Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 167 (2020) 1230–1239

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Computational Intelligence and Data Science (ICCIDS 2019)

Differential Evolution Wrapper Feature Selection for Intrusion Detection System

Faezah Hamad Almasoudy[a], Wathiq Laftah Al-Yaseen[b] , Ali Kadhum Idrees[a]

[a]Department of Computer Science, University of Babylon, Babylon, Iraq
{stud.faaeza.hamad; ali.idrees}@uobabylon.edu.iq
[b]Kerbala Technical Institute, Al-Furat Al-Awsat Technical University, 56001, Kerbala, Iraq
wathuqpro@gmail.com

Abstract

The growing volume of data on the computer network led to increasing the challenges for intrusion detection systems to deal with high dimensions that contain irrelevant and redundant features. This consumes time and difficulty in detecting the attack correctly, with increasing false alarms rate. This problem can be solved by applying dimensionality reduction. In this paper, a wrapper feature selection model based on Differential Evolution technique is proposed for intrusion detection systems. It reduces the number of features by finding the minimum number of features without effecting on the performance of the system. The main idea is to select some features from 41 features of NSL-KDD datasets using Differential Evolution and evaluate these features by computing the accuracy using Extreme Learning Machine. Differential Evolution is continued until obtaining the minimum number of features that satisfy a high accuracy. The results have shown a better detection rate with reduced false alarm rate in five and binary classification. The proposed system achieved an accuracy of 80.15 % and 87.53 for five and binary classification respectively with a reduction in training and testing time.

Keywords: Intrusion Detection System; Feature Selection; Differential Evolution; Extreme Learning Machine; NSL-KDD

1. Introduction

Recently and with increasing the need to use the Internet in all applications and domains, the number of moving

*Corresponding author
E-mail address: ali.idrees@uobabylon.edu.iq

10.1016/j.procs.2020.03.438

packets and the load over the network are increased. Therefore, the most important information is at risk despite the existence of multiple network protection systems such as the firewall system, which is an effective system of protection and prevention. The firewall systems prevent unauthorized to enter the systems but lack the ability to track the monitoring after passing the information. It will not be able to detect any attack managed to bypass it. Therefore, to keep the network under surveillance, it must be surrounded by the intrusion detection system IDS [1]. The intrusion was be defined as a threat to the confidentiality, integrity, and availability of information caused by either misuse by authorized users of the system have certain privileges, or because unauthorized users could access the system through certain gaps [2]. Generally, there are two types of intrusion detection system depending on the detection methods [3]: Signature-based IDS and Anomaly-based IDS. Signatures or misuse method uses rules that are pre-stored as a basis for comparison representing known types of attack, therefore the attack that non-stored was not be detected. While anomaly builds a file of samples describing the behavior or normal activity of the system and any abnormality from these activities is evidence of the presence of attack [4, 5]. Network data contains a large number of features which redundant and irrelevant features. Therefore, to analyze all features, it consumes time which represents one of the most challenges. Hence, it is not appropriate to use all features by the IDS. In addition, some features adversely affect the performance of the detection system. Consequently, we seek to select features that have a positive impact on performance [6, 7]. This paper gives the following contributions:

i. Propose an efficient Wrapper Feature Selection Method for Intrusion Detection System based Differential Evolution and Extreme Learning Machine techniques to reduce the features by removing the redundant features and to detect four categories of attack. It can reduce the data and processing time while selecting the useful features that provide high accuracy.
ii. Improve the performance by increasing the detection rate and reduce false alarm rate with reduction the training and testing time compared with other methods in [15, 16].

The remainder of the paper organized as follow. Section 2 presented the related literature. Section 3 reviewed the background on the methods and resources used in this paper. Section 4 provides the proposed method. The performance evaluation is shown in Section 5. Finally, the conclusion and future work are introduced in Section 6.

