Journal of Advances in Computer Research
Quarterly ISSN: 2008-6148
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 11, No. 4, November 2020), Pages: 39-55
www.jacr.iausari.ac.ir

Improving Accuracy in Intrusion Detection Systems Using Classifier Ensemble and Clustering

Ensieh Nejati * , Hassan Shakeri, Hassan Raei
Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran
nejati.ensi@gmail.com; shakeri@mshdiau.ac.ir; h.raei@mshdiau.ac.ir

Received: 2020/08/06; Accepted: 2020/10/10

Abstract

Recently by developing the technology, the number of network-based services is increasing, and sensitive information of users is shared through the Internet. Accordingly, large-scale malicious attacks on computer networks could cause severe disruption to network services so cybersecurity turns to a major concern for networks. An intrusion detection system (IDS) could be considered as an appropriate solution to address the cybersecurity. Despite the applying different machine learning methods by researchers, low accuracy and high False Alarm Rate are still critical issues for IDS. In this paper, we propose a new approach for improving the accuracy and performance of intrusion detection. The proposed approach utilizes a clustering-based method for sampling the records, as well as an ensembling strategy for final decision on the class of each sample. For reducing the process time, K-means clustering is done on the samples and a fraction of each cluster is chosen. On the other hand, incorporating three classifiers including Decision Tree (DT), K-Nearest-Neighbor (KNN) and Deep Learning in the ensembling process results to an improved level of precision and confidence. The model is tested by different kinds of feature selection methods. The introduced framework was evaluated on NSL-KDD dataset. The experimental results yielded an improvement in accuracy in comparison with other models.

Keywords: Intrusion Detection System, Ensemble Classifier, Clustering, Decision Tree, K-Nearest-Neighbor, Deep Learning

1. Introduction

Nowadays, by increasing the connectivity among computers, the users will be able to communicate with each other through the internet without time and distance limitation. So, the rapid growth of the network leads to emerging a phenomenon which is called cyber threats [1]. The widespread using of the Internet brings a great opportunity for hackers or legitimate users of the network to do malicious activities including stealing and gathering sensitive and vital data of users. These activities could be done by breaking into computer system and accessing to the network remotely [2-3]. It is obvious that if the security problems do not be addressed properly, the critical information might be leaked. As a result, the security of networks cannot be ignored, and detecting and countering cyberattacks is the primary challenge faced by many researchers [4]. There exist different kinds of cyberattacks, including wormhole, Sinkhole, Sybil, Selective Forwarding [5] and botnet attacks. Each of the mentioned threat could have a different destructive effect on the efficiency of networks. It is noteworthy that botnet attack is known as the most destructive threat compared with other malware because of its special structure [6]. Some traditional methods like user authentication, firewall and data encryption are applied by organizations for maintaining networks in a secured condition. Hence, for making systems more reliable to cope with malware attacks, it is essential to devise an efficient solution [7-8]. Intrusion detection system (IDS) could be one of the most effective and impressive approaches to discovering the cyber-attacks and enhancing the security of networks [9-10]. The concept of IDS means the process of monitoring and analyzing the behavior of the network for the purpose of detecting any suspicious activities and raising an alert as soon as any intrusion is discovered. An IDS could be a device or application program and it would be category into two major types: signature based (also known as misuse-based) and anomaly-based [11]. Signature-based IDS refers to the detection of known attacks by inspecting the current traffic including payload of received packet and comparing them with the specific patterns which are known as signatures [10]. Although signature-based IDS could be accurate in identifying known threats, it is inadequate to detect new attacks as there is no defined template for them [12]. Among different kinds of signature-based IDS, Snort is one of the most used and well-known open-source signature-based IDS [13]. An anomaly-based technique is another division of IDS which detects any action that significantly deviates from the normal behavior. A remarkable advantage of this method is to recognize more types of novel attacks which is completely unlike the signature-based IDS. On the other hand, the primary drawback of these systems is the high false alarm rate (FAR) [10-11]. On some occasions, for overwhelming the disadvantages of these methods, the combination of the signature-base and anomaly-based is employed [3]. For developing an IDS, machine learning (ML) methods have been widely used by researchers recently [14]. Machine learning concentrates on classification and prediction. As the primary and fundamental purpose of IDS is gaining a high accuracy and detection rate and a low false alarm rate (FAR), in the case that one classifier is used to generate an IDS, the results might not be sufficiently beneficial. Thus, in several papers, the authors applied more than one classifier instead of one, in their research.

The main contributions of this paper are summarized as follows:

- We introduce a new approach to sampling the records based on clustering technique for improving the performance of IDS in term of accuracy and false alarm rate (FAR) by feature selection.
- The proposed approach utilizes ensembling of three classifiers i.e. K-nearest neighbor (KNN), decision tree(DT) and deep learning for improving detection accuracy.

To show the efficiency of proposed approach, it is applied on the NLD-KDD dataset.

The rest of this paper is organized as follows. Section 2 reviews the previous related works in this area. Section 3 introduces our methodology in details. The experimental results and discussion of the implication of the findings about the proposed method in this study are introduced in section 4. Finally, section 5 concludes the paper.

2. Related Work

Up to now, many research works have been carried out on the field of designing intrusion detection systems. In this section, we will provide a brief review of some works in this context.

