ICIC Express Letters
Volume **13**, Number **2**, February **2019**
 ISSN 1881-803X
pp. **93–101**

DATA PREPROCESSING AND FEATURE SELECTION FOR MACHINE LEARNING INTRUSION DETECTION SYSTEMS

Tohari Ahmad and Mohammad Nasrul Aziz

Department of Informatics
Institut Teknologi Sepuluh Nopember
Kampus ITS, Surabaya, Jawa Timur 60111, Indonesia
tohari@if.its.ac.id; nasrul.17051@mhs.its.ac.id

Received July 2018; accepted October 2018

Abstract. *Flow-based anomaly detection is an issue that still grows in a computer network security environment. Many previous studies have applied data mining as a method for detecting anomaly in an intrusion detection system (IDS). In this paper, we further apply data mining to classifying those anomaly data. This is based on the facts that there are many data which are not ready for use by a classification algorithm. In addition, that algorithm may use all features which actually are not relevant to the classification target. According to these two problems, we define two steps: pre-processing and feature selection, whose results are classified by using k-NN, SVM, and Naive Bayes. The experimental results show that such pre-processing and combination of CFS and PSO are better to apply to SVM which is able to achieve about 99.9291% of accuracy on KDD Cup99 dataset.*
Keywords: Intrusion detection system, Network security, Data mining, Feature selection

1. **Introduction.** In the industrial revolution period, computing technology has grown fast. The development of technology is also followed by that of computer network technology. Computer networks connected to the Internet provide a lot of convenience in accessing information from around the world. However, this connection actually increases the possibility of deviating the system security. Computers have become easily accessible and at risk to be infiltrated by other parties who want to access the computer. Consequently, computer systems may be attacked anytime. This condition is very dangerous for computer systems of an organization that contains confidential data which should be accessible by legitimate users only. Some types of threats may occur, such as tapping or stealing confidential data.

Intrusion detection system (IDS) detects an attack in a computer network by analyzing current packets leading to the network. To make good decisions, the implemented machine learning must use good data (complete, correct, consistent and integrated). Prior being mined, the data needs to be pre-processed to ensure their quality. In addition, many features in the data for building a model can also reduce the performance of classification. This is because not all features are in accordance with the target classification results. It requires techniques for selecting important and relevant features for data and reducing irrelevant ones.

In this research, we conduct a flow-based anomaly detection system using machine learning. It comprises several phases: data collection, data pre-processing, classification, and performance testing. For evaluation, we use three datasets: Kyoto 2006, KDD Cup99 and UNSWNB15. Each of those data has relatively a large number of features, so we do selection to handle only important features. Normalization and discretization of data are

DOI: 10.24507/icicel.13.02.93

the steps of pre-processing. Next, we evaluate this algorithm by exploring existing classification methods to find out whether the selected features have an impact on increasing the detection accuracy.

In this process, we combine Correlation-based Feature Selection technique (CFS) and Particle Swarm Optimization (PSO) to perform feature selection. In the classification phase, three machine classification learning methods are compared: Naive Bayes (NB), Support Vector Machine (SVM), and k-Nearest Neighbor (k-NN). Furthermore, we compare the performance of before with after the feature selection. The results are obtained by performing pre-processing data (data normalization and data discretization) and CFS-PSO features selection impacted on the level of classification performance by showing an increase in accuracy.

2. **Progress of IDS.** Some research on machine learning IDS has been done, including: that in [2] which designs a flow-based IDS using two machine learning methods: decision tree J48 and Multilayer Perceptron (MLP). For this purpose, they use UNSWNB15 dataset for testing. They find that the use of J48 produces better accuracy rate than just MLP, which is 0.985 and 0.910, respectively. Additionally, they also find that increasing the number of layers raises the accuracy; however, it takes longer to process. Another research which is conducted by Muttaqien and Ahmad [3] employs feature selection, clustering and feature transformation on NSL-KDD dataset and Kyoto 2006. Here, clustering is done by implementing k-means whose radius of clusters is to be the threshold for grouping the data. It is shown that their proposed method is able to improve the classification performance through the accuracy test, whose best result on NSL-KDD is 97.42% and on Kyoto 2006 is 99.72%. Thaseen and Kumar [1] conduct a study of IDS by making a normalization stage, rank-based chi-square feature selection, and classification with multiple SVM. Their method is tested on NSL-KDD and KDD Cup99 datasets. It is demonstrated that their method is more suitable for NSL-KDD than KDD Cup99.