2. Related Work

There are many models for the IDS, and most of them have been adopted to reduce the features to make the system more efficient. Some models have used filter feature selection and the other ones have used either wrapper or embedded feature selection. In this paper, we used the wrapper feature selection, therefore this section is focused on presenting previous works related to this type of feature selection with NSL dataset. Vinutha and Poornima [8] employed some features such as Cfs, Chi-square, SU, Gain Ratio, Info Gain, and OneR with applying some ensemble and single classifier by using Weka data mining applied on benchmark NSL datasets until results are showed that the use of AdaBoost improves the classification accuracy. Subba et al. [9] reduced dimensionality using PCA and get on 17 features evaluated by some classifiers SVM, MLP, C4.5, and Naïve Bayes achieved on the multi and binary class. It is realized high accuracy using SVM in both multi and binary classification but it was only tested using training data which does not guarantee. The authors in [10] showed the purpose of feature selection by the effect on the accuracy improvement of IDS. It is executed the experimental on NSL_KDD entire dataset using Random Forest binary Classifier of the detection model based all feature without applying FS. Then it is applied Sequential Floating Search to select the best feature for achieved DR and FP. Hanafi, et al [11] are used Sequential Floating Forward Selection technique to obtain 26 features from the overall, then it is applied to two layer classification: first layer for normal detection using GADG (genetic algorithm Detection Generation) tagged as either normal or attack and the second layer for attacks classification using some classifier such as Naïve Bayes, Decision Tree, J48, BF Tree, RF Tree, and Multilayer perceptron NN . This phase is to label the attack for a specific class by using NSL_KDD and 20% KDD. Each classifier showed the best detection for the type of attack and worse case for another. Nskh et al. [12] proposed PCA to obtain on lower diminution implemented by using 10% KDDCup'99 dataset for training and entire KDDCup'99 dataset for testing which are evaluated using Support Vector Machine and achieved the experiments with PCA and without PCA. They are compared using different kernels and observed the accuracy increase and detection time decrease with PCA. Gupta & Kulariya [13] proposed two frameworks based on feature selection using correlation for the selection of the most relevant feature and using hypothesis testing-based feature selection. The performance frameworks are evaluated using KDD'99, and NSL-KDD'99 dataset. They achieved using NSL high accuracy 80.96 with hypothesis testing-based FS in 2.24 testing time. Gaikwad &Thool [14] proposed GA feature selection technique and find the best 15 features from 41 feature in NSL-KDD data set, and evaluated the test accuracy using Bagging method machine learning to implement IDS with Partial Decision Tree rule as a base classifier. The results showed 99.71% of 10-fold CV and 78.37% of test dataset compared with other classifiers. Ingre & Yadav [15] are applied NSL-KDD dataset with feature reduction using ratio gain and ANN. The results showed a good performance of 81.2% in binary classification with 29 feature and 79.9% in five class classification with 41 features. Pervez & Farid [16] proposed wrapper feature selection to and evaluated using NSL-KDD dataset by employed SVM which achieved 91% accuracy with 3 feature and 99% with 41 features on all training data but on testing achieved 82.37 with 14 features in binary classification. Al-Jarrah et al. [17] proposed two feature selection techniques: Random Forest Forward Selection Ranking (RF-FSR) and Random Forest Backward Elimination Ranking (RF-BER), the results showed efficient accuracy using NSL-KDD dataset but it tested using only 10-cross-validation which does not guarantee same if used entire testing dataset.

In this article, a differential evolution wrapper feature selection for Intrusion Detection System (IDS) is proposed. The main idea is to select some features from 41 features of NSL-KDD data set using (DE) and evaluate this some features by computing the accuracy using Extreme Learning Machine (ELM). DE is continued until obtaining the optimal features with high accuracy. It reduces the number of features by finding the optimal features in order to improve the performance of the system. The results have shown a better detection rate with reduced false alarm rate in five and binary classification.

3. Proposed Method

In this section, the proposed model is explained. The main steps consist of preprocessing the dataset, feature selection, classification and performance evaluation as shown in Figure 1

