Comparison of Intrusion Detection System Hybrid Approach in Computer Networks with Previous Methods

Mehdi Khodamoradi

Department of Computer Engineering, Faculty of Technical and Engineering,
University of Imam Reza International, Mashhad, Iran

Abstract - Various techniques have been used in designing a misuse detection system among which machine learning algorithm, smart expert systems and statistical methods can be pointed out. This study aims to compare the intrusion detection system hybrid approach in computer networks with previous methods in order to improve attack detection and reduce false alarms. The architecture of the proposed method has three stages. In the first stage, pre-processing data and feature selection using different methods such as information gain and Fisher algorithm, selecting samples was done by using various clustering methods such as self-organizing mapping, K-means clustering and data classification. In the second stage, 4 decision trees classifiers i.e. naïve Bayesian, KNN (K-nearest neighbors) and neural networks were used in order to generate median data. At the third stage, an incremental classification based on decision tree was used. Results show that the proposed hybrid method, relative to both previous individual and combined classifications, are more efficient in detecting denial of service, port scanning, remote to local (R2L) and user to root (U2R) attacks.

Keywords: machine learning algorithms, intrusion detection system, three-stage clustering and distance sum-based method;

Introduction

The increasing need for data access, and quick processing and at the same time, increasing the data size, as well as the need to provide data from different resources using computer networks, has led to the emergence of sources of threat which can exploit and cause malfunctions in systems(1). Penetrationinto a network is usually considered an attack. It could be said that the first attempts to create intrusion detection systems date back to early 80s and before (2).

When the idea to use an intrusion detection system to prevent penetrations was first put forth, it was only well-received by military and important business environments due to their high load average. Today, with the significant progress in designing and producing hardware, application-specific integrated circuit and developments in modern architectures in designing and producing software, utilizing this idea and technology for broad-spectrum of computer systems are made possible.

Among the most important challenges in intrusion detection, selecting an architecture while using a multi-classifier and feature selection can be pointed out (3). Using a classifier was more common in recent years, while currently, using hybrid methods and also utilizing an ensemble of classifiers can be seen in most projects. Evolutionary algorithms play a major role in intrusion detection systems. Tests have demonstrated the level of precision and effectiveness. At the same time, challenges in this regard also exist and solving them can improve the performance of these algorithms in the field of intrusion detection. The unpredictability of termination criterion, high time complexity and also the unbalanced data distribution in the data set are some of the challenges in this field (3). In order to achieve a higher precision rate, hybrid methods are drawing more attention. The original idea is based on combining multiple machine learning methods, in order to upgrade the performance of the system. Feature selection and data dimensionality reduction is also one of the primary challenges, not only in intrusion detection systems, but also in all fields of data-mining. Considering the high similarity of some attacks to normal activities in the network, selecting proper features to distinguish them more precisely during recognition is another challenge in this field. Better feature selection will cause more precision and reduce the recognition time. Generally, classification methods are as follows: 1. Single classification 2. Ensemble classification (4) 3. Hybrid classification. In single classification, only basic classification is used.

In ensemble classification, multiple classifications are used and after each classifier has commented on input data label, these votes are combined and the final label is determined using specific methods.

Several studies have been carried out in this field (5).A strategy based on distributed factors in combination with PCC has been used. In this strategy, the attack detection system was divided into two separate layers of host layer and classification layer. In [6], the proposed method of penetration detection system includes four components. In [7], a set of one-class classifiers is presented which use different trainers and in [8], 6 features in the KDD-CUP99 out of the 41 data sets available is selected. Results show that this combination could recognize 97.25% of the instances correctly. In [9], a hybrid method is used to detect abnormalities in wireless sensor networks. In [10], the results are improved by combining the two methods of K-means clustering and decision tree using C4.5 algorithms. In [11], Jiang presents a new method which merges the abnormality detection and misuse in a hierarchical radial basis function network (HRBFN). In [12], a support-vector machine, simulated annealing and decision tree is used to detect attacks. Method of presenting a hybrid approach for intrusion detection systems was based on distance summation in 2014 (13). In 2012, a study titled "An Implementation of Intrusion Detection System Using Genetic Algorithm" by Hoque, Mukit and Bikas was published (14). In hybrid methods, classifiers are combined in a way that input data are pre-processed and classifier uses data generated in the previous level. Hybrid methods are practically more efficient relative to the two previous methods and most studies carried out focus on these methods in recent years. This study proposes a three-stage method and strives to improve attack detection and reduce false alarms. Following the research, in the second chapter, the proposed method has been explained; in the third chapter results are evaluated and chapter four concludes the study.

2. Proposed Method

In the proposed method, initially we need to assign classifiers to each class which can distinguish and detect better relative to other classifiers. In other words, each ensemble of classes needs to perform better in the detection of a specific class. Next, in order to improve class detection, we will try to achieve the detection pattern of classifiers in the first stage. The output of each classifier in the second stage is a binary amount which specifies if an instance belongs to a class or not. Different classes can be used in the second stage and there's no requirement that the same classifiers should be used for all classes. In order to create median results in the second stage out of the four decision tree classifiers, KNN (K-nearest neighbors), naïve Bayesian and neural network for each class label has been used. In last stage, the generated median data is used to teach the final classifier. In this research, incremental method based on decision tree 1 has been used as the final classifier. The advantage is the fact that here the classifiers' detection pattern is summed and trained. This method can improve the detection rate and hybrid approach precision.