Kasliwal et al. [4] develop a hybrid model using the Latent Dirichlet Allocation (LDA) and the Genetic Algorithm (GA). LDA is used to identify the optimum set of attributes, while GA is used to calculate initial scores for fitness value evaluation to obtain new features used in the classification of KDD Cup99 datasets. Ikram and Cherukuri [5] propose a hybrid IDS model with two approaches: Principal Component Analysis (PCA) and Support Vector Machine (SVM). The step to do is to perform parameter selection optimization with PCA on the SVM classifier kernel. With optimization of punishment factors and gamma kernel parameters, this method can improve classification performance and reduce classification time in training and testing.

In further research, Mukherjee and Sharma [6] investigate three correlation-based feature selection methods applying to feature selection issues: correlation-based, information reinforcement and gain ratios. In addition, they also propose a new method for feature selection using feature vitality-based reduction method to identify and then iteratively reduce less important features. Using the Naive Bayes classification, they measure the performance with the reduced dataset. The results show that reducing the number of features provides better performance.

Akashdeep et al. [7] propose an IDS with feature selection based on the acquisition and correlation of information. To select the features, they analyze the information acquisition and correlation results. From these data, a new approach is proposed to sort out features that is useful. For this purpose, they use feed forward neural network classification in training and testing, in addition to the normalization of the dataset. Compared to that without feature selection, the use of feature selection shows better results. Amiri et al. [8] apply two feature selection methods to KDD Cup99. They compare the mutual information-based feature selection method with correlation coefficient of linear and

nonlinear measure for feature selection. It is depicted that this method has high accuracy in detecting Remote to Login (R2L) and User to Remote (U2R) attacks.

3. **Proposed Framework.** In this section, we provide the proposed framework that includes pre-processing, feature selection, and classification, inspired by [9, 10]. The details of this method are presented in Figure 1.