A cluster-based ensemble classifier for IDS was proposed by Jabbar et al. [2]. Authors have used a combination of KNN and ADTree classifiers for detecting intrusion. The Gure dataset has been used for K-means clustering in this paper [2]. The experimental results prove that using ensemble classifier Figures should be numbered results in high accuracy and low value for false alarm rate. In another work, bagging and boosting ensemble methods and tree based algorithms were used in order to construct an intrusion detection system. The authors have applied NSL-KDD dataset and selected 35 features for evaluating. Based on the results obtained using the proposed model, they observed an improvement in accuracy and FAR [15]. A new feature selection for IDS based on Genetic Algorithm and SVM was introduced by Gharaee et al [16]. The authors used Genetic algorithm for reducing the features, and the combination of this algorithm by SVM is applied to discover the anomalies in this study. To evaluate the performance of the proposed model KDD CUP 99 and UNSW-NB15 datasets are used. Thanigaivelan et al. [17] presented a distributed internal anomaly detection system for Internet of Things (IOT). The principle of IDS in this study is that each node has duty to monitor its neighbors and detect any anomaly in the network and reports it to the parent node. The parent node sends a message to the border router in turn. Based on different alarms which have received, edge-router inspecting them and then makes a final decision about intrusion and in necessary time sends a notification to users. Another model based on ensemble classifier is proposed in order to enhance performance of intrusion detection [18]. PSO, KNN and SVM were selected as the contributed classifier in this paper. By taking into account the weight majority voting (WMV) approach, the gained result of classifiers is compared with each other. The designed model was validated on five subsets of features which were chosen from KDD99 dataset. In a study conducted by Hamed et al [19], an NIDS based on recursive feature addition (RFA) and bigram technique was reported. Bigram technique was proposed for encoding the payload features. By this method, the useful features were extracted. Another innovation of this article was a combined method for evaluating, which made a compound of three metrics that is accuracy, detection rate, and false alarm rate. The designed model has been tested on the ISCX 2012 data set [20]. In other work, Gaikwad and Thool [21] performed experiments using the NSL-KDD data set to model an intrusion detection system. Bagging scheme was used for implementing the proposed method. Because of the simplicity of the partial decision tree classifiers it was employed as the base classifier. For reducing the dimensionality of the input feature space from 41 to 15, the genetic algorithm optimization was used. The observation results achieved to a high accuracy on cross validation. A filter-based feature selection algorithm was presented in [22] to design Least Square Support Vector Machine based IDS (LSSVM-IDS). The proposed model was tested on three different datasets including KDD Cup 99 [23], NSL-KDD [24] and Kyoto 2006+ [25]. The evaluation results of proposed model achieve higher accuracy in comparison with other advanced methods. The interest in the use of ensemble method for designing IDS was suggested by some other authors [26-27-28]. Although there have has been plenty of research in this field, there is still a lot more needed in the area. However, to the best of our knowledge, little work has been performed on algorithms to reduce process time of intrusion detection by sampling the data records. Also no previous work has incorporated the classifiers suggested in our work in an ensembling framework for IDS. Therefore, motivated from the literature, we propose a new approach to intrusion detection based on clustering and ensembling three classifiers which are different from the ones applied in other works.

3. Methodology

3.1 System Framework

In this section, the proposed method will be introduced in details. Our methodology is comprised of four main steps that can be defined as follows. For testing the proposed model NSL-KDD dataset is selected in the first step. In the second phase, feature selection techniques are done on the dataset, in order to eliminate irrelevant and redundant features. For reducing the process time, the samples in dataset will be clustered by applying K-means method in the third step. The most outstanding part of our approach is step four. Combination of three classifiers including Decision Tree (DT), K-nearest Neighbor (KNN) and Deep Learning is performed in this step. Eventually, the final result is gained by voting the individual results. For assessing the obtained results from prior step, evaluation measures such as accuracy, recall, precision and false alarm rate (FAR) will be calculated. The overall model of our approach is depicted in Figure 1.

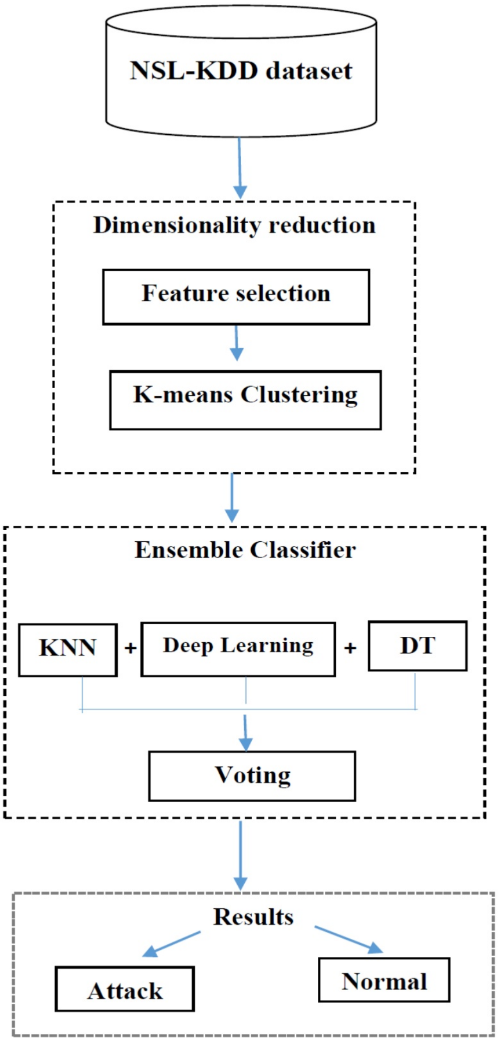


Figure 1. Flowchart of proposed method

3.2 Feature Selection

Each sample in a dataset has several features despite the fact that not all of them are necessary for taking into account by an IDS. The term of feature selection refers to the process of identifying and selecting the most efficient and relevant features from a feature set. Feature selection could be done for different reasons such as improving the accuracy of the classifier and increasing the speed of learning and classifying. This action is also helpful for decreasing the computational load during classification. Totally by feature selection, a more interpretable model will be created [29]. As there exist 41 features in NSL-KDD dataset and some of them have no or minimum role and even lead to extra overhead in detection time of our proposed model, we have applied different feature selection methods as a pre-processing step. Each of these methods makes a subset of features which are Wrapper [30], CFS, FVBRM [31] and Gain Ratio [15]. Table 1 presents the list of features selected by each method.