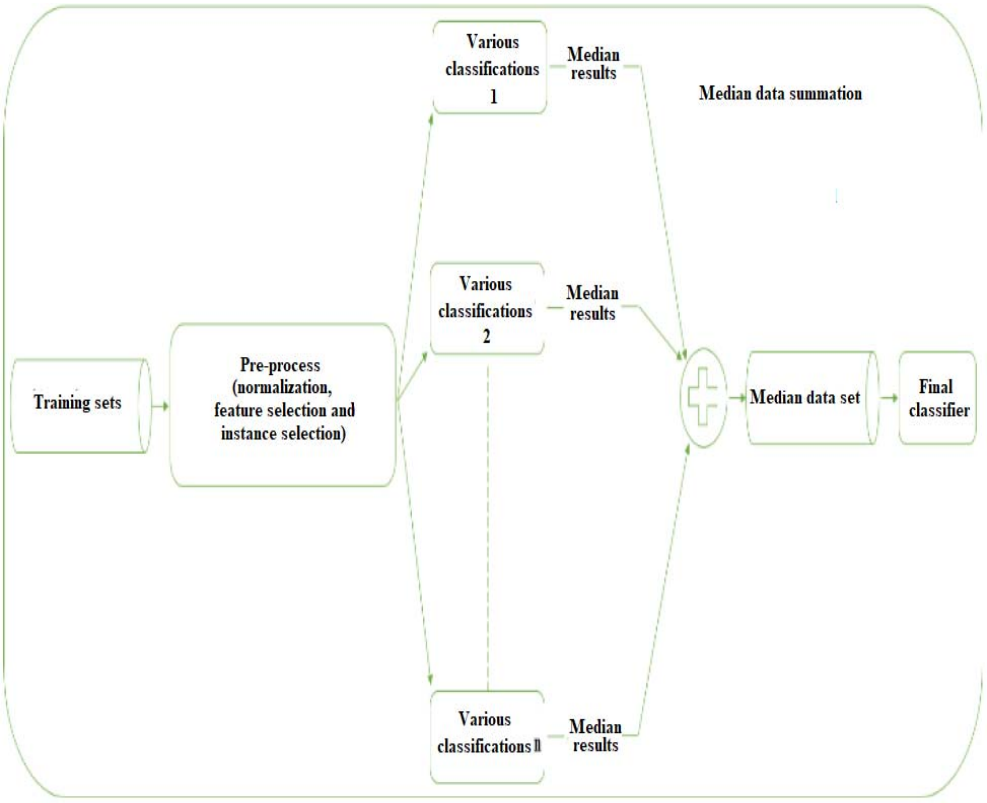


Figure 1. General architecture of the proposed method.

3. Evaluation Results

In order to examine the performance of the proposed hybrid approach, the KDD-CUP99 data set was used. Evaluation criteria used in order to compare the efficiency of the proposed method with others include precision, recall, precision, false alarm rate, error rate and F-value.

3.1 Examination of topological similarity of train and test instances

In order to examine the topologic similarity of train and training dataset instances, first a network of 11x11 was created and using KDD-CUP99 training data set instances, models of instances were achieved in the form of 2-dimensional map. Then, in order to examine the similarity, test instances on created models were applied and results were drawn in the form of the figures below. What follows is the figures of the 2-dimensional map related to each class in the form of train and test instances along with explanations.

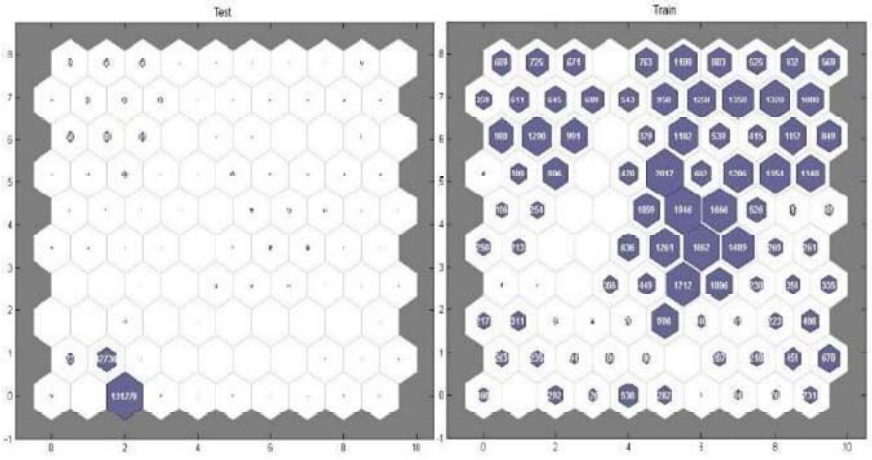


Figure 2. Results drawn from a self-organizing network over DoS attacks - left side shows train, right side shows test.

As you can see in Figure 2, DoS training data are distributed fairly. Also in the training dataset, a major area of the instances is focused on the two cells of the first line of the third column and the second line of the second column. We can simply conclude that a huge percentage of the instances of major training datasets should be trained and recognized with very few of the train data.