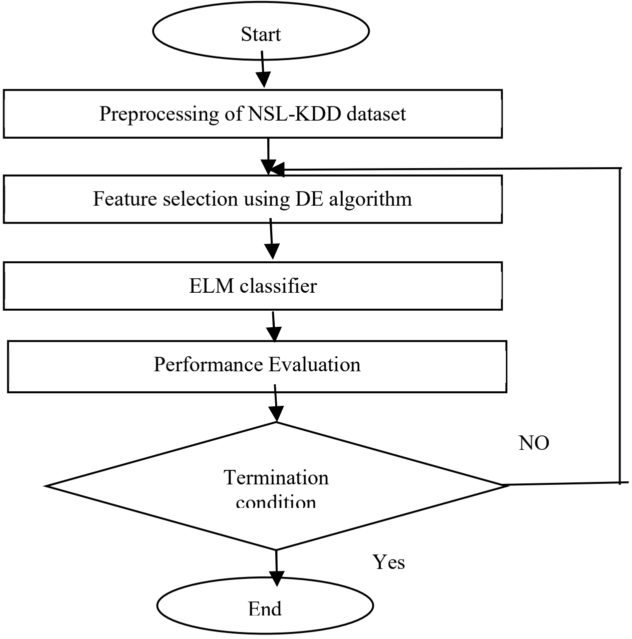

Fig. 1. Proposed method

3.1. Preprocessing of NSL-KDD dataset

The first step is to prepare the data for classification by converting the symbolic features to numeric features during the representation process. For example, the second feature called protocol contains {tcp, udp, and icmp}. It can be converted to {0,1, and 2} for each symbol such that the third and fourth features. In addition, in the last feature, the classes must be converted to a specific number. For example, it represents the normal, DOS, PROBE, R2L, and U2R for 0, 1, 2, 3, and 4 respectively. After the representation, the normalization process is to make all values between (0,1) using the equation (1) to prevent the classifier from basis toward the high values. Table1 shows the preprocessing stages to samples from NSL testing data.

$$x = \frac{x - min}{max - min} \tag{1}$$

Where min is the minimum value in feature, max is the maximum value in feature.

3.2 Feature Selection using DE.

3.2.1 Initialization of Population

The initial population is randomly generated from random numbers in range [0, 41]. Each number is index to position of the feature in the dataset. Since the problem in this case is to make sure that the selected feature must be chosen once in the same chromosome, the solution is tested each feature being selected with the feature chosen before it. An example for chromosome with length 7 and how a solution is represented is introduced as follow.

Table.1 Example for chromosome where each gen represent one feature

6	10	40	16	5	8	33

3.2.2 Mutation

To achieve this process, it must be the size of the population at least consists of four individuals. For each target vector $X_i(G)$ in the current population, the mutant vector mi will be generated as follow:

- Choose three individuals a, b, and c from the current population and $X_i(G) \neq$ a, b, and c such that $a \neq b \neq c$.
- Determine the mutation factor F that is random number in the range [0, 1].
- Calculate the mutant vector by adding the third individual c to the scaled difference of two other individuals a and b.

$$M_i(G+1) = c + F.(a - b) \tag{2}$$

When the value of the mutant vector is less than the low bound 0 or higher than the higher bound 41, this value is ignored and the value of the feature will be selected randomly with ensuring that it is not repeated.

3.2.3 Crossover

This phase is the recombination phase that follows the mutation to produce a trail vector or called offspring from combing the target vector with the mutant vector. This is performed in the pure DE, but in our proposed method, some modification is achieved by combing the mutant vector with random selection to the feature. However, the crossover is based on CR crossover factor, which is a random number in the range of [0, 1].

3.2.4 Selection

After introducing the trail vector, the comparison process occurs with the target vector after calculating the fitness function for them to choose the best one to the next generation.

$$X_i(G+1) = \begin{cases} T_i(G+1) & if\ f\big(T_i(G+1)\big) > f(X_i(G)) \\ X_i(G) & otherwise \end{cases} \tag{3}$$

3.3. Fitness Evaluation using ELM (Classification)

ELM learning machine is applied on the trail and target vector (training and testing) the accuracy is calculated as fitness function.