Table 1. List of selected features with different feature selection methods

Feature selection method	List of selected features	#Feature
FVBRM	1,3,4,5,6,7,8,9,10,11,12,13,14,15, 16,17,18,19,23,24,32,33,36,38,40	25
CFS	3,4,5,6,12,26,29,30,37,38	10
Wrapper	2,3,4,5,6,8,10,12,24,25,29,35,36, 37,39,40	16
Gain Ratio	1,2,3,4,5,6,8,9,10,11,12,13,14,16, 17,18,19,22,23,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41	35

3.3 Clustering

Datasets which are used for designing an IDS are primarily large. By applying clustering techniques, we could reduce the size of dataset. Clustering algorithms group samples together which are similar to each other (each group called a cluster). As no Label Attribute is necessary, clustering can be used on unlabeled data and is an algorithm of unsupervised machine learning. There are various types of clustering methods including Hierarchical, Partitioning, Grid-Based Methods, Constraint-Based Clustering [32], and Graph-based Clustering [33]. Working with a large dataset could be time-consuming. So, for reducing the number of samples in the dataset, we have applied K-means clustering in our approach because of its simplicity and popularity. The k-means determines a set of k clusters and assigns each sample to exact one cluster. The clusters consist of similar samples and the similarity between samples is based on a distance measure between them. A cluster in the k-means algorithm is determined by the position of the center in the n-dimensional space of the n Attributes of the Sample Set. This position is called centroid [34].

3.4 Normalization

The value of features in a dataset are in different types. Some of them are text and some others are numeric. The value of the features which are numeric could be on different scale. A large difference in the value of features (For instance the duration value belonging to interval [0.58329]) might lead to imbalance in the result of clustering. For solving this issue, normalization process is applied before starting clustering. Normalization is used to scale values so they fit into a specific and well-proportioned range. Using this method, the value range of all features will be rescaled and adjusted into the range [0, 1] [22].

3.5 classification

One of the primary data mining problems which comes under the machine learning technique and considered as an instance of supervised learning is classification. It is used to predict the class of membership for data samples which do not contain a class label. In the case of applying large training datasets, using the classification technique could lead to an improvement in the accuracy of models [35]. A lot of classification algorithms are available. Next subsections make a brief description of the used classification algorithms in our model.

3.5.1 Decision tree classifier

Decision tree is a kind of supervised and predictive Machine Learning model which is used for both classification as well as regression purposes. It used a tree-like graph that is a collection of nodes. These nodes contain two types namely decision nodes and leaf nodes. Decision nodes are the ones where the data is split into branches. On the other hand, the leaf nodes are the ends of the branches those do not be split anymore and called the final outcomes. Each node represents an object and splitting rule for one specific attribute. For classifying the instances, decision tree sorts them down from the root to the leaf nodes and this process is repeated until a leaf is encountered. The most common algorithms of this kind are C4.5, CART, ID3 [1-36].

3.5.2 K-Nearest Neighbor classifier

K nearest neighbor which is often abbreviated as KNN is another classification method. It is one of the simplest machine learning methods and an example of lazy algorithm in the sense that it does not generate a model of the data set beforehand. The only calculations it makes are when it is asked to poll the data point's neighbors. It belongs to the supervised learning domain. This classifier tries to appoint which group a data point is in by looking at the data points around it. It means this algorithm works based on comparing an unknown sample with k sample which are the nearest neighbors of it. Applying the classifier includes two steps. In the first step, the KNN algorithm is to find the k closest training samples. "Closeness" is defined as a distance in the n-dimensional space, referred to the n attributes in the training sample set. In the second step, the unknown sample will be classified by a majority vote of the found neighbors. In this classifier, K depicts the number of neighbors and commonly odd numbers chose as the value of K, i.e. K=1, K=3, etc. Different values of K might have various performances.

For example, a high value of K will spend more time and has an effect on the accuracy of prediction. Figure 2 displays K nearest neighbor classification [2-37- 38].

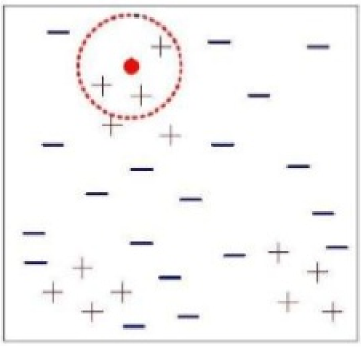


Figure 2. K nearest neighbor classification [2]

3.5.3 Deep learning

Deep learning is a subset of machine learning that is based on multi-layered artificial neural networks. The network can contain a large number of hidden layers which are organized in a cascade mode. The "deep" in "deep learning" refers to the number of layers through which the data is transformed. In this classifier, the input of each successive layer is the output of the previous layer [39-40]. Different layers (input, output, and hidden) are presented in Figure 3.

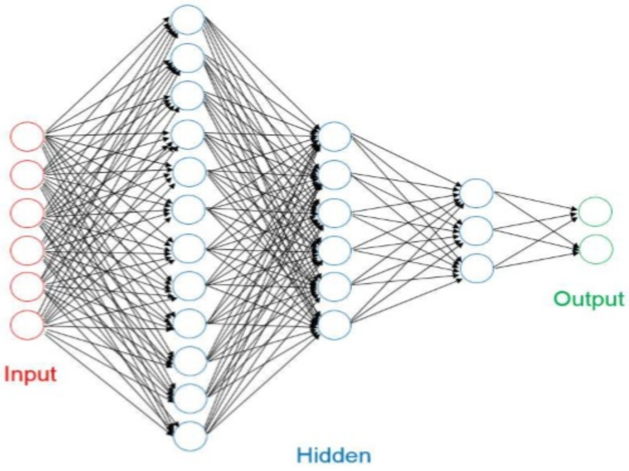


Figure 3. Deep Learning network model [36]

3.5.4 Ensemble classifier

Ensemble classifier refers to classifiers with different approach which work cooperatively and trained on a dataset in order to reduce some weaknesses of individual classifiers and obtain a higher accuracy. Ensemble classifier applied two strategies for combining classifiers, one of which is voting that is arguably the most popular one and another one is taking weight [41-42].

4. Experimental results and discussion

Simulation tool used in this research was RapidMiner 8.2.000. RadipMiner i is an open-source system for data mining. It is written in the Java programming language and has a user-friendly GUI [43]. According to figure 2, our approach contains four main components, of which third and fourth steps i.e. clustering and ensembling the three classifiers could be the most remarkable parts of the model.

Initially, the proposed model was applied to NSL-KDD dataset using different feature selection methods mentioned in table 1 and evaluated based on different metrics. Subsection 4.1 raised the issue of NSL-KDD and in subsection 4.2, the evaluation metrics are introduced.

