IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 4, Ver. V (Jul.-Aug. 2016), PP 01-11
www.iosrjournals.org

Variance-Index Based Feature Selection Algorithm for Network Intrusion Detection

B. Basaveswara Rao[1], K.Swathi[2]
[1] *Computer Center, Acharya Nagarjuna University, Guntur,*
[2] *Department of Computer Science and Engineering,Acharya Nagarjuna University, Guntur,*

Abstract: *This paper presents a feature selection methodology in the domain of Network Intrusion Detection System (NIDS). An Unsupervised Variance Indexed Feature Selection Algorithm (VIFS) is proposed and demonstrated by considering a benchmark dataset, NSL KDD. A Fast KNN classification algorithm Indexed Partial Distance Search k Nearest Neighbor(IKPDS) is applied for finding the classification accuracy. The algorithm selects a subset of features based on fitness threshold value with tolerable loss in classification accuracy but gain in computational time. In a trade of NIDS classification accuracy and computational time are two factors to decide the performance of NIDS. Feature subset length is one of the parameter to influence the computational time. So in order to evaluate the fitness value to consider the classification accuracy and feature subset length in this study. Two parameters α and β are related to the presence of classification accuracy quality and feature subset length. A numerical illustration presented for 0.5 ≤ α ≤ 9, where α =[0,1] and β =1- α. Then identified three feature selection scenarios. Finally the merits and demerits of these three scenarios are discussed. This VIFS fulfills the gain in computational time objective with a tolerable loss in classification accuracy.*

Keywords: *Feature Selection, Intrusion Detection, NSL-KDD data set.*

I. Introduction

Security is a complex and time critical activity in the field of computer networks. With the growing needs of the World Wide Web in the society, network security is also need to be strengthened so that it can identify various types of vulnerabilities. Network Intrusion Detection System (NIDS) is a popular defiance mechanism in field of network security that provides a strong and in-time detection of various types of attacks with less cost. One of the advantages of NIDs is the availability of huge collection of network data on which machine learning algorithms can be applied to detect attacks. At the same time with the availability of such a huge data, the computational time will be increased drastically. To overcome this problem data reduction techniques are helpful to survive from the bundles of data that available. Another problem with the bulk data sets is it may contain irrelevant information that misleads the learning algorithms. Feature selection is a kind of data reduction technique which helps to select relevant information from the data.

Feature selection is a process of selecting useful features from the available. Lot of work is going on feature selection from past few decades especially for large data sets like KDD Cup'99[9][10]. Feature selection is helpful not only to reduce the computational time by minimizing the data size but also increases the classification accuracies by avoiding irrelevant features that mislead the machine learning algorithms[10] by addressing curse of dimensionality.

Classification is one of the machine learning techniques that aim to learn a model that maps a multi-valued input into a single valued categorical output (decision). Thus, classification algorithms can be used to predict the output of an unseen input. One of the popular or widely used classifier is k- Nearest Neighbor (kNN) Algorithms which yield high accuracy however it is resource hungry and derives high computational time. IKPDS[1][2] is an improved over the kNN classifier that addresses the computational time of the kNN classifier based on partial distance algorithm and variance indexing algorithm.

The fundamental objective of feature selection is to preserve discernibility ability for knowledge synthesis. Feature or derived features with inherent high heterogeneity is expected to possess these characteristics, thus the classical PCA is in place. The variability of a feature is of the statistical measure in the heterogeneity. This paper attempts to explore variance indexing methods developed in kNN classifiers [1][2].The proposed VIFS is developed and discussed in this paper is a iterative and feature eliminative approach. For each iteration applying IKPDS[1][2] for finding accuracy of the classification and eliminate least variance feature. A fitness value is finding for each iteration and this iterative process repeated upto desired fitness value. Many of the feature selection algorithms are class label dependent algorithms and also more computational complexity, computational time. In network intrusion detection the computational time is very important for implementing detection and defense mechanisms. The objectives of the present work is as follows.

i) To develop a class label independent feature selection algorithm for easy implementation.
ii) To minimize the computational time with an acceptable loss of accuracy for a given fitness threshold value. The different feature subsets are drawn from original data set based on threshold value.
iii) The fitness values are evaluated quality of the classification accuracy and the number of features selected. Then identify fitness threshold value for stopping criteria of the feature elimination process.

