy, true positive rate, and false positive. The results are tested by the UNSW-NB15 dataset.

Keywords**—**Intrusion detection system, decision tree, support vector machine, feature selection.

I. Introduction

IN recent years, the expansion of computer networks and communications systems, has paved the way for intruders. Therefore, IDSs are increasingly important against threats and attacks. The IDSs help to detect unauthorized use, modify, and destroy information systems and networks.

IDSs are divided into two types based on their locations: network-based IDS and host-based IDS [1]. Host-based IDSs monitor the data and processes of a particular host's software environment. Network-based IDS detects the attacks through the network traffic monitoring. Detection methods in the IDS are grouped into three categories: signature based, anomaly based and hybrid methods [1]. In the signature based methods, detection is performed by matching the new traffic data with the pattern of known attacks stored under the title of the signature in a database. Anomaly based methods, after learning the normal behavior of the system, detect any deviation from the normal traffic of the host or network. The main advantage of these systems is the ability to detect unknown attacks compared to signature-based systems.

It is very useful to use machine learning methods to train and build the model of IDSs [1]. One of the most important methods of machine learning can be referred to support vector machine (SVM) [2]. To benefit from other methods of machine learning to improve the performance of SVMs, in many studies, SVMs are used in combination with other machine learning methods [3]. Many machine learning methods can be used to perform the classification alongside SVMs to enhance the performance of SVM.

For this purpose, Yang et al. have optimized the parameters of SVM by the PSO[1] algorithm [4]. In the work of Kaur and Bansal, the data are split into two normal and attack groups by the genetic algorithm, then, the exact type of attack is determined by SVM [5].

Many methods can also be used as preprocessors to reduce data attributes before SVM. Genetic algorithm [6]-[8], rough set theory [9], principal component analysis (PCA) [10], Information Gain criterion in ID3 decision tree algorithm [11], mutual information [12], ant colony optimization [13], all algorithms are used prior to the training of SVMs to select the best features.

As noted, the decision tree is one of the methods that can be used to select the features. In the proposed method by Landress, the feature selection is performed by the decision tree C4.5, and then, the model is constructed using the k-near neighbors and the self-organizing map (SOM) [14].

In this article, the proposed method is performed in two phases of preprocessing and model construction. In the first phase, the pruning method of C5.0 decision tree algorithm is used to select the attributes. In the next phase, training the IDS, is provided by the least-square SVM (LS-SVM) with the radial basis kernel function (RBF). After testing the LS-SVM model, its efficiency measures are calculated. Then, the most important predictor criterion is considered for features that remain after pruning. With the aim of finding the smallest number of features, we remove features with the least important criterion in a loop; and each time we perform the feature selection with the remaining features, then we train and test LS-SVM. The features that lead to the highest accuracy for the LS-SVM are selected as the final features.

The possibility of Error-Based Pruning (EBP) pruning in the decision tree C5.0 algorithm, as well as the ability to determine the predictor importance criterion, makes it suitable for choosing the appropriate attribute [15].

Due to effect of the kernel function on SVM results, the RBF function is selected as the best kernel function for IDS [10].

Previous studies show that the LS-SVM with the RBF kernel, because of solving the local optima problem [16], has better performance and higher detection rates and accuracy

Hossein Gharaee Garakani and Elaheh Vaezpour are with Iran Telecommunication Research Center – ITRC, Tehran, Iran (e-mail: gharaee@itrc.ac.ir; e.vaezpour@itrc.ac.ir).

Elham Serkani and Naser Mohammadzadeh are with Shahed Univercity, Tehran, Iran (corresponding author, phone: +982151212098; fax: +982151212021; e-mail: mohammadzadeh@shahed.ac.ir; e.serkani.e@gmail.com).

[1] particle swarm optimization

[17]. In this article, the UNSW-NB15 dataset is used to build IDS. This dataset contains 10 classes, including one normal and 9 attack types. This dataset contains 10 classes including normal and 9 attack types including Analysis, Backdoor, DoS, Exploit, Generic, Reconnaissance, Shellcode, Worm.

Each sample of this dataset is defined by 49 features. In the following, decision trees and SVMs with least squares are presented in Sections II and III, respectively. Then, the presented method is described in Section IV. Finally, Sections V and VI present the results of the experiments and the conclusion of the article.