4.1 NSL-KDD dataset

In the recent years, KDDcup99 was a common dataset among researchers for assessing their designed IDS. However, unfortunately this dataset had some drawbacks. These disadvantages have an adverse impact on the performance of the proposed model. To deal with these problems, some refinements have been done on KDDcup99 dataset which are mentioned below:

Redundant records are removed in train set which provides an opportunity for classifier to have an unbiased result. In test dataset the duplicated records have been discarded resulting in improved performance comparing the situations where learners are biased due to lack of frequent records. The result of applying above operation on the KDDcup99 dataset leads to deriving a new dataset named NSL- KDD. Lately NSL-KDD dataset that is a refined version of the KDDcup99 dataset is used by many researchers to come up with an effective IDS. This dataset is partitioned into two parts namely KDDTrain+, and KDDTest+, which contain 125973 and 22544 records respectively. Each record has 41 features. These features have three types: nominal, binary, and numeric. Aside from this 41 features, any sample has another feature that indicates the class of record and labeled either as attack or normal [44]. Since collecting data is a critical part in designing an IDS, we have also used this dataset in our proposed approach. Table 2 shows the features in this dataset.

Table 2. List of feature in NSL-KDD dataset [24]

Number	Name of feature	Number	Name of feature
1	Duration	22	Is_guest_login
2	Protocol_type	23	Count
3	Service	24	Srv_count
4	Flag	25	Serror_rate
5	Src_bytes	26	Srv_serror_rate
6	Dst_bytes	27	Rerror_rate
7	Land	28	Srv_rerror_rate
8	Wrong_fragment	29	Same_srv_rate

9	Urgent	29	Diff_srv_rate
10	Hot	30	Srv_diff_host_rate
11	Num_failed_logins	31	Dst_host_count
12	Logged_in	32	Dst_host_srv_count
13	Num_compromised	33	Dst_host_same_srv_rate
14	Root_shell	34	Dst_host_diff_srv_rate
15	Su_attempted	35	Dst_host_same_src_port_rate
16	Num_root	36	Dst_host_srv_diff_host_rate
17	Num_file_creations	37	Dst_host_serror_rate
18	Num_shells	38	Dst_host_srv_serror_rate
19	Num_access_files	39	Dst_host_rerror_rate
20	Num_outbound_cmds	40	Dst_host_srv_rerror_rate
21	Is_host_login		

4.2 Evaluation Measures

After completing all the aforementioned steps, the obtained results were evaluated in terms of various metrics. As stated earlier, they include accuracy, recall (detection rate), precision, false alarm rate (FAR) and F-measure. For calculating these performance measures, the confusion matrix is applied [45]. A confusion matrix also called error matrix which is illustrated in table 3, is a table with two rows and two columns and the cells contain four main parameters that are described below:

Table 3. Confusion matrix

		Predicted	
		Normal	**Attacks**
Actual	**Normal**	TN	FP
	Attacks	FN	TP

True Positive (TP): Number of instances where the attack has been detected correctly. True Negative (TN): Number of instances where the normal behavior has been detected correctly. False Positive (FP): Number of instances where detected as attack, while it is actually a normal behavior. False Negative (FN): Number of instances where detected as normal behavior, while it is actually an attack.

Based on the above four quantities, the evaluation measures are defined as follows and calculated using Eqs. (1–5).

Accuracy: shows the ratio of the total number of predictions that were correct. It is determined using the Eq. (1).

$$AC = \frac{TP + TN}{TP + TN + FP + FN} \qquad (1)$$

Recall: shows the ratio of the total number of correctly classified positive samples divide to the total number of positive samples. High Recall (also called detection rate) indicates the class is correctly recognized (small number of FN), as calculated using the Eq. (2).

$$\mathrm{DR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{2}$$

Precision: shows the proportion of predicted positive values which are actually positive. High precision indicates a sample labeled as positive is indeed positive (small number of FP), as calculated using the Eq. (3).

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{3}$$

False alarm rate (FAR): shows the measure of misidentification of normal nodes as attackers. It is calculated using the Eq. (4).

$$\mathrm{FAR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{4}$$

F-measure: Since we have two measures (Precision and Recall) which are in conflict with each other, it is necessary to make a tradeoff between them and have a measurement that represents both of them. It is given in the Eq. (5).

$$\mathrm{F - Measure} = 2 * \frac{\mathrm{precision} * \mathrm{Recall}}{\mathrm{precision} + \mathrm{Recall}} \tag{5}$$

4.3 Test results

By applying the above metrics, the results in table 4 were gained.

Table 4. Experimental results of applying ensembling three classifiers

Feature selection method	accuracy	recall	precision	f-measure	FAR
Gain Ratio	**99.80**	99.68	99.93	99.79	**0.00063**
CFS	99.77	99.65	99.85	99.79	0.0012
Wrapper	99.73	99.64	99.86	99.76	0.0011
FVBRM	**99.83**	99.73	99.88	99.76	0.0010

Table 4 is quite revealing in several ways. From the results, it is apparent that when the proposed approach was implemented by different feature selection methods, if accuracy is going to be considered as the only evaluation measure, FVBRM with 25 features has the highest value of accuracy (99.83) than other methods. On the other hand, in terms of accuracy and FAR, Gain Ratio with 35 features could be an appropriate option. Although the value of accuracy (99.80) for this method has a negligible difference with the FVBRM, the value of FAR (0.00063) has a noticeable difference by the FAR (0.0010) of FVBRM method. Unlike the value of recall (99.68) that is slightly less than the FVBRM

(99.73), the value of precision (99.83) and F-measure (99.79) of Gain Ratio is higher than the other one.