The remaining paper is organized as follows: the section II and III provide a brief review on this research area and an over view of NSL-KDD data set respectively. The proposed methodology developed and demonstrated in section IV. Experimental results and its analysis are compiled in section V with concluding remarks and feature work.

II. Related Research

To increase the robustness, and accuracy of IDS system S. Chebrolu et al.[3]has proposed ensemble classifier based approach. They used ensemble classifiers for better accuracy for each category of attack pattern. An ensemble method constructs a linear combination of some fitting method, rather than using a single fit of the model. S.Chebrolu [3] proposed a CART-BN approach that increases the efficiency as well as the detection rates.

Anazida Zainal et al. [5] has proposed a Feature selection based on 2-tier structure. A wrapper approach was introduced in which both Rough sets and Particle Swam Optimization techniques are adopted for better representation of data. This method suggested 6 features out of 41 features of KDD cup data set and got 93.408% of accuracy.

Iftikhar Ahmad et.al. [9] proposed an optimized intrusion detection mechanism using soft computing techniques. In this paper they have used PCA, GA and SVM for feature selection and optimized feature selection and classification algorithms respectively. This model has achieved the classification accuracy of 99.6% from 22 feature component elements.

Shafigh Parsazad et al. [10] has proposed a fast feature selection method to eliminate features that has no helpful information. The model has eliminated redundant features from the total set. The method was compared with other models that adopts Correlation Coefficient, Least Square Regression Error and Maximal Information Compression Index. After that they have recommended 10, 20, and 30 number of features by each of these algorithms in two popular classifiers including: Bayes and KNN classifier to measure the quality of the recommendations.

Yinhuiet. al[11] has proposed an intrusion detection method based on Support Vector Machines. The model combines ant colony and support vector machine algorithms and gradually eliminates each feature and finally suggested 19 critical features. The accuracy of this model is about 98.6249%.

Amin Dastanpour et al.[12] has implemented a genetic algorithm (GA) with Support Vector Machine (SVM) classification method for feature selection, and applied Forward Feature Selection Algorithm (FFSA) and Linear Correlation Feature Selection (LCFS) in detecting different types of network attacks. According to this paper, for effective detection of attacks FFSA requires 31 features whereas for GA with SVM and for LCFS require only 21 features. They have achieved about 99% of detection rate.

H. F. Eid etal. [13] has proposed a linear correlation-based feature selection method for building NID model. The proposed method introduced a way of analyzing feature redundancy. The model has two layers, where the first layer selects a feature subset based on the analysis of Pearson correlation coefficients between the features. While, at the second layer a new set of features is selected from within the first layer features subset, by analyzing the Pearson correlation coefficients between the selected features and the classes. The model has implemented on NSLKDD dataset. This method achieves an accuracy of 99.1%, and the subset of features selected are of 17 features.

Zhao et al. [16] has proposed a two-stage feature selection algorithm. This Method combines Information Gain filter approach and Binary Particle Swarm Optimization wrapper approach and tested on Foreign Fibers data set.

The Table I summarized various methods in a chronological order, the datasets used by the respective authors for demonstrating their methodology is provided in column 2. The methods that are integrated in their methodologies of feature selection as well as classification are indicated in column 3, column4 and 5 indicates the number of features selected out of 41 features of the data sets and the corresponding classification accuracy.

Table I: The summarization of feature selection and classification methods of some of the authors.

Author	Data set used	Algorithms adopted for feature selection/ classification	# of features selected	Accuracy in %
C. H. Tsang and S. Kwong.[4] (2005)	KDD cup99	Multi Agent Approach		92.23
S.Chebrolu [3] (2005)	NSL KDD	Ensemble method	12	95.86
Zainal[5] (2007)	KDD Cup99	Rough-PSO	6	93.4
Iftikhar Ahmad [9] (2011)	NSL KDD	PCA, GA, SVM	22	99.6
ShafighParsazad [10] (2012)	10% KDD	KNN, Bayes networks	10,20,30	Upto 98.14
Yinhui[11] 2012	KDD Cup 99	SVM and Ant Colony	19	98.6249
Amin Dastanpour[12] (2013)	NSL KDD	GA, SVM	22	99
H. F. Eid [13] (2013)	NSL KDD	Correlation coefficient	17	99.1
Pervez et. al [15] (2014)	NSL KDD	SVM	36, 29, 17	99
Zhao [16] 2015	Foreign fiber	BPSO	34-42 out of 75	91.48