II. C5.0 Decision Tree Algorithm

The decision tree method is one of the most commonly used methods of inductive learning and machine learning for classification and regression issues. The C5.0 algorithm is the newest version of the ID3 algorithm that was introduced after C4.5. The algorithm uses the Gain Ratio Criterion and Error-Based Pruning (EBP) that is based on pessimistic pruning method. The algorithm can use boosting and winnowing possibilities [18].

A. Attribute Selection Criteria at Each Stage of Tree Construction

At each stage of tree construction, a certain criterion is used to select the attribute. This criterion varies in different decision tree algorithms. The criterion used in the C5.0 is the Gain Ratio, which is calculated based on the ability of each attribute to reduce the entropy of the dataset. At each step, the feature is selected with the highest Gain Ratio.

The Gain Ratio criterion is the normalized value of the information gain criterion in the ID3 decision tree algorithm; it is calculated by the following formula:

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

In this formula and subsequent formulas, S is the dataset in which entropy is computed, and S_i is the probability of the existence of samples in class i. P is the ratio of the number of class x members to the total number of S dataset samples. c is equal to the number of classes in the S dataset.

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

In this formula, Value A is the set of all values of the characteristic A; and S_v is a subset of S, in which the value of the Feature A is equal to v for all member instances. In (2), the first statement is equal to the amount of entropy of the data, and the second expression is equal to the entropy value, after separating the data by the feature A.

The Gain Ratio criterion is calculated by (3):

$$GainRatio(S,A) = \frac{IG(S,A)}{SplitInformation(S,A)} \tag{3}$$

In this formula, the existence of split information in the denominator causes the features that have a large number of values with uniform distribution to be deleted. This value is obtained by (4).

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

B. Decision Tree Pruning

The pruning in the decision tree means the removal of the nodes and extra branches of the tree. Removing these nodes or branches is aimed at preventing over fitting in learning and eliminating the distorting features nodes of the tree that reduce the accuracy of the tree. Pruning methods are generally divided into pre-pruning and post-pruning methods. In C5.0, an advanced version of error-pruning, which is a post-pruning technique, is implemented [15]. After making the full decision tree of C5.0, pruning is used to remove unnecessary features.

III. Least Squares Support Vector Machine

The SVM is a machine learning method used for classification and regression issues. LS-SVM is a rewording of the original SVM. The higher speed and accuracy, and the solution to the local optimal problem, is the main advantage of the LS-SVM towards the original SVM. The least squared SVM formulas are as follows:

$$y(x) = w^T \varphi(x_i) + b \tag{5}$$

In this expression, w and b are the model parameters (super-pages), and φ (.) is the kernel function to convert the feature space to a higher dimensional space. x_i is the *i-th* sample with p features. The y_i is the corresponding class of the i-th instance. With N training samples in dataset, the variables w, b, and e are calculated on the hyper-plane:

$$min_{w,b,e} j(w,b,e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum\nolimits_{i=1}^{N} e_i^2 \hat{a} \tag{6}$$

where:

$$y_i [w^T \varphi(x_i) + b] = 1 - e_i \quad i = 1, \ldots, N \tag{7}$$

This optimization problem can be solved in a dual space. To solve this problem, Lagrange equation and its coefficients are defined as follows:

$$\tau(w,b,e;\alpha) = j(w,b,e) - \sum\nolimits_{i=1}^{N} \alpha_i \{ y_i [w^T \varphi(x_i) + b] - 1 + e_i \} \tag{8}$$

$$\frac{\partial\tau}{\partial w} = 0 \rightarrow \sum_{i=1}^{N} \alpha_i y_i \varphi(x_i)$$

$$\frac{\partial\tau}{\partial b} = 0 \rightarrow \sum_{i=1}^{N} \alpha_i y_i = 0$$

$$\frac{\partial\tau}{\partial \alpha_i} = 0 \rightarrow y_i [w^T \varphi(x_i) + b] - 1 + e_i = 0 \qquad (9)$$

$$i = 1, \ldots, N$$

The result can be obtained by the following linear equations:

$$\left[\begin{array}{ccc|c} I & 0 & 0 & -Z^T \\ 0 & 0 & 0 & -Y^T \\ 0 & 0 & YI & -I \\ \hline Z & Y & I & 0 \end{array}\right] \left[\begin{array}{c} \omega \\ b \\ e \\ \hline \propto \end{array}\right] = \left[\begin{array}{c} 0 \\ 0 \\ 0 \\ \hline \vec{1} \end{array}\right] \qquad (10)$$

IV. The Proposed Method

The architecture of the proposed IDS based on decision tree and SVM is shown in Fig. 1.