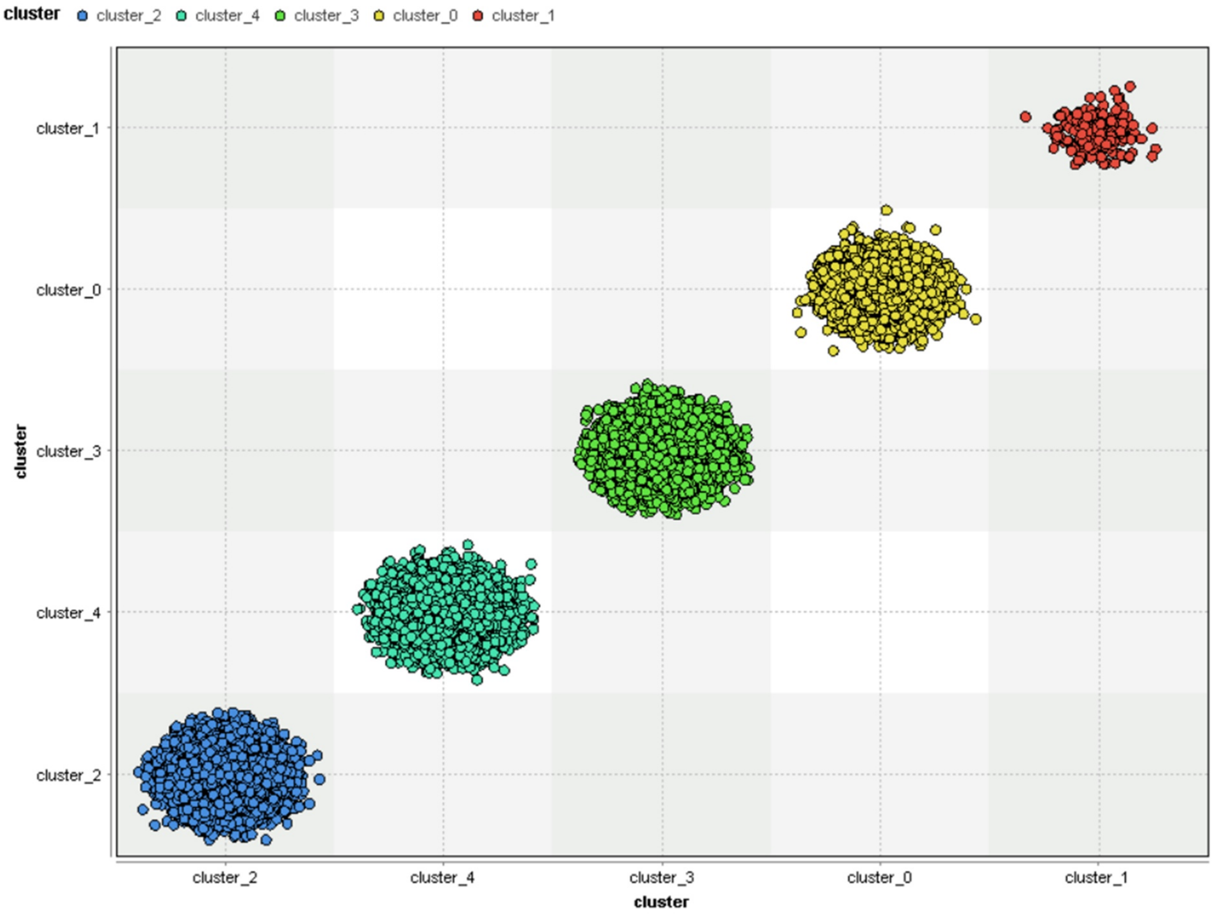
So the most obvious finding to emerge from above analysis is that among mentioned feature selection methods, Gain Ratio works better than others with our proposed model. Note that before starting the clustering, the value of features should be normalized as a preprocessing stage. The following steps were taken to address the clustering:

As it was indicated in subsection 3.3, the K-means clustering is applied in this study. Table 5 displays the number of samples in each cluster. Figures 4 and 5 also illustrate the scatter chart of obtained results from clustering in two ways. Figure 5 is the graphical representation of table 4, while in figure 5 the result of clustering the samples based on separation into attack and normal classes is shown. For instance, a higher percentage of samples in the cluster 4 belong to class 0 which is known as attack.

Table 5. Variables and attributes of cognitive styles

Cluster	Number of samples
0	15067
1	171
2	38752
3	37165
4	34818

Figure 4. scatter chart of clustering



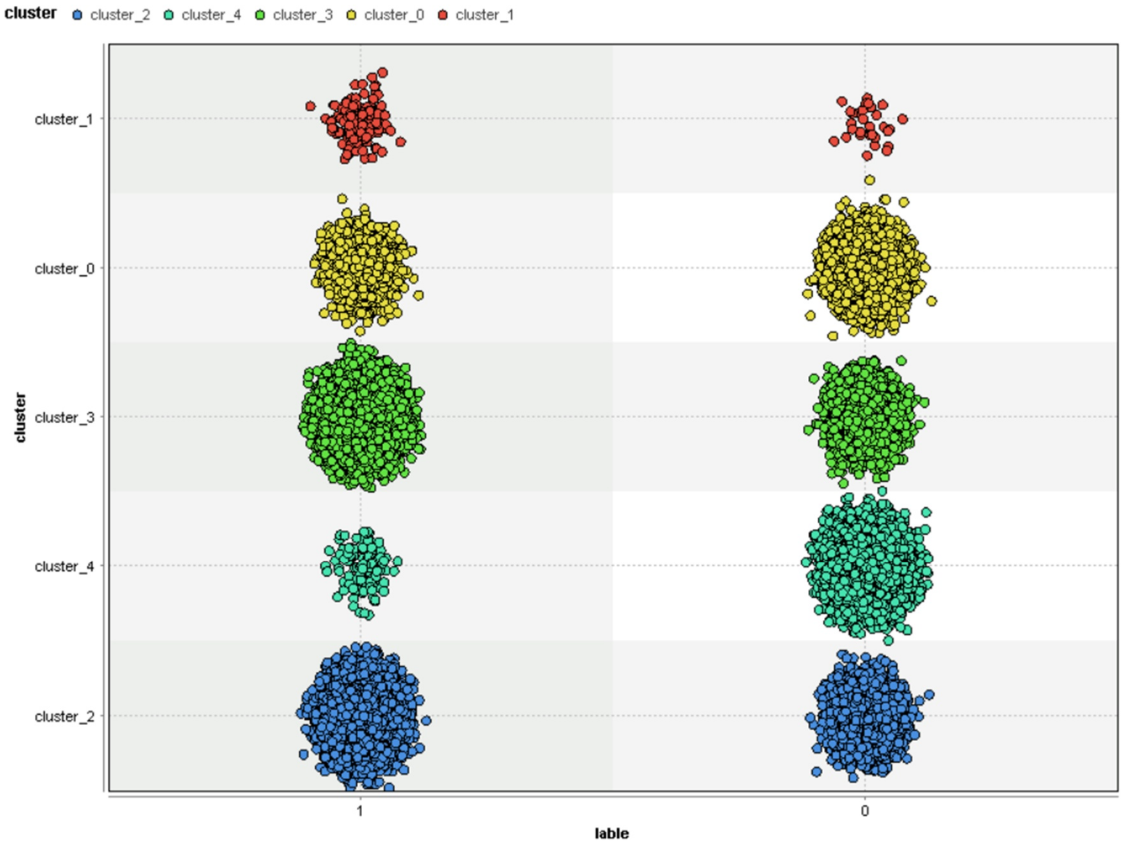


Figure 5. result of clustering based on attack and normal

After completing the clustering phase, 60 percent of each cluster is chosen randomly and that way a new dataset was created. Then the procedure was assessed by the mentioned metrics on the new dataset. The experimental results of this part are shown in table 6.