III. Network Intrusion Data Set

In this paper the NSL-KDD dataset [7] is used as a benchmark dataset because for avoiding redundancy, whereas in DARPA KDDCUP'99 dataset have higher redundancy. It contains seven weeks of training data and two weeks of test data. KDD dataset is widely used as a benchmark dataset for offline network traffic, which helps the researchers to test and implement their algorithms [6][7]. The NSL-KDD dataset a modified version of KDD Cup'99 data set. The KDD Cup'99 data set contains 41 features. As class labels are provided, this data set is widely used for classification algorithms. Each sample is labeled as either normal or attack. Denial of Service (DOS), Probe, U2R and R2L are the categories of attacks available [6]. The description of the dataset is given in Table II[7].

TABLE II Profile of the NSL-KDD Dataset

NSL-KDD dataset	DoS	Probe	R2L	U2R	Normal	Total records
KDDTrain+	45927	11656	995	52	67343	125973
20%KDDTraining+	9234	2289	209	11	13449	25192
KDDTest+	7458	2421	2554	400	9711	22544
KDDTest-21	4342	2402	2554	400	2152	11850

IV. Methodology

The proposed approach has undergone into two phases as shown in Figure 1, they are i) preprocessing, ii) feature selection. In the preprocessing data transformation and normalization are carried out inorder to avoid feature influences on data values. For transformation[1] numeric labels are assigned corresponding to all categorical features, for normalization mean-scale normalization [1][8] is used as proposed in the reference[1]. The feature selection phase contains ranking, indexing, reordering and fitness value evaluation as sub phases. In ranking sub phase, the 41 features are ranked based on the variance of each feature. Features are indexed in the descending order of their ranks. In the reordering phase the total data set is reordered based on variance indexed features and stored in annulated file. After data reordering phase, IKPDS algorithm is applied on the data for calculating accuracies. This process is carried out by eliminating the least variance feature one at a time. The IKPDS algorithm is implemented in a 10 fold cross validation, with k value 3.

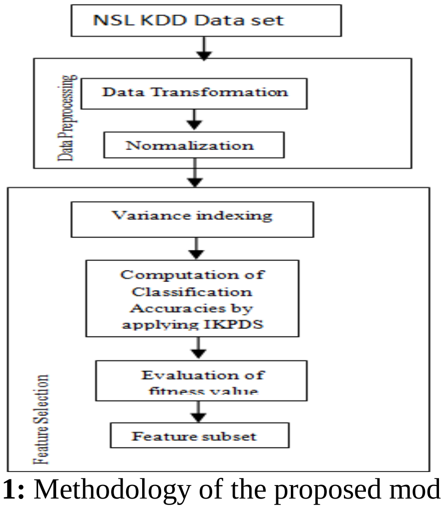


Fig 1: Methodology of the proposed model

After implementing the IKPDS algorithm the classification accuracies VA[], computed with the below algorithm based on these accuracies. To finding the fitness value, the methodology is adopted based on [11] for the selection of feature subset. To derive fitness value is based on two values, they are accuracy and number of features that are not eliminated in the iterations i.e., length of the feature subset. The iteration process repeated upto 41 because the original set contains 41 features but the elimination process of the feature starts with second iteration with least variance feature. The fitness value FV_i is defined for this experiment is as :

$$FV_i = \alpha \times (VA[i]) + \beta \times (41 - R)/41$$

Where $i = 1 \, . \, . \, 41$ and R is the length of feature subset $1 \leq R \leq 41$. The parameters α and β are related to the presence of classification accuracies quality and feature subset length. α =[0,1] and β =1- α.

4.1 Indexing:

Variance is a statistical quantitative measure for identifying the variability in the feature vector. It is class label independent. The variance (v_i)for each feature vector i is calculated for 1 < i < 41 based on the formula:

$v_i=\frac{\sum(x_{ij}-\overline{x_i})^2}{n-1}$where$\overline{x_i}$ is the mean of the feature x_i, n is the total number of samples in the data set.