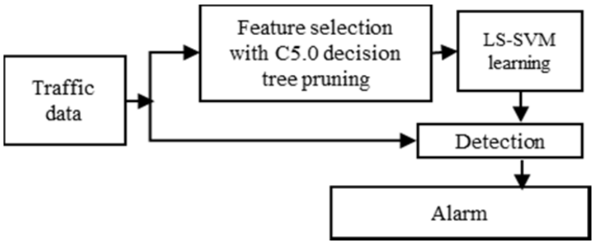

Fig. 1 System architecture of the proposed IDS

We called this method DT-SVM[2]. As shown in Fig. 1, the input of this system is network traffic, and its output is to alert when an attack is detected.

A. Phase 1: Feature Selection with Decision Tree Pruning

Fig. 2 shows the feature selection flowchart. The input of this stage is the traffic data after the feature is extracted. We use the UNSW-NB15 dataset in this article; where each instance is defined as a vector of features. In this step, the decision tree C5.0 is created by these data, then, based on the advanced error-based pruning technique, branches and noise rules (paths from root to leaf) or low impact are removed. The output of this stage is the remaining attributes after pruning and the importance of each feature in classifying instances of different classes.

B. Phase 2: Optimize the Number of Features

This phase is shown in Figs. 2 and 3. The steps are as follows:

1) Calculate the Predictor Importance Criterion for Each Feature

In this step, the predictor importance criterion is calculated for every remaining attribute of the pruning step.

2) Sorting Features Based on Predictor Importance Criterion

In this step of phase 2, features are sorted according to predictor importance criterion calculated values.

3) Applying Feature Selection to the Training and Testing Datasets

At this point, the feature selection is applied to the training and test the dataset. Then, the LS-SVM model is built and tested with these data (Fig. 3).

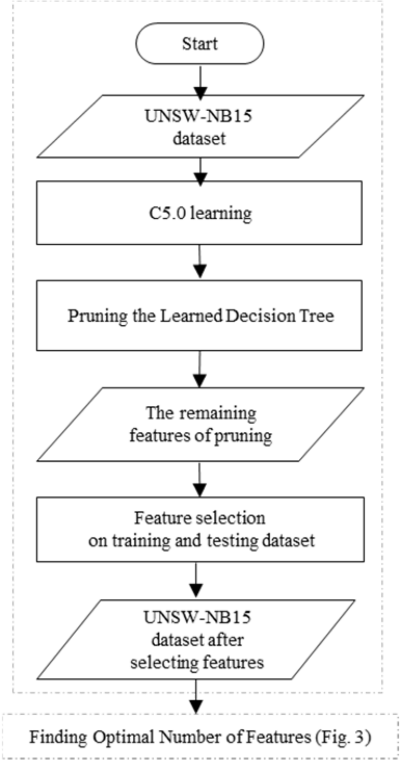
Start
UNSW-NB15 dataset
C5.0 learning
Pruning the Learned Decision Tree
The remaining features of pruning
Feature selection on training and testing dataset
UNSW-NB15 dataset after selecting features
Finding Optimal Number of Features (Fig. 3)

Fig. 2 Feature Selection with Pruning

4) Feature Deletion with the Lowest Predictor Importance Criterion

The least important feature is eliminated based on the predictive value criterion and the method returns to step 3. These steps are repeated as long as all the features are removed.

5) Ultimate Features

Finally, the set of features that has given the most accuracy in LS-SVM is selected as the final features.

V. Experiments and Results

Experiments are conducted on the Windows 10 operating system, with Intel Core i7 2.80 GHz and 16 GB of memory. We use the LS-SVM Lab toolbox to build a least squared SVM.