Algorithm 1. DE feature selection with ELM classifier

Input: N$Sl_KDD\ DATA\ SET$

Output: BF // Best subset features

```
Initialize population p; //select randomly from (41) features.
Initialize Condition stop; //number of generation G or stop improving.
F←random (0,1);
CR←random (0,1);
While (! condition stop)
For i ← 1 to N do
Set Target vector $X_i(G)$ ← individual [i] ;
Call ELM $X_i(G)$ to get the accuracy $X_{accuracy}$;
Select three individuals $X_i(G)$ randomly from the Population $X_i(G) \neq X_i(G)$ r ← $r_1, r_2,$ and $r_3$ such that $r_1 \neq r_2 \neq, r_3$;
Set Mutant vector $M_i(G+1)$; // Mutant process
Crossover $M_i(G+1)$ and $X_i(G)$ with Probability of CR to produce trail vector $T_i(G+1)$;
call ELM $(T_i(G+1))$ to get the accuracy $T_{accuracy}$;
fitness function for selection;
If $T_{accuracy} > X_{accuracy}$ then
    $p_{new} \leftarrow T_i(G+1)$;
Else
    $p_{new} \leftarrow X_{i\ G}$ ;
endif
BF ← $p_{new}$
endfor
p ← $p_{new}$;
endwhile
Return BF;
```

4. Performance Evaluation

In this section, the datasets, performance measurements, experimental and results are introduced. The proposed method is implemented in eclipse java programming language on a laptop with Intel® Core™ i5-2430M CPU @ 2.40GHz processor with 4GB of RAM. The proposed method is compared with two recent existing methods named Method1 and Method2. Method1 uses Artificial Neural Network (ANN) technique proposed by Ingre and Yadav [15] while Method2 is a FS+SVM technique that proposed by Pervez and Farid [16].

4.1 Benchmark Datasets.

In recent years, KDD'99 data set has been widely used by researchers to evaluate the intrusion detection system, but it contains some problems such as redundant and irrelevant records which significantly affect the performance of the system. Therefore, a new data named NSL-KDD is suggested by Tavallaee et al [18] that selected from original KDD-99 data but with overcoming its problems. In addition, these data are stable compared to other data. One main challenge in the IDS is if the network data is normal or noisy. The results are achieved without noise are not necessarily the same with data containing noise [19]. Although there are new data which are existed and contain species of attacks like UNSW_NB15 dataset proposed for the Australian Centre for CyberSecurity which is included nine types of attack, NSL dataset is the most reliable in the research and it is the best for evaluation and comparison [20]. Generally, it consists of two datasets for training and testing (125973 and 22544 records respectively) with 41 features for each as shown in Figure 2.

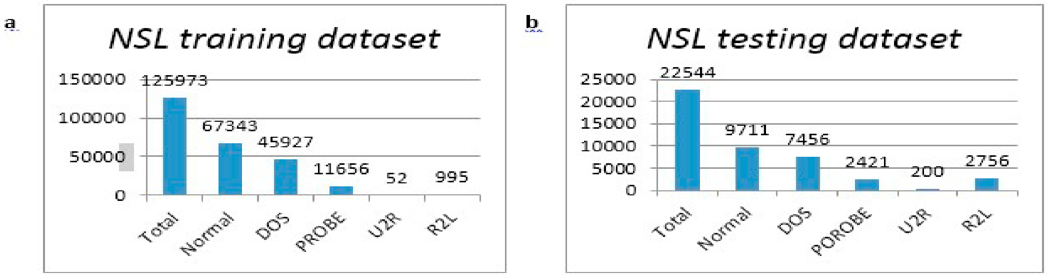

Fig. 2. Datasets. (a) Number of records in training; (b) Number of records in testing

Each record is classified into normal or abnormal, were the abnormal represent 22 attack in the training set and 39 attack in the test set and both belong to four categories [21]:

- **DOS:** resources were reserved by sending more requests to the system to prevent their availability to the user.
- **Probe:** search on the information about the target host by network scanning.
- **User to Root (U2R):** Attempts unauthorized access to the controlling account by guessing the password to manipulate system information.
- **Remote to User (R2U):** access to the system as legal user.

Table.2 Categorization of Attacks for training and testing NSL Dataset (the bold refers to new attack in testing)

DOS	aphache2,back, land, mailbomb , Neptune , pod, processtable, smurf, teardrop, udpstorm.
Probe	Ipsweep ,mscan ,nmap ,saint ,satan,portsweep.
R2L	ftp_write , guess_passwd , imap, multihop, named, phf, sendmail, snmpgetattack, snmpguess, warezmaster,warezclient,spy, worm, xlock, xsnoop.
U2R	buffer_overflow, loadmodule, perl, rootkit, ps, sqlattack, xterm, httptunnel.