It was noted previously that the main contribution of clustering in this research is reducing the process time. Thus, the process time is tested in two different scenarios, i.e. without and with clustering the samples. Figures ٦ displays the process time without clustering and in figure ٧ a comparison between the time of testing the model with and without clustering is presented.

Table 6. Variables and attributes of cognitive styles

Feature selection method	Accuracy	Recall	Precision	f-measure	FAR
Gain Ratio	99.76	99.89	99.64	99.77	0.0040
CFS	99.20	99.68	99.88	99.78	0.0010
Wrapper	99.71	99.81	99.64	99.73	0.0041
FVBRM	99.70	99.86	99.63	99.72	0.0047

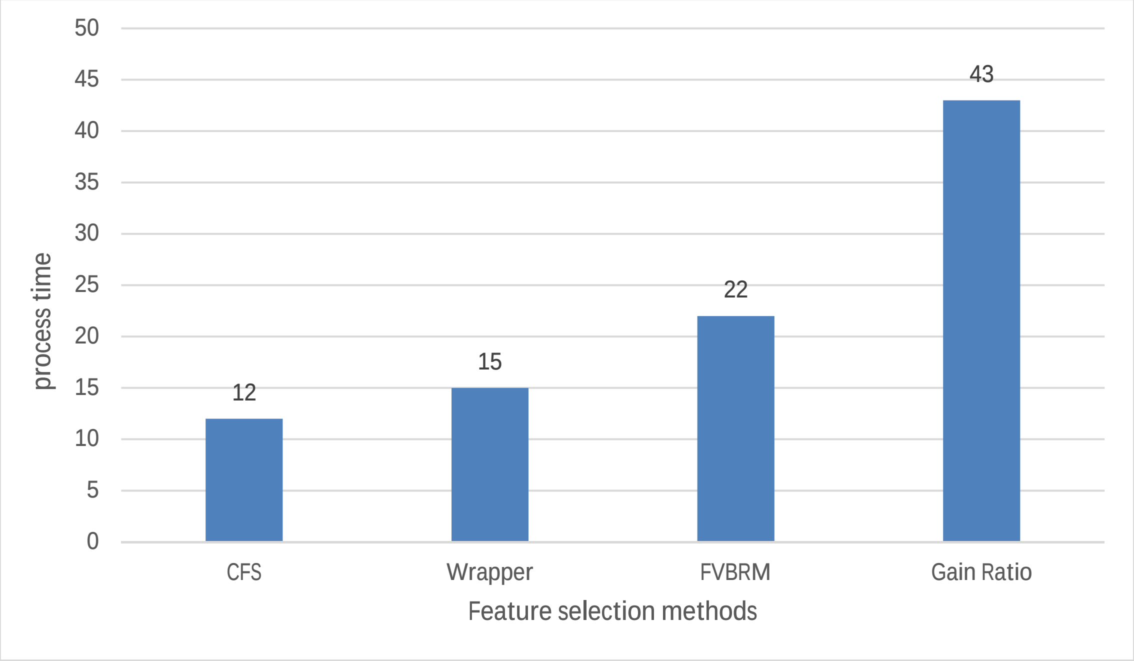


Figure 6. Process time by ensembling

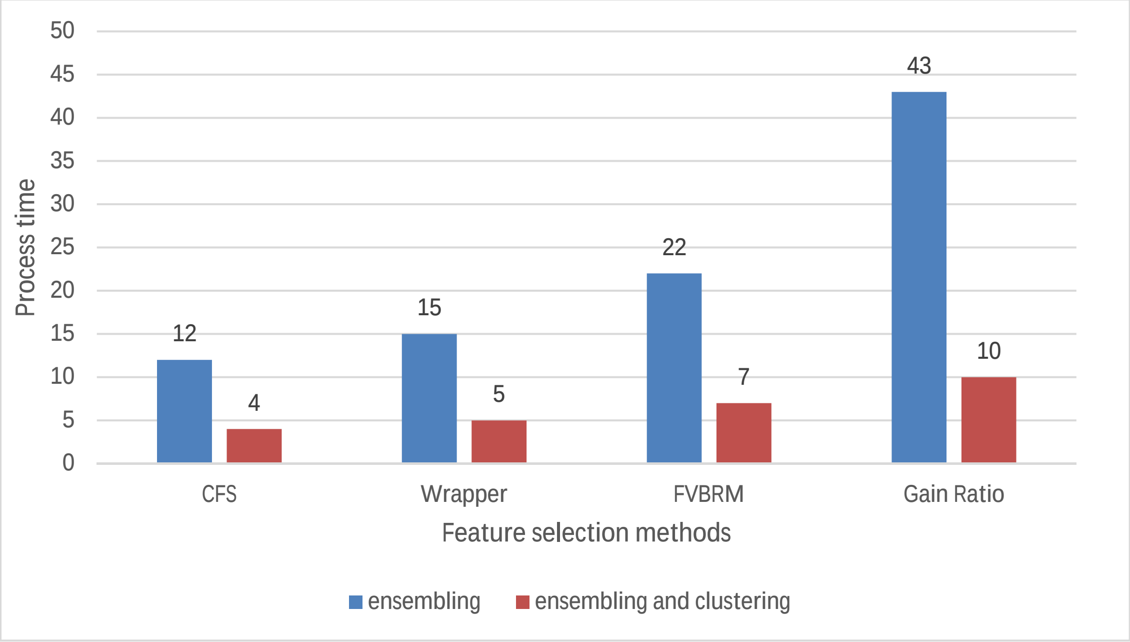


Figure 7. comparing the process time by ensembling and clustering

Turning now to the experimental results discussed in the last part. Based on the result in figure ۶, the process time of our model with Gain Ratio method is the most (43 min) and with CFS is the lowest (12 min). What is striking about this chart is that the number of features can have an effect on the process time. As can be seen in table 1, the number of features for Gain Ratio and CFS are 35 and 10, respectively.