A. Results and Discussion

In the method presented in this article, the decision tree C5.0 is made based on the UNSW-NB15 dataset data. Then, the pruning of the decision tree is done to remove the

[2] Decision Tree based SVM

inadequate or ineffective features in classifying the dataset. To find the minimum number of possible attributes, features are sorted according to the predictor importance criterion. After applying attribute selection to data, SVM is taught by these data.

TABLE I
SELECTED FEATURES FOR UNSW-NB15

Class	Model	Features Number	Selected Features
ShellCode	**DT_SVM**	**7**	**4 - 24 - 29 - 14 - 10 - 9 - 23**
	[19]	5	14 - 8 - 10 - 23 - 46
	[7]	7	4-10-14 -23 - 37 - 44 - 45
	[20]	11	6-9-10-12 - 13 - 14 - 15 - 16 - 17 - 18 - 23
Reconnaissance	**DT_SVM**	**8**	**24 - 4 - 10 - 12 - 23 - 29 - 14 - 8**
	[19]	5	14 - 8 - 10 - 23 - 46
	[7]	14	10 - 14 - 19 - 20 - 27 - 30 - 31 - 34 - 42 - 43 - 44 - 45 - 46 - 47
	[20]	11	10 - 14 - 37 - 41 - 42 - 43 - 44 - 9 - 16 - 17 - 28
Generic	**DT_SVM**	**4**	**8 - 41 - 10 - 2**
	[19]	5	14 - 8 - 10 - 23 - 46
	[7]	9	10 - 11 - 19 - 23 - 28 - 31 - 33 - 34 - 46
	[20]		6 - 9 - 10 - 11 - 12 - 13 - 15 - 16 - 17 - 18 - 20
Fuzzer	**DT_SVM**	**15**	**4 - 10 - 24 - 41 - 13 - 26 - 14 - 11 - 25 16 - 8 - 18 - 29 - 37 - 46**
	[19]	5	14 - 8 - 10 - 23 - 46
	[7]	13	2 - 4 - 10 - 14 - 28 - 29 - 31 - 41 - 43 - 44 - 45 - 46 - 47
	[20]	11	6 - 11 - 14 - 15 - 16 - 36 - 37 - 39 - 40 - 41 - 42
Exploit	**DT_SVM**	**15**	**10 - 41 - 9 - 13 - 16 - 4 - 23 - 12 - 8 - 25 - 28 - 34 - 26 - 14 - 45**
	[19]	5	14 - 8 - 10 - 23 - 46
	[7]	6	13 - 14 - 16 - 17 - 31 - 33
	[20]	11	10 - 41 - 42 - 6 - 37 - 46 - 11 - 19 - 36 - 5 - 45
Analysis	**DT_SVM**	**14**	**4 - 37 - 11 - 41 - 42 - 29 - 26 - 33 - 16 - 34 - 23 - 38 - 10 - 14**
	[19]	5	14 - 8 - 10 - 23 - 46
	[7]	-	-
	[20]	11	6 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 34 - 35 - 37
Backdoor	**DT_SVM**	**12**	**2 - 23 - 25 - 13 - 8 - 28 - 14 - 41 - 10 - 45 - 29 - 11**
	[19]	5	14 - 8 - 10 - 23 - 46
	[7]	-	-
	[20]	11	6 - 10 - 11 - 14 - 15 - 16 - 37 - 41 - 42 - 44 - 45
Normal	**DT_SVM**	**4**	**10 - 37 - 4 - 41**
	[19]	5	14 - 8 - 10 - 23 - 46
	[7]	7	1 - 2 - 15 - 18 - 21 - 29 - 31
	[20]	11	11 - 34 - 19 - 20 - 21 - 37 - 6 - 10 - 11 - 36 - 47
DoS	**DT_SVM**	**16**	**2 - 29 - 41 - 8 - 37 - 39 - 25 - 9 - 12 - 15 - 10 - 16 - 7 - 18 - 31 - 4**
	[19]	5	14 - 8 - 10 - 23 - 46
	[7]	12	10 - 13 - 14 - 15 - 17 - 23 - 31 - 42 - 43 - 44 - 45 - 47
	[20]	11	6 - 11 - 15 16 - 36 - 37 - 39 - 40 - 42 - 44 - 45
worm	**DT_SVM**	**9**	**10 - 8 - 23 - 4 - 7 - 34 - 26 - 2 - 18**
	[19]	5	14 - 8 - 10 - 23 - 46
	[7]	-	-
	[20]	11	41 - 37 - 9 - 11 - 10 - 46 - 23 - 17 - 14 - 5 - 13

We then delete the attribute with the least amount of predictor importance, and LS-SVM will be trained with the training data again. The feature deletion phase and the LS-SVM learning are repeated as long as all remaining attributes of pruning phase are removed. The features where LS-SVM is most accurate with them are selected as the ultimate features. Results of the important features in each class type for UNSW-NB15 dataset have been shown in Table II.