Also, all features in NSL dataset are Numeric and only three features are symbolic which are 2 ,3, 4. Therefore, it is required to represent them in the preprocessing stage to prepare them for classification algorithms [21].

4.2 Performance Measurements

There are standard performance measurements used for evaluation. In this paper, the following measurements are used after training the best feature:

- Accuracy: is the proportion of the total number of the correct predictions (attack & normal) to the actual data set size.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

- Detection Rate or true positive rate: is the proportion of correctly predicted attack cases to the actual size of the attack class.

$$Detection\ Rate = \frac{TP}{TP + FN} \quad (5)$$

- Precision: is the proportion of attack cases that were correctly predicted relative to the predicted size of the attack class.

$$Precision = \frac{TP}{\mathrm{TP} + \mathrm{FP}} \quad (6)$$

- F-score: scores the balance between precision and detection rate.

$$F_{score} = \frac{2 * Precision * Detection\ Rate}{Precision + Detection\ Rate} \quad (7)$$

- False Alarm Rate: is the proportion of incorrectly predicted normal (classify as attack) cases to the actual size of the normal class.

$$False\ Alarm\ Rate = \frac{FP}{\mathrm{FP} + \mathrm{TN}} \quad (8)$$

4.3 Experimental Results and Discussion

This section introduced the main results which are conducted after applying the proposed method. Table 3 shows the classification of NSL-testing data with 41 features to choose the number of nodes in the hidden layer of the ELM classifier. The proposed method is employed the full NSL training and testing to select the optimal number of nodes for the hidden layer. The results explain that with 25 nodes in the hidden layer of the ELM classifier can give better accuracy and results.

Table.3 choose number of nodes

Number of nodes	Accuracy
15	73.97
20	73.98
25	**76.44**
30	73.76
35	73.93

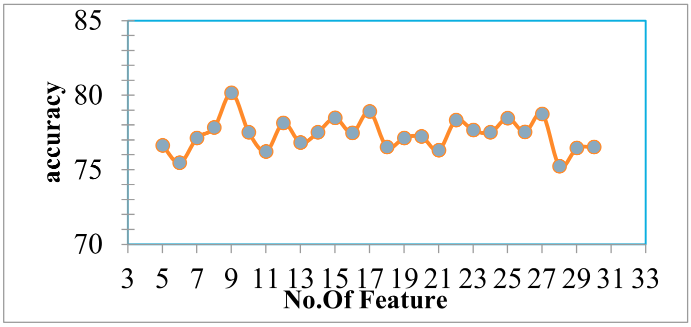

Fig. 3. The number of features vs accuracy for five classes classification.

Moreover, the accuracy of proposed method based on binary classification is achieved, and comparison with another Method2 [16]. Table 4 shows that the proposed method can achieve a better performance through obtaining 87.53% of accuracy with only 5 features in comparison with Method2 that obtained 82.68 % of accuracy with only 14 features.

Table.4 Binary classification using NSL_KDD data

No. of Features	**Mthod2 [16]** **on testing data (%)**	**Proposed Method** **on testing data (%)**
3	78.85	81.91
4	80.03	83.81
5	80.53	**87.53**
6	81.03	86.92
9	81.38	85.26
14	**82.68**	84.57

In Table 5, the results demonstrated the best accuracy in both binary and five classes classification compared with Method1 [15]. The proposed method was achieved 80.15 % of accuracy with 9 features in five class and 87.53 % with only 5 features in binary class. Method1introduces an accuracy reached to 76.3% and 81.2% with 29 features in five and binary respectively. The most important point is that the proposed method achieves a higher accuracy and less time and false alarm rate compared with the 41 features.

Table. 5 Accuracy in binary and five classification.

Technique	**No. of Features**	**Binary classes**	**Five classes**
Proposed Method	**9**	85.26	**80.15 %**
Proposed Method	**5**	**87.53 %**	76.63
Mthod1 [15]	**29**	**81.2 %**	**76.3 %**

The results in Table 6 illustrates the improvement in the performance after applied the proposed method for feature selection. The Table 7 show performance measures.