Besides the detailed analysis above, here we can reach the following key findings when clustering the samples is done. From the results in figure ۷ which provide an overview of comparing the process times, the significant affection of clustering is evident. By applying clustering, the process time of all feature selection methods except Gain Ratio has almost reduced to less than the half. It was surprising that the time for Gain Ratio has reached less than a quarter (from 43 to 10 min) when clustering is applied. The results which are shown in table ۶ point out some issue.

One unexpected finding in this table was the extent to which the value of accuracy and F-measure of the proposed method for all feature selection methods reduced. In term of accuracy for some of them, this decrement is slight; for instance, in the Wrapper method, 99.73 has decreased to 99.71, but for CFS the reduction is a little more than the others (from 99.77 to 99.20). Although the value of precision for all methods have also been reduced (expect for CFS), the value of recall has increased after clustering. Data from this table can be compared with the data in table ۴. Similar to the results without clustering, Gain Ratio has the highest accuracy (99.76) and lowest FAR (0.040) than other method. Taken together, the results in table ۶ and figure ۷ provide important insights into our proposed model. As the goal of ensembling the classifiers was improving the accuracy of IDS and declining the FAR and reducing the process time by applying clustering, Wrapper method could be a proper feature selection method. Although the value of accuracy for this technique is less than Gain Ratio, it takes less training time (5 min) in comparison with Gain Ratio (10 min). This time is exactly the half of the one of Gain Ratio method. In order to verify the effectiveness of the proposed method, we compare the performance of our model with other existing approaches. Table 7 illustrates the comparison. It is clear from the table that the proposed method has the best performance among the others.

Table 7. accuracy comparison with other model

Feature selection method	Classifier(s)	dataset	Accuracy
Principal Component Analysis (PCA)	Bagging Algorithm (base classifier -SVM)	NSL-KDD	88.28%[46]
Gain Ratio	Bagging (Base classifier - J48)	NSL-KDD	84.25%[15]
Wrapper	**KNN+DT+Deep Learning**	**NSL-KDD**	**99.71**

5. Conclusion

Detecting the cyberattacks is an essential action in the field of computer network security. Intrusion detection system plays an important role in identifying various network attacks. There has been plenty of research in the area of designing IDS, but each of them has its own disadvantages. Thus, there is still more opportunity to improve the performance of IDS. This study was set out to propose an IDS by ensembling three classifiers (decision tree, K nearest neighbors and deep learning) which were not used in the previous works. Another aim of this project was to reduce the process time, and K-means clustering was used to this end. Based on the results, it was indicated that our approach outperforms the other works in term of accuracy. For future work, as there exist different kinds of classification algorithms, a combination of another group of them and even applying other methods of ensembling might leads to a higher accuracy in the future.

References

[1] Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., ... & Wang, C. (2018). Machine Learning and Deep Learning Methods for Cybersecurity. IEEE Access.

[2] Jabbar, M. A., Aluvalu, R., & Reddy, S. (2017, February). Cluster based Ensemble classification for Intrusion Detection System. In Proceedings of the 9th International Conference on Machine Learning and Computing (pp. 253-257). ACM

[3] Siddique, K., Akhtar, Z., Khan, M. A., Jung, Y. H., & Kim, Y. (2018). Developing an Intrusion Detection Framework for High-Speed Big Data Networks: A Comprehensive Approach. KSII Transactions on Internet and Information Systems (TIIS), 12(8), 4021-4037.

[4] Ashfaq, R. A. R., Wang, X. Z., Huang, J. Z., Abbas, H., & He, Y. L. (2017). Fuzziness based semi-supervised learning approach for intrusion detection system. Information Sciences, 378, 484-497.

[5] Sherasiya, T., & Upadhyay, H. (2016). Intrusion detection system for internet of things. Int. J. Adv. Res. Innov. Ideas Educ.(IJARIIE), 2(3).

[6] Li, S. H., Kao, Y. C., Zhang, Z. C., Chuang, Y. P., & Yen, D. C. (2015). A network behavior-based botnet detection mechanism using PSO and K-means. ACM Transactions on Management Information Systems (TMIS), 6(1), 3.

[7] Al-Jarrah, O. Y., Alhussein, O., Yoo, P. D., Muhaidat, S., Taha, K., & Kim, K. (2016). Data randomization and cluster-based partitioning for botnet intrusion detection. IEEE transactions on cybernetics, 46(8), 1796-1806.

[8] Roshan, S., Miche, Y., Akusok, A., & Lendasse, A. (2018). Adaptive and online network intrusion detection system using clustering and Extreme Learning Machines. Journal of the Franklin Institute, 355(4), 1752-1779.

[9] Aburomman, A. A., & Reaz, M. B. I. (2017). A survey of intrusion detection systems based on ensemble and hybrid classifiers. Computers & Security, 65, 135-152.

[10] Mazini, M., Shirazi, B., & Mahdavi, I. (2018). Anomaly network-based intrusion detection system using reliable hybrid artificial bee colony and AdaBoost algorithms. Journal of King Saud University-Computer and Information Sciences.

[11] Kabir, E., Hu, J., Wang, H., & Zhuo, G. (2018). A novel statistical technique for intrusion detection systems. Future Generation Computer Systems, 79, 303-318.

[12] Nascimento, G., & Correia, M. (2011, June). Anomaly-based intrusion detection in software as a service. In Dependable Systems and Networks Workshops (DSN-W), 2011 IEEE/IFIP 41st International Conference on (pp. 19-24). IEEE.

[13] Sheikh, N. U., Rahman, H., Vikram, S., & AlQahtani, H. (2018). A Lightweight Signature-Based IDS for IoT Environment. arXiv preprint arXiv:1811.04582.