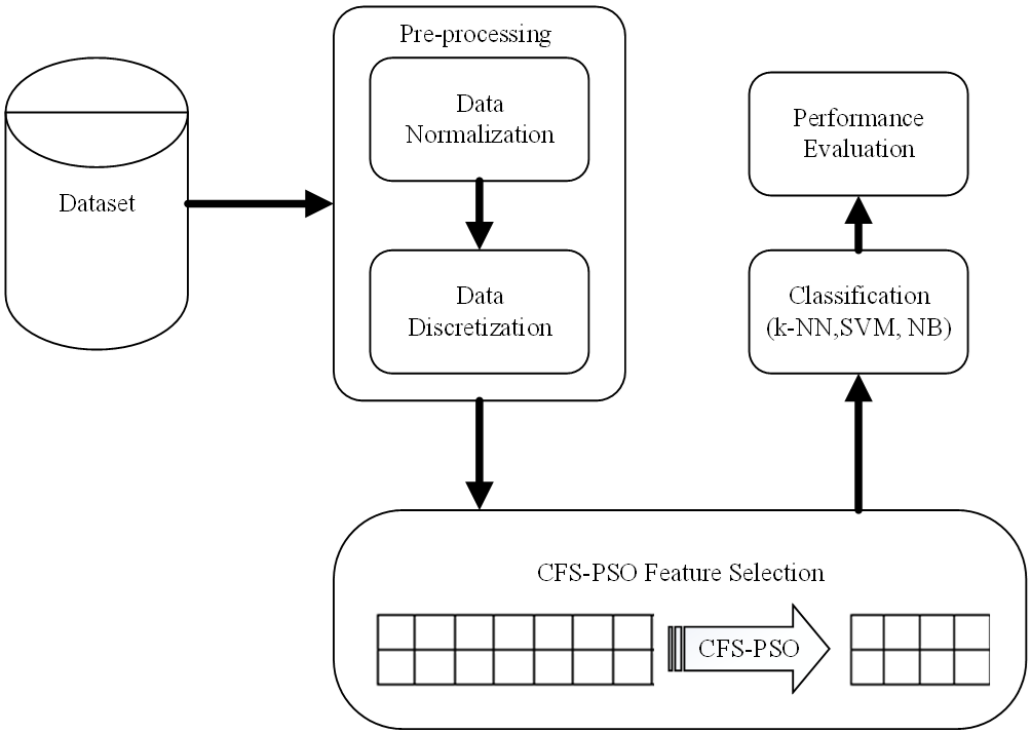

FIGURE 1. Proposed framework

3.1. **Data pre-processing.** This is the initial phase of the method. Pre-processing data is intended to transform the raw data to a format that is easier and more effective to use for future processing steps. In the early phase we normalize data using the min-max method. Normalization can improve the training time because all data used in the training have the same scale, for example, in the range of 0 and 1. For this purpose, we implement (1) as provided in [11], where X_{norm} is the result of normalization, X is the initial value before being normalized. Here, $X_{\max}$ and $X_{\min}$ represent the maximum and the minimum values of each feature, respectively.

$$X_{norm} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{1}$$

The second method we use for pre-processing is data discretization using the Minimum Description Length (MDL) [12]. Data discretization is defined as the process of converting the value of the continuous data attribute into a series of finite intervals by minimizing the loss of information in the data [13].

3.2. **Feature selection.** Feature selection aims to select the best feature in the data set. Machine learning algorithms can classify the data into a set of class features and targets. Machine learning or pattern recognition applications, feature domains have grown from tens to hundreds of variables or features used in the application. Several techniques are developed to overcome the problem of reducing irrelevant and excessive variables. Feature selection (variable elimination) helps understand the data, reduces computing needs, reduces dimensional curse effects and improves the performance [14].

This research uses the Correlation-based Feature Selection (CFS) technique that is optimized with Particle Swarm Optimization (PSO). CFS is a method for selecting features by exploring a multivariate approach as a filter for selecting subset features. This is to find features whose correlation to the target class is high. In fact, CFS may choose a subset of less optimum features if the feature expression values are located in a less large search area [15]. CFS accepts a ranking pattern on a feature subset derived from a heuristic evaluation function based on the correlation level. The bias of the evaluation function is a set of features that are highly correlated with the class and not correlated to each other. For features that have low correlation tendency towards the class, the respective features must be ignored [9]. The redundant feature must be filtered because it is highly correlated with one or more of the features. Acceptance of the feature depends on how far it can predict the class in the sample space area that has not been predicted by other features. To calculate the function of the CFS feature subset, evaluation can be written in (2) [10].

$$M_s = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \tag{2}$$

M_s is the "merit" heuristic of the feature subset S which contains the feature k. Here, $\overline{r_{cf}}$ is the average of correlation of the feature class $f \in S$; and $\overline{r_{ff}}$ is the average of the intercorrelation features. The numerator at (2) can be considered to give an indication of how the predictive class of a feature set is denominator about how much redundancy exists among its features. Furthermore, this formula forms the core of the CFS and determines the ranking of the feature subset in the search space of all possible feature sets. For the search and optimization of the features that CFS has established the PSO algorithm is used, the detail of the CFS-PSO algorithm can be seen in Figure 2, as an improvement of the selection flow obtained from [9].