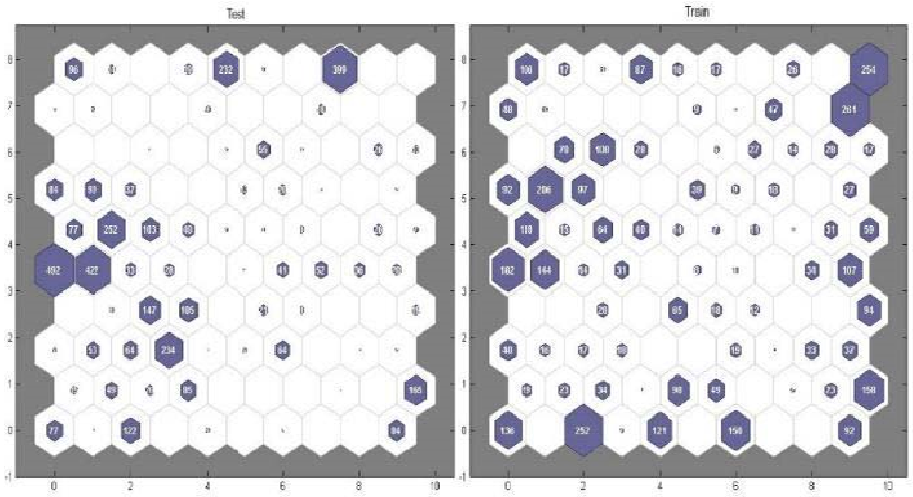


Figure 3. Results drawn from a self-organizing network over port scan attack data - left side shows test, right side shows train.

According to Figure 3, in port scanning attack training dataset, we can say that the distribution of instances is suitable in total space. Also, in test set, there are no training samples regarding some specific areas of the data.

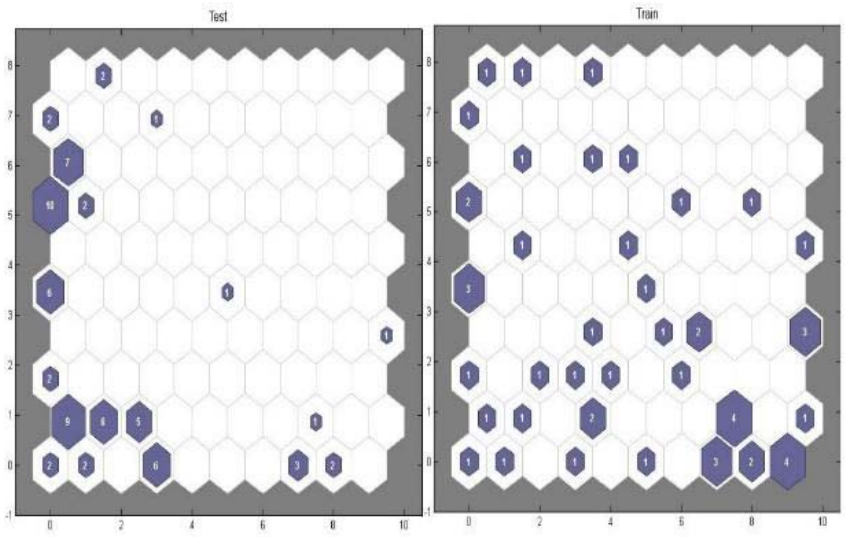


Figure 4. Results drawn from a self-organizing network over U2R attack - right side shows training, left side shows test.

According to Figure 4, in U2R data, unbalanced distribution can be observed between the training and the test side. Furthermore, in some areas of the test data, there are unique unseen cases in training sets.

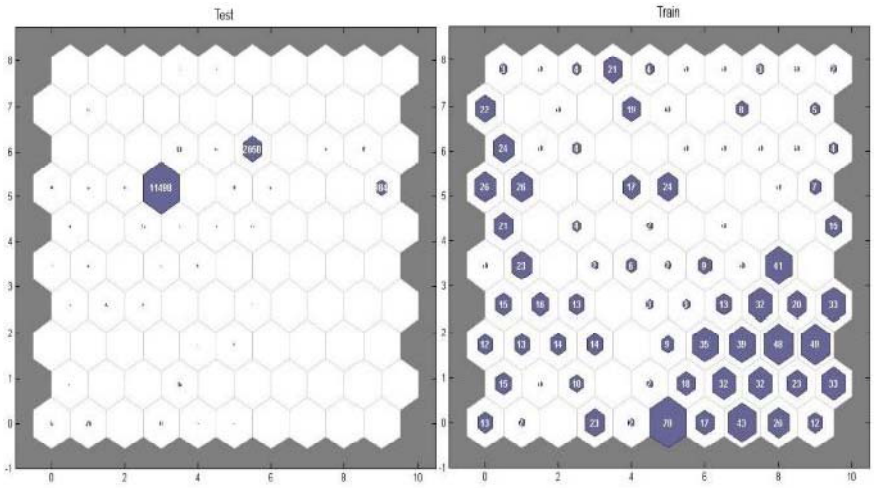


Figure 5. Results drawn from a self-organizing network over R2L attack data – right side shows training, left side shows test.

According to Figure 5, R2L attack data include distribution in training area, while test data are almost concentrated in one area. As you can see, the number of available training samples in mentioned cell is very small and it is highly likely that the detection precision in data related to this attack is very low. Since R2L attack is very similar to normal network activity and also the low number of training samples, especially in denser areas of train instances, make the detection of these types of attacks a major challenge in intrusion detection systems.