[14] Nguyen, S. N., Nguyen, V. Q., Choi, J., & Kim, K. (2018, February). Design and implementation of intrusion detection system using convolutional neural network for DoS detection. In Proceedings of the 2nd International Conference on Machir
34-38). ACM.

[15] Pham, N. T., Foo, E., Suriadi, S., Jeffrey, H., & Lahza, H. F. M. (2018, January). Improving performance of intrusion detection system using ensemble methods and feature selection. In Proceedings of the Australasian Computer Science Week Multiconference (p. 2). ACM.

[16] Gharaee, H., & Hosseinvand, H. (2016, September). A new feature selection IDS based on genetic algorithm and SVM. In Telecommunications (IST), 2016 8th International Symposium on(pp. 139-144). IEEE.

[17] Thanigaivelan, N. K., Nigussie, E., Kanth, R. K., Virtanen, S., & Isoaho, J. (2016, January). Distributed internal anomaly detection system for Internet-of-Things. In Consumer Communications & Networking Conference (CCNC), 2016 13th IEEE Annual (pp. 319-320). IEEE

[18] Aburomman, A. A., & Reaz, M. B. I. (2016). A novel SVM-kNN-PSO ensemble method for intrusion detection system. Applied Soft Computing, 38, 360-372.

[19] Hamed, T., Dara, R., & Kremer, S. C. (2018). Network intrusion detection system based on recursive feature addition and bigram technique. Computers & Security, 73, 137-155.

[20] https://www.unb.ca/cic/datasets/ids.html

[21] Gaikwad, D. P., & Thool, R. C. (2015). Intrusion detection system using bagging with partial decision treebase classifier. Procedia Computer Science, 49, 92-98

[22] Ambusaidi, M. A., He, X., Nanda, P., & Tan, Z. (2016). Building an intrusion detection system using a filter- based feature selection algorithm. IEEE transactions on computers, 65(10), 2986-2998.

[23] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[24] https://www.unb.ca/cic/datasets/nsl.html

[25] http://www.takakura.com/kyoto_data/

[26] Gholipour Goodarzi B, Jazayeri H, Fateri S. Intrusion detection system in computer network using hybrid algorithms (SVM and ABC). Journal of Advances in Computer Research. 2014 Nov 1;5(4):43-52..

[27] Shokripoor Bahman Bigloo I. A Parallel Genetic Algorithm Based Method for Feature Subset Selection in Intrusion Detection Systems. Journal of Advances in Computer Research. 2019 May 1;10(2):1-6.

[28] Siddiqui, A. K., & Farooqui, T. (2017). Improved Ensemble Technique based on Support Vector Machine and Neural Network for Intrusion Detection System. INTERNATIONAL JOURNAL ONLINE OF SCIENCE, 3(11).

[29] Post, M. J., van der Putten, P., & van Rijn, J. N. (2016, October). Does feature selection improve classification? a large scale experiment in OpenML. In International Symposium on Intelligent Data Analysis (pp. 158-170). Springer, Cham.

[30] Sindhu, S. S. S., Geetha, S., & Kannan, A. (2012). Decision tree based light weight intrusion detection using a wrapper approach. Expert Systems with applications, 39(1), 129-141.

[31] Mukherjee, S., & Sharma, N. (2012). Intrusion detection using naive Bayes classifier with feature reduction. Procedia Technology, 4, 119-128.

[32] Berkhin, P. (2006). A survey of clustering data mining techniques. In Grouping multidimensional data (pp. 25-71). Springer, Berlin, Heidelberg.

[33] Moradi, P., Ahmadian, S., & Akhlaghian, F. (2015). An effective trust-based recommendation method using a novel graph clustering algorithm. Physica A: Statistical mechanics and its applications, 436, 462-481.

[34] Fahim, A. M., Salem, A. M., Torkey, F. A., & Ramadan, M. A. (2006). An efficient enhanced k-means clustering algorithm. Journal of Zhejiang University-Science A, 7(10), 1626-1633.

[35] Zaki, M. J., Ho, C. T., & Agrawal, R. (1999, March). Parallel classification for data mining on shared-memory multiprocessors. In Data Engineering, 1999. Proceedings., 15th International Conference on (pp. 198-205). IEEE.

[36] Amor, N. B., Benferhat, S., & Elouedi, Z. (2004, March). Naive bayes vs decision trees in intrusion detection systems. In Proceedings of the 2004 ACM symposium on Applied computing(pp. 420-424). ACM.

[37] Adeniyi, D. A., Wei, Z., & Yongquan, Y. (2016). Automated web usage data mining and recommendation system using K-Nearest Neighbor (KNN) classification method. Applied Computing and Informatics, 12(1), 90-108.

[38] Tsai, C. F., Hsu, Y. F., Lin, C. Y., & Lin, W. Y. (2009). Intrusion detection by machine learning: A review. Expert Systems with Applications, 36(10), 11994-12000.

[39] Tang, T. A., Mhamdi, L., McLernon, D., Zaidi, S. A. R., & Ghogho, M. (2016, October). Deep learning approach for network intrusion detection in software defined networking. In Wireless Networks and Mobile Communications (WINCOM), 2016 International Conference on (pp. 258-263). IEEE.

[40] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural networks, 61, 85-117

[41] Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials, 18(2), 1153-1176.

[42] Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., & Herrera, F. (2012). A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42(4), 463-484.

[43] Jovic, A., Brkic, K., & Bogunovic, N. (2014, May). An overview of free software tools for general data mining. In Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on (pp. 1112-1117). IEEE.

[44] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009, July). A detailed analysis of the KDD CUP 99 data set. In Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on (pp. 1-6). IEEE.

[45] Tiwari, P., Dao, H., & Nguyen, G. N. (2017). Performance evaluation of lazy, decision tree classifier and multilayer perceptron on traffic accident analysis. Informatica, 41(1).

[46] Tengl, S., Zhang, Z., Teng, L., Zhang, W., Zhu, H., Fang, X., & Fei, L. (2018, May). A Collaborative Intrusion Detection Model using a novel optimal weight strategy based on Genetic Algorithm for Ensemble Classifier. In 2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design ((CSCWD)) (pp. 761-766). IEEE.