B. Comparison DT-SVM with Other Techniques

This section shows performance comparison of our proposed DT-SVM model with tree other intrusion detection techniques introduced in [19], [7], [20]. To this end, we examined the feature set obtained in [19], [7], [20] in the LS-SVM algorithm. The features and performance review results of each reference are presented in TABLE I and

In the above statements, *TP* is the number of attack samples that are correctly detected, *FP* is the number of attack samples that are detected normal, *TN* is the number of normal samples that are detected normal, and *FN* defines the number of abnormal samples that are detected normal.

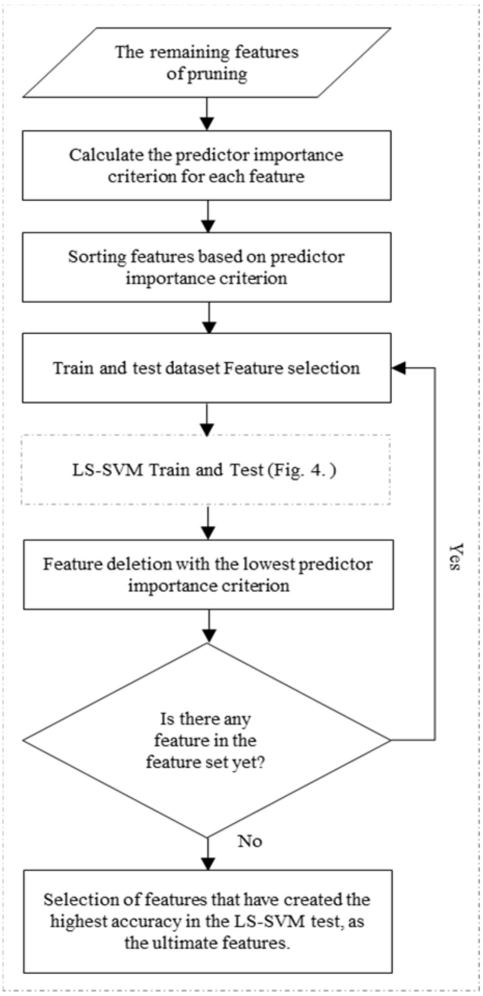
The remaining features of pruning
Calculate the predictor importance criterion for each feature
Sorting features based on predictor importance criterion
Train and test dataset Feature selection
LS-SVM Train and Test (Fig. 4.)
Feature deletion with the lowest predictor importance criterion
Is there any feature in the feature set yet?
Yes
No
Selection of features that have created the highest accuracy in the LS-SVM test, as the ultimate features.

Fig. 3 Finding optimal number of features

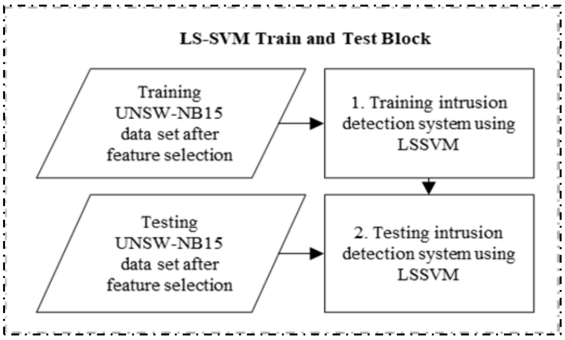
LS-SVM Train and Test Block
Training UNSW-NB15 data set after feature selection
1. Training intrusion detection system using LSSVM
Testing UNSW-NB15 data set after feature selection
2. Testing intrusion detection system using LSSVM

Fig. 4 LS-SVM train and test

VI. Conclusions and Suggestions

In this article, the IDS based on the combination of C5.0 decision tree and SVM is presented. The pruning algorithm of Tree Decision eliminates additional features. Then, the features are sorted according to the predictor importance criterion. We eliminate the features that have the smallest criterion of importance for prediction. We have eliminated the features that have the lowest value for the predictive importance criterion. After deleting all the features, we apply the remaining features to the dataset. Then, LS-SVM is created by this dataset. The features that create the highest accuracy for the LS-SVM are selected as the ultimate features.