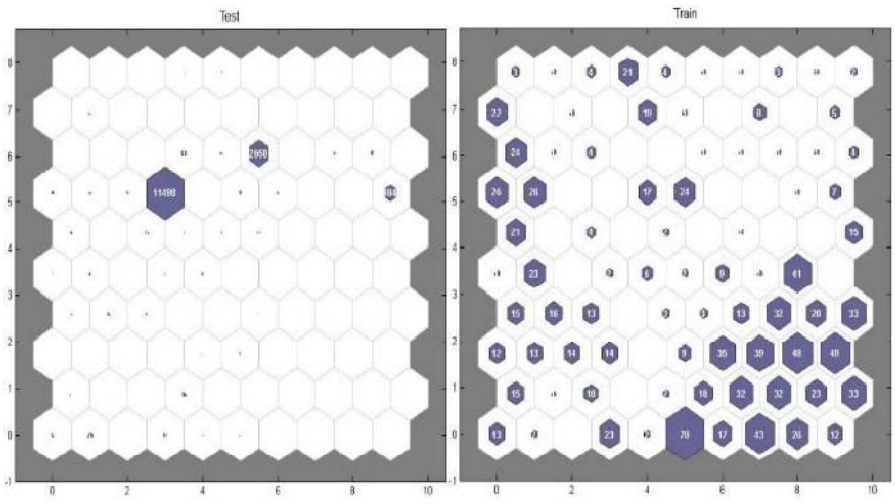


Figure 6. Results drawn from a self-organizing network over normal activity data. Right side shows train, left side shows test.

Based on Figure 6, we can see that the distribution of training and test instances of normal activity in data set is balanced. Also, the number of training set instances in each area is evaluated to be proper distribution-wise.

3.2 Efficiency Evaluation

Evaluating the efficiency of the proposed method through efficiency evaluation and the effect of sample selection is done by means of clustering and each of the subsystems, utilizing other classifiers as final classifier and comparing with previous methods.

1. Efficiency evaluation and the effect of sample selection by means of clustering

In Table 1, a comparison of random and interval selection based on distance till class average, clustering by K-means and clustering based on self-organizing network for each class including DoS, port scan, U2L, R2L and normal activity is presented separately. All figures are based on binary decision tree classifier training.

Table 1. Evaluation of different methods of sample selection

SOM	*K-Means*	*D-AVG*	*Random*		
0.995092	0.994565	0.995061	0.993975	*Precision*	*Normal*
0.909174	0.911862	0.914609	0.915715	*Recall*	
0.950195	0.951419	0.95314	0.953242	*F-Value*	
0.998799	0.967823	0.998876	0.998776	*Precision*	*DOS*
0.940427	0.998806	0.973914	0.940693	*Recall*	
0.968734	0.938704	0.986237	0.968864	*F-Value*	
0.88776	0.849277	0.890947	0.776837	*Precision*	*Prob*
0.860058	0.87374	0.819731	0.880701	*Recall*	
0.873689	0.861335	0.853857	0.825515	*F-Value*	
0.589618	0.584151	0.875576	0.328999	*Precision*	*R2L*
0.036826	0.047348	0.034869	0.015477	*Recall*	
0.069323	0.087596	0.067067	0.029563	*F-Value*	
0.525	0.545455	0.478261	0.461538	*Precision*	*U2R*
0.3	0.342857	0.314286	0.257143	*Recall*	
0.381818	0.421053	0.37931	0.330275	*F-Value*	

Comparing the results of Table 1, we can conclude that the best result of sample selection in normal class and DoS is based on intervals distance till the average of class. At this rate, the best result in port scan class is from sample selection based on clustering by self-organizing map method and eventually, the best result is in R2L access class and U2R attack is from sample selection based on K-means clustering. Overall, sample selection based on different methods causes diversity and variety in training data and it is expected that this process will improve the detection in the proposed system.

We will continue to evaluate and examine the effects of sample selection in the final results. The method of evaluation first does the training and classification using randomly selected instances which is the common method in papers and researches and then, in other modes, final results are extracted using diverse sample selection ways in the proposed methods. The results of stated modes are shown in Table 2.

Table 2. Final results of the proposed method

Proposed Approach	*Random*		
0.736468	0.718107	*Precision*	*Normal*
0.994818	0.992474	*Recall*	
0.846367	0.833287	*F-Value*	
0.998887	0.997753	*Precision*	*DOS*
0.972274	0.964142	*Recall*	
0.985401	0.98066	*F-Value*	
0.849577	0.834403	*Precision*	*Prob*
0.820211	0.74868	*Recall*	
0.834636	0.789221	*F-Value*	
0.808023	0.830203	*Precision*	*R2L*
0.069003	0.067597	*Recall*	
0.127149	0.125014	*F-Value*	
0.529412	0.168224	*Precision*	*U2R*
0.257143	0.257143	*Recall*	
0.346154	0.20339	*F-Value*	

2. Efficiency Evaluation of Proposed Method with the Subsystems

At a general glance, we can express the evaluation criteria in the form of two normal activity and attack classes (Table 8-4.).