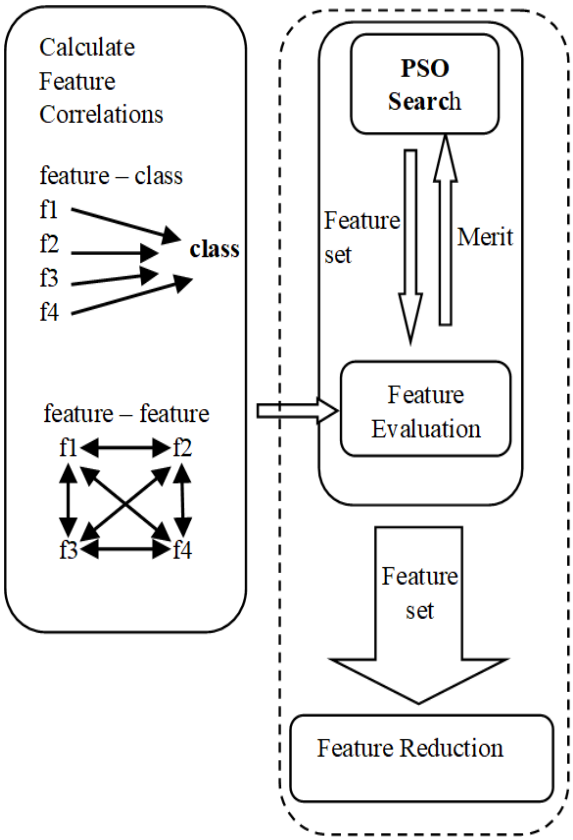

FIGURE 2. Flow of CFS-PSO as an improvement of [9]

PSO is an algorithm which is often used to optimize decision making. Additionally, it can also be implemented for path searching. Generally, the PSO algorithm has four stages: initialization, evaluation, update and termination as in depicted in Figure 3. To find an optimum value, the PSO algorithm repeats until the optimum value is obtained. The initialization stage consists of sub-stages c_1 and c_2 where they are the learning factor constants for the particle's capability and the influence of the set, respectively. The iteration stage serves to find the best positioned particle (Pbest) and the best position of all the particles present in a population (Gbest).

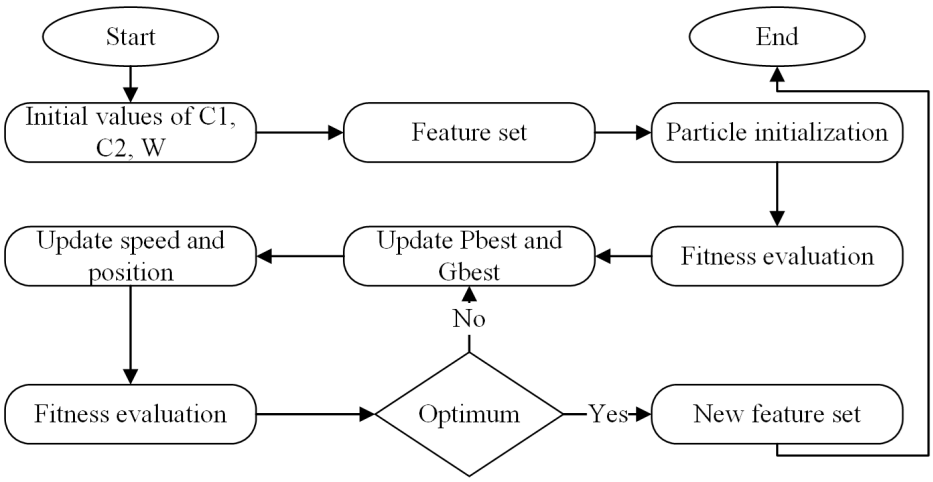

FIGURE 3. Flow of PSO

PSO optimizes the process by continually counting prospective solutions using a quality reference. The algorithm optimizes the problem by moving the particle/potential solution in the problem space using the speed function (3) and the function to locate the position (4) of the particle [10].

$$v_i^{t+1} = w \times v_i^t + c_1 \times r_1 \times \left(gb - x_i^t\right) + c_2 \times r_2 \times \left(x_i^* - x_i^t\right) \tag{3}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{4}$$

In (3) and (4), t is a representation of the iteration in the optimization process, and w is the weight of inertia that controls the impact of the previous velocity at a new velocity. Parameters c_1 and c_2 are acceleration constants or learning parameters and r_1 and r_2 are uniformly distributed random values between 0 and 1. Speed v can have any value but is usually limited in the range of $[0, v_{\max}]$.

3.3. **Classification.** Classification is grouping data based on a label or target class. So, the algorithms to solve classification problems are categorized into supervised learning. In this research, we use three classification algorithms to compare the impact of our proposed framework on classification performance. We use the k-Nearest Neighbor (k-NN) classification, Support Vector Machine (SVM), and Naive Bayes (NB) as a pilot in our research.

The k-NN classifies objects based on the learning data closest to the object. This method aims to classify new objects based on attributes and training samples. It is very simple and easy to implement, similar to the clustering technique, which is to group a new data based on the new data distance to some data/nearest neighbor.