Experiments and numerical results are derived from False Positive Rate, True Positive Rate and Accuracy calculations on the UNSW-NB15 dataset.

The results obtained with the GF-SVM model improve the detection accuracy to 99.459 % for Shellcode class, 97.69% for Analysis class, 87.47% for Exploit class, 98.27% for Fuzzer class, 93.72% for Reconnaissance, 99.33 for Worm attack, 94.67 for Backdoor, 97.68 for Generic, 90.85 for DoS attack and 98.29 for normal traffic in UNSW-NB15 Dataset.

TABLE III
Performance

Class	Model	Accuracy (%)	TPR (%)	FPR(%)
ShellCode	**DT-SVM**	**99.459**	**100**	**9.69**
	[19]	94.41	100	100
	[7]	99.30	100	12.50
	[20]	94.66	97.45	52.50
Reconnaissance	**DT-SVM**	**95.37**	**93.41**	**2.03**
	[19]	93.54	90.50	2.45
	[7]	89.54	88.39	8.93
	[20]	90.46	87.98	06.26
Generic	**DT-SVM**	**96.12**	**98.96**	**7.42**
	[19]	94.01	96.77	9.45
	[7]	85.51	99.26	30.17
	[20]	97.29	96.49	1.70
Fuzzer	**DT-SVM**	**98.27**	**98.60**	**2.04**
	[19]	96.19	96.20	3.80
	[7]	96.76	97.38	3.84
	[20]	96.06	94.34	2.27
Exploit	**DT-SVM**	**87.47**	**87.72**	**12.83**
	[19]	83.52	85.09	18.40
	[7]	79.19	67.31	6.23
	[20]	84.29	88.27	20.57
Analysis	**DT-SVM**	**97.69**	**99.92**	**27.00**
	[19]	93.59	99.57	82.29
	[7]	-	-	-
	[20]	93.19	99.53	87.14
Backdoor	**DT-SVM**	**94.67**	**98.94**	**59.43**
	[19]	93.59	99.59	82.29
	[7]	-	-	-
	[20]	93.19	99.53	87.14
Normal	**DT-SVM**	**98.29**	**100**	**4.38**
	[19]	98.23	99.90	4.38
	[7]	41.13	6.79	4.98
	[20]	41.13	6.79	4.98
DoS	**DT-SVM**	**89.65**	**90.23**	**10.86**
	[19]	90.10	92.17	11.73
	[7]	83.45	92.89	24.91
	[20]	80.056	92.09	29.46
Worm	**DT-SVM**	**99.33**	**100**	**70**
	[19]	99.05	100	100
	[7]	-	-	-
	[20]	99.21	99.98	80

TABLE IV
DT-SVM COMPARE TO LS-SVM

Class	Feature Selection	Accuracy (%)	TPR (%)	FPR (%)
ShellCode	DT-SVM	99.459	100	9.69
	SVM without feature selection	5.58	0	0
Reconnaissance	DT-SVM	95.37	93.41	2.03
	SVM without feature selection	43.14	0	0
Generic	DT-SVM	96.12	98.96	7.42
	SVM without feature selection	44.37	0	0
Fuzzer	DT-SVM	87.47	87.72	12.83
	SVM without feature selection	50.76	0	0
Exploit	DT-SVM	87.20	86.97	12.50
	SVM without feature selection	44.89	0	0
Analysis	DT-SVM	97.69	99.92	27.00
	SVM without feature selection	8.29	0	0
Backdoor	DT-SVM	93.88	97.99	58.28
	SVM without feature selection	7.30	0	0
Normal	DT-SVM	98.29	1	4.38
	SVM without feature selection	46.17	18.86	11.00
DoS	DT-SVM	89.65	90.23	10.86
	SVM without feature selection	53.03	0	0
Worm	DT-SVM	99.33	100	70
	SVM without feature selection	0.94	0	0

REFERENCES

[1] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials,* vol. 18, no. 2, pp. 1153-1176, 2016.