Table 3. Proposed Method Evaluation Criteria

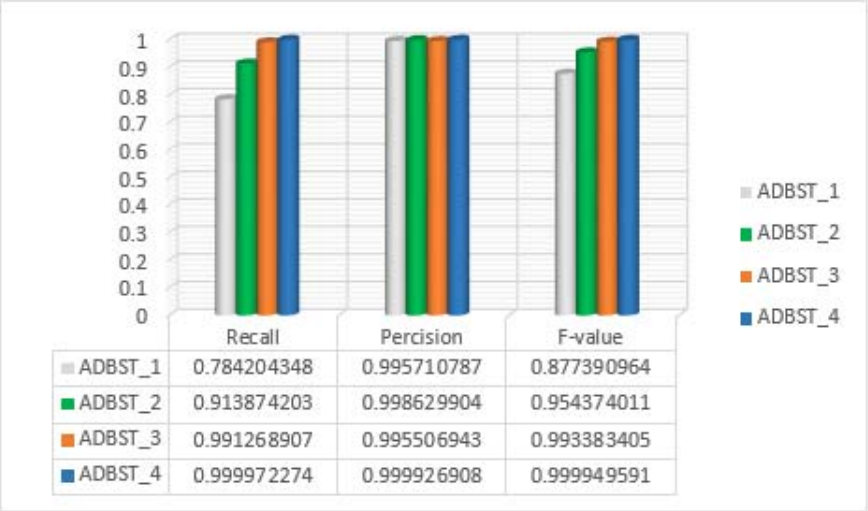
F-Value	*Precision*	*Recall*
0,95655	**0,99863**	**0,917867**

Here, the difference stages of the proposed method are expressed and the equivalent title is shown in Table 4.

Table 4. Different sections of the proposed method along with their equivalent titles.

Description	**Title of stage in the diagram**
Execution of Adaboost alogorithm over main data without creating median results	***ADBST_1***
Execution of Adaboost alogorithm over median classifiers results	***ADBST_2***
Execution of Adaboost alogorithm over main data without creating 10-fold median resuts	***ADBST_3***
Execution of Adaboost alogorithm over 10-fold median classifiers results	***ADBST_4***

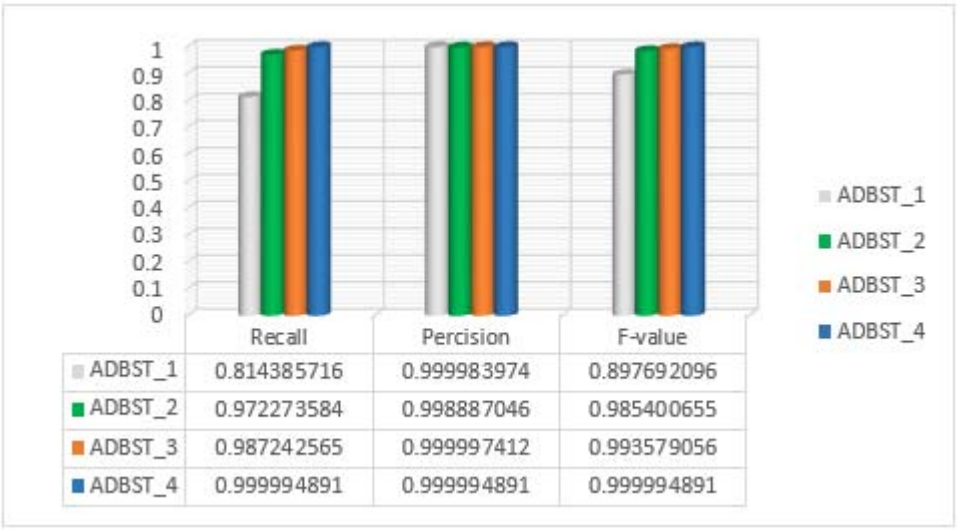
Next, the results of each class including normal, DoS, port scan, R2L and U2R is calculated and shown based on explanations from Table 4.



	Recall	Percision	F-value
ADBST_1	0.784204348	0.995710787	0.877390964
ADBST_2	0.913874203	0.998629904	0.954374011
ADBST_3	0.991268907	0.995506943	0.993383405
ADBST_4	0.999972274	0.999926908	0.999949591

Figure 7. Comparison of results in different sections of the proposed method- normal user class

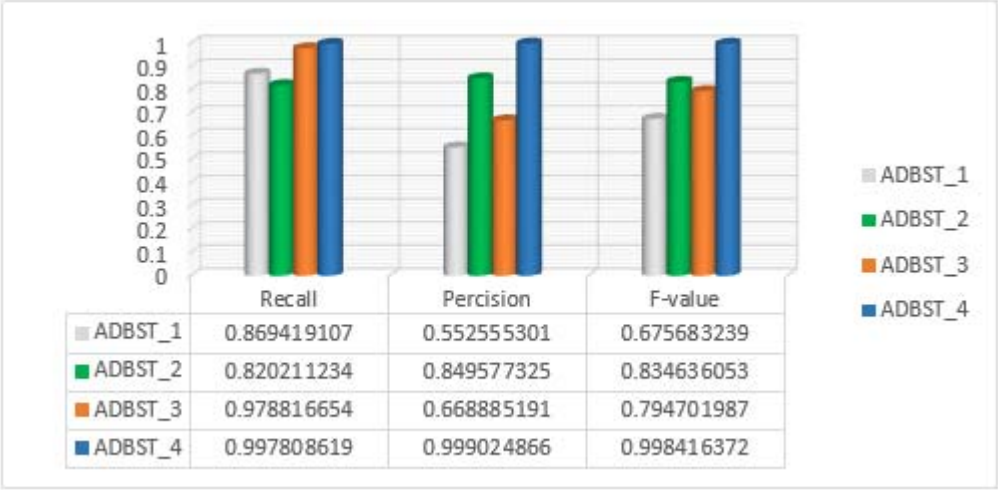
As you can see in Figure 7, execution of the proposed method using the 11-fold method has produced better results.