Before finding the distance between the data and the neighbor, it needs to determine the value of neighboring k (neighbor). Then, to define the distance between two points, i.e., the point in the training and the point in the testing, the Euclidean formula (5) is

used. In this formula, $d(a, b)$ is the Euclidean distance, x is the first data, y is the second data of i feature, and n is the total number of features.

$$d(a, b) = a_0 + \sum_{i=0}^{n} (x_i + y_i)^2 \tag{5}$$

The SVM concept can be simply explained as a method for finding the best hyperplane used as a separator of two classes. Discrimination boundaries or alternate line in SVM can be simplified into two class members from $+1$ and -1. In SVM there is also a margin that is the closest distance between hyperplane the closest pattern of each class. Support vector is obtained from the selection of the nearest pattern. So, it can be said the core of the SVM algorithm is the determination of the location of the best hyperplane.

Naive Bayes is a supervised machine learning algorithm that performs a probabilistic classification by computing a set of probabilities from the sum of the frequency and the combination of values of the processed dataset. It is based on the assumption that the value of an attribute is a condition of values that are not bound each other when given the value of output. If it is assigned, the probability will be observed simultaneously to obtain the individual probability value.

4. **Experimental Results.** We do experiments based on some machine learning algorithms. Further testing is performed on the pre-processing data and also feature selection to get a better accuracy level. We divide categories according to the available dataset: KDD Cup99, Kyoto 2006 and UNSWNB15. To test the performance of each classification, we use the confusion matrix to look for True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values. Then the results are used to find the accuracy, True Positive Rate (TPR), and False Negative Rate (FPR).

TP is a successful attack classified as attack by the system; FP is an activity that should be normal but is marked as an attack by the system; TN is a normal flow that can be classified correctly by the system; the FN is the attack flow but is not detected by the system [3]. The experimental results on various datasets are described as follows.

4.1. **KDD Cup99.** KDD Cup99 [16] has 41 features and has a label whether it is normal or attack. The attack label on KDD Cup99 can be categorized into 24 types of attack label as training data. Among the label attacks are: DoS, User to Root (U2R), Remote to Local (R2L) and Probing Attacks. KDD Cup99 has several features that are divided into several clusters including: basic features of TCP connections (duration, byte, TCP mark, port number), traffic features obtained from time intervals every two seconds taking account connection to the host, the content feature, generated from unpacking the information contained on the payload.

In KDD Cup99 data, we carry out two probation scenarios. The first is resampling 10% of the total data to be used as training data and data testing in three machine learning classifications. The 10 fold cross validation method is used to form training and testing data. The performance test results from this first scenario of KDD Cup99 data can be found in Table 1. It depicts that k-NN classification performs the best with 99.88% of accuracy and 99.9% of TPR.

Table 1. The first and second test results of KDD Cup99

Classifier	**Results (%)**		
	Accuracy	**TPR**	**FPR**
k-NN	99.8745/99.8765	99.9/99.9	0/0
SVM	99.8583/99.9291	99.9/99.9	0/0
Naive Bayes	91.7723/91.4655	91.8/99.5	0/0

The second experiment on KDD Cup99 data is continued by normalizing and discretizing data, followed by performing feature selection. From feature selection, we obtain those with better correlation. Among 41 features, we have 13 features: protocol_type, service, flag, src_bytes, dst_bytes, land, wrong_fragment, num_failed_logins, lroot_shell, count, diff_srv_rate, dst_host_srv_count, dst_host_same_src_port_rate. Still in Table 1, by using these features, the results of similar experiment to the first scenario are presented. It is shown that there is an increase of accuracy of all methods. More specifically, SVM achieves the highest accuracy. However, concerning the difference, Naive Bayes generates the highest, where it goes up from 91.7723% to 99.4655%, while k-NN is the lowest, i.e., only from 99.8745% to 99.8765%. Similar patterns are applied to TPR, where Naive Bayes has the highest increase. For those both scenarios, the value of FPR remains 0, which means that all normal data are recognized correctly. Results of this experiment can be compared with those done by [17] using the combining Gated Recurrent Unit (GRU) and SVM, where they achieve 91.2% of TPR and ours is 99.9%. In more detail, GRU is developed from long-short term memory as a part of recurrent neural network. Here, the softmax and cross-entropy functions are implemented.