Table. 6 Improvement in the performance after applied the proposed method

Proposed method	**accuracy**	**Time**	**False alarm rate**
41 feature five class	76.44	0.265 ms	0.3
9 feature five class	80.15	0.141 ms	0.2
5 feature binary class	87.53	0.125 ms	0.05

Table. 7 performance measure

	Five class 41 features					**Five class 9 features**					**Binary class 5 features**	
	Normal	**Dos**	**Probe**	**R2L**	**U2R**	**Normal**	**Dos**	**Probe**	**R2L**	**U2R**	**Attack**	**Normal**
DR	98.14	79.55	73.07	0.0	0.0	96.79	91.50	62.74	11.91	0.0	82.12	52.27
Precision	68.15	94.90	76.65	0.0	0.0	77.58	81.18	89.78	99.09	0.0	95.32	80.03
F_score	80.44	86.55	74.81	0.0	0.0	86.13	86.03	73.86	21.26	0.0	88.23	63.24

Although the ELM introduced a low accuracy that reached 76.44% with all features as shown in Table 8, but the performance is improved with the lowest features that reached to 80.15% with nine features. This indicates that the proposed method is efficient in choosing the best features. Table 8 gives the WEKA3.9.3 implementation using some classifiers such as SVM, Random forest, NB, NN, MLP, and C4.5 to compare the performance using 41 features. Figure 4 shown how the ELM obtain on the highest accuracy with proposed feature selection and comparison the results with other classifiers.

Table. 8 Accuracy some classification using Weka3.9.9

Classifier	**Full features 41**
Naive Bayes	52.66
Random Forest	72.26
SVM	75.81
C4.5	73.00
NN	76.65
MLP	72.66
ELM	**76.44**

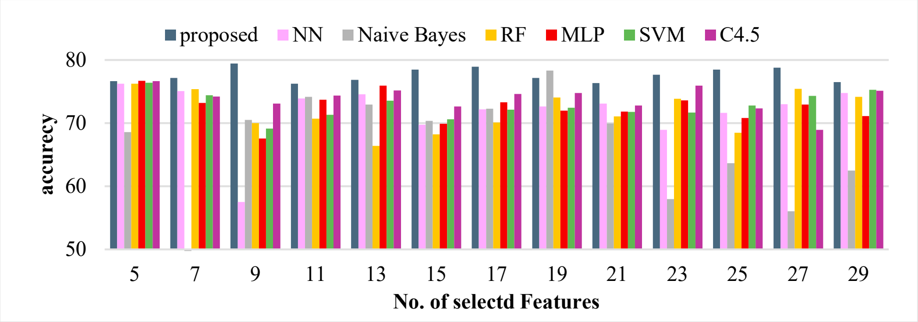

Fig. 4. Accuracy vs No. of selected Features for some classifiers using WEKA3.9.3.

5. Conclusions and Future work

The main goal of feature selection to reduce datasets with improve the performance of the system. This paper presented feature selection-based DE and we have applied ELM on several features and the experimental shown that this technique is able to achieve high accuracy of 80.15 % with nine features selected in five classes classification. While with 41 features achieved accuracy of 76.44% accuracy. in addition, this reduction in the number of features led to reducing both the training and testing times. In future, we plan to generate connections from live networks to test the model and use a complex classifier to obtain higher detection can detect the U2R attack which represents one of the challenges in IDS because the behavior of this type is very near to the normal, therefore, it is very difficult to detect it.

References

[1] Ganapathy, Sannasi, Kanagasabai Kulothungan, Sannasy Muthurajkumar, Muthusamy Vijayalakshmi, Palanichamy Yogesh, and Arputharaj Kannan. (2013)"Intelligent feature selection and classification techniques for intrusion detection in networks: a survey. " EURASIP Journal on Wireless Communications and Networking **271**: 1–16.

[2] A. Sadek, Rowayda, M. Sami Soliman, and Hagar S. Elsayed. (2013)"Effective Anomaly Intrusion Detection System based on Neural Network with Indicator Variable and Rough set Reduction." International Journal of Computer Science Issues **10 (6)**: 227–233.

[3] Lee, Wenke, Salvatore J.Stolfo, and Kui W. Mok. (2000) "Adaptive intrusion detection: A data mining approach," Artificial Intelligence Review **14 (6)**: 533–567.

[4] Ait Tchakoucht , Taha and Mostafa Ezziyyani. (2018)"Building a fast intrusion detection system for high-speed-networks: Probe and dos attacks detection. " Procedia Computer Science **127**: 521–530.