	Recall	Percision	F-value
ADBST_1	0.814385716	0.999983974	0.897692096
ADBST_2	0.972273584	0.998887046	0.985400655
ADBST_3	0.987242565	0.999997412	0.993579056
ADBST_4	0.999994891	0.999994891	0.999994891

Figure 8. Comparison of results in in difference sections of the proposed method – DoS class

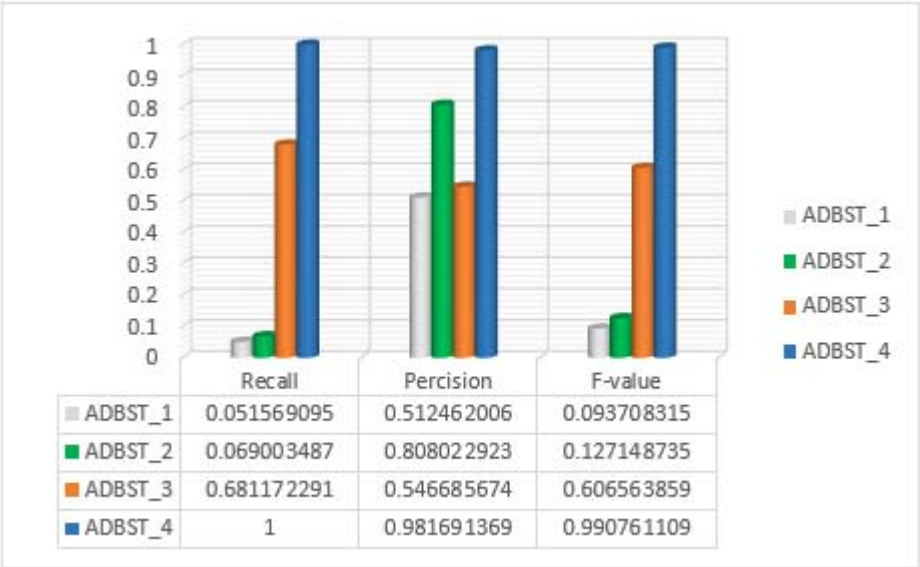
According to Figure 8, we can point out to the fact that using the 11-fold method via median values does not make much difference efficiency-wise. In fact, we can conclude that the train and test data have a similar structure and are not much different topological-wise.



	Recall	Percision	F-value
ADBST_1	0.869419107	0.552555301	0.675683239
ADBST_2	0.820211234	0.849577325	0.834636053
ADBST_3	0.978816654	0.668885191	0.794701987
ADBST_4	0.997808619	0.999024866	0.998416372

Figure 9. Comparison of results in different sections of the proposed method – port scan class

We can conclude from Figure 9 that using median classifier results has fairly increased the detection precision in port scan class based on F-value.This procedure can also be seen in 11-fold method using the median values.



	Recall	Percision	F-value
ADBST_1	0.051569095	0.512462006	0.093708315
ADBST_2	0.069003487	0.808022923	0.127148735
ADBST_3	0.681172291	0.546685674	0.606563859
ADBST_4	1	0.981691369	0.990761109

Figure 10. Comparison of results in different sections of the proposed method – R2L

According to figure 10, using the results of basic classification as median data has improved the R2L attack detection in 11-fold method greatly.

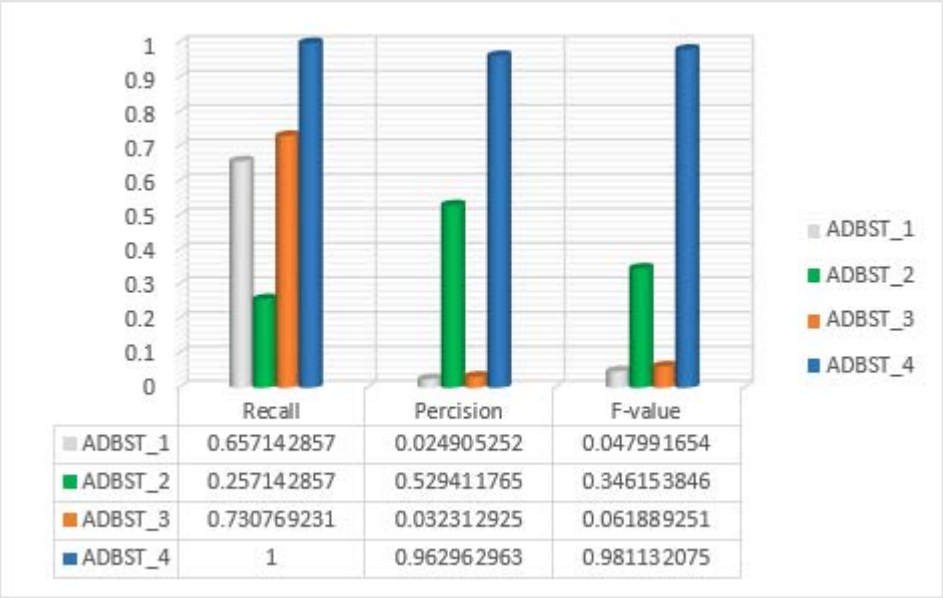


Figure 11. Comparison of results in difference sections of the proposed method – U2R

The number of U2R attack class instances compared to other classes is much lower and the possibility of training learning machines of these instances is very low. This fact can be useful when using median results. Results in Figure 11 confirm this claim.