4.2. **Kyoto 2006.** The Kyoto 2006 dataset contains real data traffic built at Kyoto University by placing 348 honeypots outside or inside the Kyoto University [18]. The period of placement of honeypots started from November 2006 to August 2009. This dataset comprises 14 features. Furthermore, there are 10 additional features that are used for further analysis purposes.

The same experiment is done, i.e., scenarios 1 and 2, whose results are provided in Table 2. It is found that the use of the feature selection method is also able to rise the accuracy, where the highest increase is obtained by Naive Bayes, similar to that in KDD Cup99. More specifically, there is about 7% increase, from 92.921% to 99.184%. Furthermore, its TPR goes up around 7%, while that of SVM is around 0.3%. In terms of FPR, there is a significant decrease for SVM, from 21.3% to 9.4%. However, k-NN and Naive Bayes suffer from higher value. Overall, the accuracy and TPR can be maintained at more than 99%, similar to those in KDD Cup99.

Table 2. The first and second test results of Kyoto 2006

Classifier	**Results (%)**		
	Accuracy	**TPR**	**FPR**
k-NN	99.757/99.766	99.8/99.8	6.4/9.4
SVM	99.547/99.752	99.5/99.8	21.3/9.4
Naive Bayes	92.921/99.184	92.9/99.2	3.3/9.3

We compare the experimental results with those of [3] which also applies the method to Kyoto 2006 dataset. They have 99.72% of accuracy, while this proposed method is able to achieve 99.76% of accuracy, especially for the k-NN classifier.

4.3. **UNSWNB15.** UNSWNB15 dataset [19] uses an automated attack generator tool called IXIA Perfect Storm, to implement nine types of real and updated attacks against multiple servers. They collect traces of tcp dump from network traffic, for a total duration of 31 hours in early 2015. UNSWNB15 has 49 features consisting of several feature groups: flow features, basic features, feature features, time features, additional generated features, and connection features. Concerning the label, the data contain two classes of label: normal and attack. The attack can be further categorized into nine types: fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms [19].

Table 3. The first and second test results of UNSWNB15

Classifier	Results (%)		
	Accuracy	**TPR**	**FPR**
k-NN	92.8425/92.8767	92.3/92.9	16.2/14.8
SVM	84.7781/92.2151	84.8/92.2	17.7/16.2
Naive Bayes	75.9211/84.7781	75.9/84.8	15.3/17.7

The experimental results on this dataset are presented in Table 3. It is found that the use of feature selection is also useful for increasing the performance when this dataset is used, even though the accuracy is lower than that of others.

It is found that the use of k-NN in the first experiment on UNSWNB15 dataset still has the highest performance than the other two classifiers, with 92.84% of accuracy, 92.3% of TPR, and 16.2% of FPR. In the second experiment, it is depicted that k-NN is also the highest, even though its increase is not significant. This condition is different from the other two methods.

Concerning TPR, this proposed method is also able to rise the performance with various values, in the range of 0.6%-8.9%. This improvement is followed by the reduction of FPR for k-NN and SVM, which means good. In the case of Naive Bayes, the FPR inconsiderably rises about 2.5%.

In other research, [2] implements MLP and J48 on the UNSWNB15 dataset. Their method obtains an accuracy of 91.0% for MLP and 98.5% for J48; while this proposed method achieves 92.8767% for k-NN.

5. **Conclusion.** In this research, we have proposed the combination of PSO and CFS for selecting features, which are then tested over three methods (i.e., k-NN, SVM and Naive Bayes) on different data sets (KDD Cup99, Kyoto 2006 and UNSWNB15). Before this selection, the data are firstly pre-processed. This comprises normalization and discretization data.

It is shown that, overall this process is able to improve the performance, in terms of the accuracy, TPR and FPR. In more detail, the best performance is achieved by SVM, where the accuracy, TPR and FPR are 99.9291%, 99.9% and 0, respectively. Furthermore, the evaluation is also carried out by comparing the performance of the proposed method with another existing one, where in general, the proposed method is superior.