[5] Akhtar Khan, Javed, and Nitesh Jain. (2016)"A Survey on Intrusion Detection Systems and Classification Techniques. " International Journal of Scientific Research in Science, Engineering and Technology **2 (5)**: 202–208.

[6] Acharya, Neha, and Shailendra Singh. (2018)"An IWD-based feature selection method for intrusion detection system." Soft Comput. **22 (13)**: 4407– 4416.

[7] Zuech, Richard, and Taghi M. Khoshgoftaar. (2015) "A survey on feature selection for intrusion detection. " International Conference on Reliability and Qualit 150–155.

[8] Vinutha, H. P, and B. Poornima. (2018)" An ensemble classifier approach on different feature selection methods for intrusion detection. " Information systems design and intelligent applications 443–451.

[9] Subba, Basant, Santosh Biswas, and Sushanta Karmakar. (2017) "Enhancing performance of anomaly based intrusion detection systems through dimensionality reduction using principal component analysis." International Conference on Advanced Networks and Telecommunications Systems IEEE .

[10] Lee, Jinlee, Dooho Park, and Changhoon. Lee. (2017)"Feature selection algorithm for intrusions detection system using sequential forward search and random forest classifier." Transactions On Internet And Information Systems **11 (10)**: 5132–5148.

[11] Sayed A.Aziz, Amira, Sanaa EL-Ola Hanafi, and Aboul Ella Hassanien. (2017) "Comparison of classification techniques applied for network intrusion detection and classification." Journal of Applied Logic **24**: 109–118.

[12] Nskh, Praneeth, Naveen Varma M, and Roshan Ramakrishna Naik. (2016)"Principle Component Analysis based Intrusion Detection System Using Support Vector Machine." International Conference On Recent Trends In Electronics Information Communication Technology 1344–1350.

[13] P Gupta, Govind, and Manish Kulariya. (2016) "A Framework for Fast and Efficient Cyber Security Network Intrusion Detection Using Apache Spark." International Conference on Advances in Computing & Communications **93 (7)**: 824–831.

[14] Gaikwad, D. P, and Ravindra C. Thool. (2015)"Intrusion detection system using Bagging with Partial Decision Tree base classifier." Procedia Computer Science **49(1)**: 92–98.

[15] Ingre, Bhupendra, and Anamika Yadav. (2015)"Performance analysis of NSL-KDD dataset using ANN. " Communication. Engneering. System IEEE pp. 92–96.

[16] Pervez, Muhammad Shkil, and Dewan Md. Farid. (2014) "Feature Selection and Intrusion classification in NSL-KDD Cup 99 Dataset Employing SVMs." International Conference on Software, Knowledge, Information Management and Applications,Dhaka,Bangladesh pp. 1.

[17] Al-Jarrah, O. Y, A. Siddiqui, M. Elsalamouny, P. D. Yoo, S. Muhaidat, and K. Kim. (2014) "Machine-learning-based feature selection techniques for large-scale network intrusion detection." International Conference on Distributed Computing Systems Workshops **30**: 177–181.

[18] Tavallaee, Mahbod, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani. (2009) "'A detailed analysis of the KDD CUP 99 data set,' in Computational Intelligence for Security and Defense Applications." Symposium on Computational Intelligence in Security and Defense Applications pp. 1–6.

[19] Aggarwal, Preeti, and Sudhir Kumar Sharma. (2015) "Analysis of KDD Dataset Attributes - Class wise for Intrusion Detection." International Conference on Recent Trends in Computing **57**: 842–851.

[20] J. Miller, Nicholas, and Mehrdad Aliasgari. (2018) "Benchmarks for Evaluating Anomaly Based Intrusion Detection Solutions." International Journal of Network Security & Its Applications **10 (5)**: 01-12.

[21] Paulauskas, Nerijus, and Juozas Auskalnis. (2017) "Analysis of data pre-processing influence on intrusion detection using NSL-KDD dataset." Electrical ,Electronic and Information Sciences,Vilnius, Lithuania pp. 1–5.

[22] NSL-KDD dataset, http://www.unb.ca/research/iscx/dataset/iscx-NSL-KDD-dataset.html.