3. Comparison of other classifiers instead of final classifier of proposed methods

In Table 5,we will examine the results from other classifiers instead of AdaBoost classifier based on decision tree in the proposed algorithm.

Table 5. Examination of other classifiers as final classifier

Tree	*KNN*	*NN*	*Adaboost*		
0.907861	0.908488.	0.908739	0.913874	*Recall*	*Normal*
0.997906	0.997532	0.998464	0.99863	*Percision*	
0.950756	0.95093	0.951491	0.954374	*F-Value*	
0.972613	0.972104	0.972917	0.972274	*Recall*	*DOS*
0.99619	0.997651	0.995672	0.998887	*Percision*	
0.98426	0.984712	0.984163	0.985401	*F-Value*	
0.655545	0.785406	0.710514	0.820211	*Recall*	*Prob*
0.9324	0.912946	0.912454	0.849577	*Percision*	
0.769838	0.844387	0.79892	0.834636	*F-Value*	
0.016211	0.02661	0.004588	0.069003	*Recall*	*R2L*
0.543033	0.960265	0.925926	0.808023	*Percision*	
0.031482	0.051786	0.009131	0.127149	*F-Value*	
0.057143	0.185714	0.042857	0.257143	*Recall*	*U2R*
0.5	0.168831	0.5	0.529412	*Percision*	
0.102564	0.176871	0.078947	0.346154	*F-Value*	

According to Table 5, we can say that the normal and DoS class of all classifiers have almost the same performance and there is no significant difference. In port scan class, the best performance is the AdaBoost classifier based on decision tree and next is KNN (K-nearest neighbors). In the two classes of R2L and U2R the best performance is the AdaBoost based on decision tree.

4. Comparison of the proposed method with 3-stage clustering method and distance summation

The method of presenting a distance sum-based hybrid method for intrusion detection (13) was proposed in 2014. In Table 6, the results of this method with the proposed hybrid approach is compared in the form of evaluation criteria. Also, in Figure 12, the F-value evaluation is drawn in diagram.

Table 6. Comparison of the proposed method results with the 3-stage clustering and distance summation method

F-Value	*Precision*	*Recall*	*Method*	*Class*
0.851241	0.750022	0.984041	DSSVM (Chun, et al., 2014)	*Normal*
0.846367	0.736468	0.994818	Proposed Approach	
0.985328	0.998967	0.972056	DSSVM	*DOS*
0.985401	0.998887	0.972274	Proposed Approach	
0.712958	0.601584	0.87494	DSSVM	*Prob*
0.834636	0.849577	0820211	Proposed Approach	
0.114561	0.61827	0.063129	DSSVM	*R2L*
0.127149	0.808023	0.069003	Proposed Approach	
0.036364	0.044586	0.030702	DSSVM	*U2R*
0.346154	0.529412	0.257143	Proposed Approach	

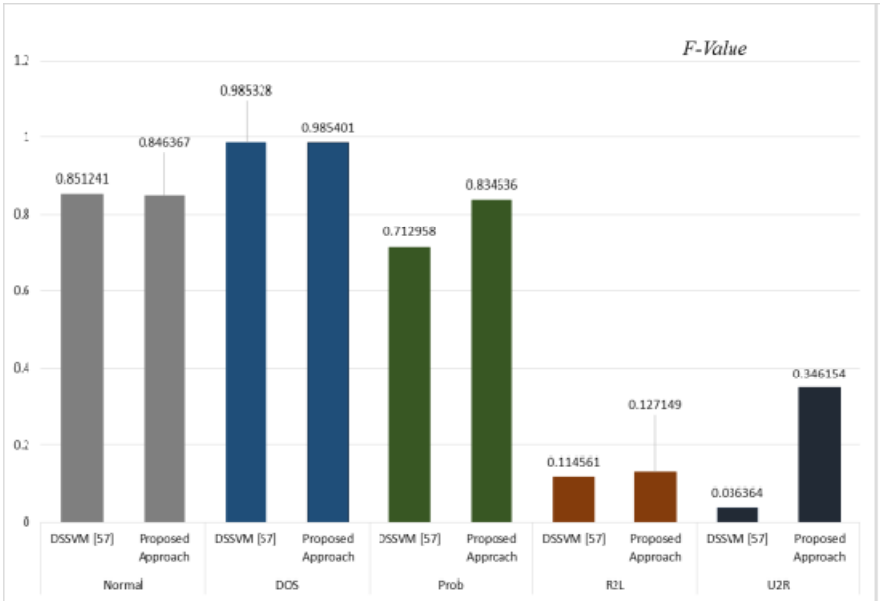


Figure 12. Comparison of F-value evaluation of the proposed method with 3-stage clustering and distance summation method

As we can see in Table 6 and Figure 12, it is clear that the proposed method has improved the detection precision in port scan, R2L and U2R attacks according to F-value. This precision increases in two port scan and U2R attacks are clear.