In the future, this proposed method can be implemented to other datasets. This is to measure its capability to work on various characteristics of data. Also, more data reduction may be done to have simpler data. It is intended to reduce running time and complexity.

REFERENCES

[1] I. S. Thaseen and C. A. Kumar, Intrusion detection model using fusion of chi-square feature selection and multi class SVM, *J. King Saud Univ. – Comput. Inf. Sci.*, vol.29, no.4, pp.462-472, 2017.

[2] L. van Efferen and A. M. T. Ali-Eldin, A multi-layer perceptron approach for flow-based anomaly detection, *Proc. of International Symposium on Networks, Computers and Communications (ISNCC)*, pp.1-6, 2017.

[3] I. Z. Muttaqien and T. Ahmad, Increasing performance of IDS by selecting and transforming features, *Proc. of IEEE International Conference on Communication, Networks and Satellite*, pp.85-90, 2016.

[4] B. Kasliwal, S. Bhatia, S. Saini, I. S. Thaseen and C. A. Kumar, A hybrid anomaly detection model using G-LDA, *Proc. of IEEE International Advance Computing Conference*, pp.288-293, 2014.

[5] S. T. Ikram and A. K. Cherukuri, Improving accuracy of intrusion detection model using PCA and optimized SVM, *CIT Journal of Computing and Information Technology*, vol.24, no.2, pp.133-148, 2016.

[6] S. Mukherjee and N. Sharma, Intrusion detection using Naive Bayes classifier with feature reduction, *Procedia Technol.*, vol.4, pp.119-128, 2012.

[7] Akashdeep, I. Manzoor and N. Kumar, A feature reduced intrusion detection system using ANN classifier, *Expert Syst. Appl.*, vol.88, pp.249-257, 2017.

[8] F. Amiri, M. Rezaei Yousefi, C. Lucas, A. Shakery and N. Yazdani, Mutual information-based feature selection for intrusion detection systems, *J. Netw. Comput. Appl.*, vol.34, no.4, pp.1184-1199, 2011.

[9] M. A. Hall, *Correlation-Based Feature Selection for Machine Learning*, Ph.D. Thesis, University of Waikato, 1999.

[10] S. Singh and A. K. Singh, Web-spam features selection using CFS-PSO, *Procedia Comput. Sci.*, vol.125, pp.568-575, 2018.

[11] S. C. Nayak, B. B. Misra and H. S. Behera, Impact of data normalization on stock index forecasting, *International Journal of Computer Information Systems and Industrial Management Applications*, vol.6, pp.257-269, 2014.

[12] U. M. Fayyad and K. B. Irani, Multi-interval discretization of continuous-valued attributes for classification learning, *IJCAI*, pp.1022-1027, 1993.

[13] R. Jin, Y. Breitbart and C. Muoh, Data discretization unification, *Proc. of the 7th IEEE International Conference on Data Mining*, pp.183-192, 2007.

[14] G. Chandrashekar and F. Sahin, A survey on feature selection methods, *Comput. Electr. Eng.*, vol.40, no.1, pp.16-28, 2014.

[15] I. Jain, V. K. Jain and R. Jain, Correlation feature selection based improved-Binary Particle Swarm Optimization for gene selection and cancer classification, *Appl. Soft Comput.*, vol.62, pp.203-215, 2018.

[16] S. Stolfo, W. Fan, W. Lee, A. Prodromidis and P. Chan, *Cost-Based Modeling and Evaluation for Data Mining with Application to Fraud and Intrusion Detection: Results from the JAM Project*, 1999.

[17] A. F. M. Agarap, A neural network architecture combining gated recurrent unit (GRU) and support vector machine (SVM) for intrusion detection in network traffic data, *Proc. of the 10th International Conference on Machine Learning and Computing*, pp.26-30, 2018.

[18] N. B. Aissa and M. Guerroumi, Semi-supervised statistical approach for network anomaly detection, *Procedia Comput. Sci.*, vol.83, pp.1090-1095, 2016.

[19] N. Moustafa and J. Slay, UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set), *Proc. of Military Communications and Information Systems Conference*, pp.1-6, 2015.