[2] B. W. Masduki, K. Ramli, F. A. Saputra, and D. Sugiarto, "Study on implementation of machine learning methods combination for improving attacks detection accuracy on Intrusion Detection System (IDS)," in *Quality in Research (QiR), 2015 International Conference on*, 2015, pp. 56-64: IEEE.

[3] R. K. Sharma, H. K. Kalita, and P. Borah, "Analysis of machine learning techniques based intrusion detection systems," in *Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics*, 2016, pp. 485-493: Springer.

[4] Q. Yang, H. Fu, and T. Zhu, "An Optimization Method for Parameters of SVM in Network Intrusion Detection System," in *Distributed Computing in Sensor Systems (DCOSS), 2016 International Conference on*, 2016, pp. 136-142: IEEE.

[5] R. Kaur and M. Bansal, "Multidimensional attacks classification based on genetic algorithm and SVM," in *Next Generation Computing Technologies (NGCT), 2016 2nd International Conference on*, 2016, pp. 561-565: IEEE.

[6] A. Nema, B. Tiwari, and V. 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*, 2016, pp. 26-31: ACM.

[7] H. Gharaee and H. Hosseinvand, "A new feature selection IDS based on genetic algorithm and SVM," in *Telecommunications (IST), 2016 8th International Symposium on*, 2016, pp. 139-144: IEEE.

[8] H. Gharaee and M. Fekri, "A New Feature Selection For Intrusion Detection System," *International Journal of Academic Research,* vol. 7, 2015.

[9] R. R. Reddy, Y. Ramadevi, and K. N. Sunitha, "Effective discriminant function for intrusion detection using SVM," in *Advances in Computing, Communications and Informatics (ICACCI), 2016 International Conference on*, 2016, pp. 1148-1153: IEEE.

[10] P. Nskh, M. N. Varma, and R. R. Naik, "Principle component analysis based intrusion detection system using support vector machine," in *Recent Trends in Electronics, Information & Communication Technology (RTEICT), IEEE International Conference on*, 2016, pp. 1344-1350: IEEE.

[11] P. Wang, K.-M. Chao, H.-C. Lin, W.-H. Lin, and C.-C. Lo, "An Efficient Flow Control Approach for SDN-Based Network Threat Detection and Migration Using Support Vector Machine," in *e-Business Engineering (ICEBE), 2016 IEEE 13th International Conference on*, 2016, pp. 56-63: IEEE.

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

[13] T. Mehmood and H. B. M. Rais, "SVM for network anomaly detection using ACO feature subset," in *Mathematical Sciences and Computing Research (iSMSC), International Symposium on*, 2015, pp. 121-126: IEEE.

[14] A. D. Landress, "A hybrid approach to reducing the false positive rate in unsupervised machine learning intrusion detection," in *SoutheastCon, 2016*, 2016, pp. 1-6: IEEE.

[15] L. Rokach and O. Maimon, "Data Mining With Decision Trees: Theory and Applications," 2014.

[16] L. Li Zhong, Z. Ya Ming, and Z. Yu Bin, "Network intrusion detection method by least squares support vector machine classifier," in *Computer Science and Information Technology (ICCSIT)*, Chengdu 2010.

[17] L. L. Zhong, Z. Y. Ming, and Z. Y. Bin, "Network intrusion detection method by least squares support vector machine classifier," in *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, 2010, vol. 2, pp. 295-297: IEEE.

[18] M. Kuhn and K. Johnson, *Applied predictive modeling*. Springer, 2013.

[19] T. Janarthanan and S. Zargari, "Feature selection in UNSW-NB15 and KDDCUP'99 datasets," in *Industrial Electronics (ISIE), 2017 IEEE 26th International Symposium on*, 2017, pp. 1881-1886: IEEE.

[20] N. Moustafa and J. Slay, "The significant features of the UNSW-NB15 and the KDD99 data sets for Network Intrusion Detection Systems," in *Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS), 2015 4th International Workshop on*, 2015, pp. 25-31: IEEE.