In F-value of the DoS attack class, there is no significant difference between the proposed hybrid method and the 3-stage clustering and distance summation.

In detecting normal activity, the 3-stage method based on distance summation has performed better than the F-value of proposed method. Of course, it is worth noting that the F-value being a better criterion does not necessarily signify that the F-value is overall the best criterion in this class, since the proposed method has a false alarm value of 1.115 compared to the previous method being 1.1159 in normal activity class. False alarm rate criterion proves that by reducing the F-value by 0.5%, the proposed method has improved the false alarm rate in the intrusion detection system.

Conclusion

In this study, the results drawn from evaluation of the proposed method of intrusion detection was presented in 4 sections. First, the results drawn from the effect of sample selection via different methods were shown. The results displayed that the use of sample selection using various methods will lead to improved results. Next the results and median classification effects were reviewed. According to the evaluation diagram presented, the use of median classifiers' results has significant effect on the detection precision rate.

Next, results drawn from utilizing various other machine learning methods such as decision trees, KNN and neural networks as final classifiers were evaluated. It was clarified that the final classifier utilized as AdaBoost method based on decision tree has performed better than other methods in comparison. Then, the proposed method was compared to the 3-stage clustering and distance summation method.

Table 7. Comparison of F-value evaluation of the proposed method and the three-stage clustering and distance summation based on 5 classes.

Proposed Approach	DSSVM(Chun, et al., 2014)	Criteria	Class
0,846367	0,851241	*F-Value*	*Normal*
0,985401	0,985328	*F-Value*	*DOS*
0,834636	0,712958	*F-Value*	*Prob*
0,127149	0,114561	*F-Value*	*R2L*
0,346154	0,036364	*F-Value*	*U2R*

In Table 7, comparisons of F-measure evaluation of the proposed method with the three-stage clustering and distance summation method has been reviewed and displayed. Also, in the evaluation of the proposed hybrid method in binary classification, results show that the proposed method was able to reach a detection precision of 1.95655 in while being able to reduce the false alarm by 1.115 F-value evaluation which is the geometric mean recall and precision criteria.

Table 8. Comparison of the proposed method with three previous methods based on normal and attack classifiers.

Error Rate	*F-Value*	*False Alarm*	Method
0,067	0,9567	0,0159	DSSVM(Chun, et al., 2014)
0,067	0,9565	0,005	Proposed method

In Table 8, you can see a review of evaluation results of the proposed method's efficiency compared to the three-stage clustering and distance summation method in binary classification.

References:

[1] Kumar, & Sandeep. (1995). Classification and detection of computer intrusions. P.H.D. Thesis, Purdue University.
[2] Innella, P. (2111). The Evolution of Intrusion Detection systems. Tetrad Digital Integrity, LLC.
[3] Catania, C., & Garino, C. (2112). Automatic network intrusion detection: Current techniques and open issues. Computers and Electrical Engineering archive. Volume 38 Issue 5, September, 2012. Pages 1062-1072. Pergamon Press, Inc.
[4] Tsai, C., Hsu, Y., Lin, C., & Lin, W. (2119). Intrusion detection by machine learning: A review. Expert Systems with Applications, 11994-12111.
[5] Xie, Z., Quirino, T., & Shyu, M. (2116). A Distributed Agent-Based Approach to Intrusion Detection Using the Lightweight PCC Anomaly Detection Classifier. SUTC.
[6] Xiang, C., Yong, P., & Meng, L. (2118). Design of multiple-level hybrid classifier for intrusion detection system using Bayesian clustering and decision trees. Cheng Xiang, Png Chin Yong, Lim Swee Meng, “Design of Multiple-Pattern Recognition Letters, 918-924.
[7] Zainal, Anazida, Maarof, Aizaini, M., Shamsuddin, & Mariyam, S. (2119). Ensemble Classifier for Network Intrusion Detection System. Journal of Information Assurance and Security, 217-225.
[8] Agarwal, B., & Mittal, N. (2112). Hybrid Approach for Detection of Anomaly Network Traffic using Data Mining Techniques. Procedia Technology, 996-1113.
[9] Curiac, D., & Volosencu, C. (2112). Ensemble based sensing anomaly detection in wireless sensor networks. Expert Systems with Applications, 9187-9196
[10] Muniyandi, A., Rajeswari, R., & Rajar, R. (2112). Network Anomaly Detection by Cascading K-Means Clustering and C4.5 Decision Tree algorithm. Procedia Engineering, 174-182.
[11] Jiang, J., Zhang, C., & Kamel, M. (21-24). RBF-based real-time hierarchical intrusion detection systems. Proceedings of the International Joint Conference on Neural Networks, 2113.
[12] Lin, S., Ying, K., & Lee, C. (2112). An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection. Applied Soft Computing, 3285-3291.
[13] Chun, G., Yajian, Z., Ping, Y., Zhang, Z., Lio, G., & Yang, Y. (2114). A distance sum-based hybrid method for intrusion detection. Springer US Applied Intelligence Journal, 178-188.
[14] Hoque, M., Mukit, M., & Bikas, M. (2112). AN IMPLEMENTATION OF INTRUSION DETECTION SYSTEM USING GENETIC ALGORITHM. International Journal of Network Security & Its Applications (IJNSA).