The list of ordered features based on variance indexing is presented in table III. The features are ordered in descending order of their variance measures.

Table III: Variance indexing of features

S.No	Feature No	Variance of Feature	Name of the feature
1	12	0.2391308319400625	logged_in
2	34	0.2015555311823547	dst_host_same_srv_rate
3	26	0.19982911404355816	srv_serror_rate
4	25	0.19932262449653118	serror_rate
5	39	0.19862096798511014	dst_host_srv_serror_rate
6	38	0.1978328514154712	dst_host_serror_rate
7	29	0.1932682611515425	same_srv_rate
8	33	0.18871017934048365	dst_host_srv_count
9	32	0.15140745736959718	dst_host_count
10	4	0.1245954207011661	Flag
11	28	0.10474752819537658	srv_rerror_rate
12	27	0.10267892295807934	rerror_rate
13	41	0.10205430215013638	dst_host_srv_rerror_rate
14	36	0.09547922657883537	dst_host_same_src_port_rate
15	40	0.09397747507068466	dst_host_rerror_rate
16	2	0.07986449260243712	protocol_type
17	31	0.06751188775383973	srv_diff_host_rate
18	3	0.05254786262718669	Service
19	23	0.05041272635249068	Count
20	35	0.03569144647918471	dst_host_diff_srv_rate
21	30	0.03251328555520514	diff_srv_rate
22	24	0.020265712911144172	srv_count
23	37	0.01267061016931209	dst_host_srv_diff_host_rate
24	22	0.009333941746052109	is_guest_login
25	8	0.007138405929330912	wrong_fragment
26	1	0.003684993348840359	Duration
27	14	0.001339768177164742	lroot_shell
28	10	7.761043760394509E-4	Hot
29	15	5.097295944029559E-4	lsu_attempted
30	7	1.984174151355603E-4	Land
31	17	1.267071698598686E-4	lnum_file_creations
32	18	1.230006173655714E-4	lnum_shells
33	19	1.212545732754819E-4	lnum_access_files
34	11	8.186318783075476E-5	num_failed_logins
35	9	2.294976357467773E-5	Urgent
36	5	1.813423233316109E-5	src_bytes
37	16	1.065408456486432E-5	lnum_root
38	13	1.023041324200907E-5	lnum_compromised
39	6	9.46463427564252E-6	dst_bytes
40	21	7.938208981285125E-6	is_host_login
41	20	0.0	lnum_outbound_cmds

From the above table logedin feature has highest variance (0.239) and lnum_outbound_cmds feature has least variance (0). Based on these variance descending feature vectors, to implement the proposed VIFS method, the following two algorithms are presented.

4.2. Algorithm 1: For finding accuracies based on variance indexed features with IKPDS starting from 41 features and by eliminating the least variance feature at a time.

Input: A = {A_1, A_2, A_3, . . . A_{41}} is a set of all features after pre-processing phase NSL-KDD data set D. Where A_i= {a_{i1}, a_{i2}, a_{i3}, ….,a_{in}} is an i^{th} feature vector where n is the number of tuples in D and a_{ij} is the value of i^{th} attribute at j^{th} tuple.

Output: FN[] 41 variance indexed feature number, D1a set of 41 features based on Variance index reorder and the accuracy vector VA[] and classification time vector VT[] for variance indexing and for fitness value FV[].

```
//Indexing and reordering:
Step 1: /* Initialization
        Declare Variance Vector v[i]
        Declare Feature Vector FN[i]
end
Step 2:
        foreach Ai ∈ A
            Find variance of Ai store into v[i]
            Feature number i in FN[i]
        end for
end

Step 3:
    foreach i=1 to 41
        foreach j=i+1 to 41
            If v[j]>v[j+1] then
                1. t=v[j]; v[j]=v[j+1]; v[j+1]=t;
                2. t=FN[j]; FN[j]=FN[j+1]; FN[j+1]=t;
            end if
        end for
    end for
end
Step 4:
        Reorder the dataset D into D1 in variance
        index order of this feature
End
Step 5:
        Calculate the classification accuracy for D1
End
Step 6:

        foreach i=41 to 1
            1. apply IKPDS
            2. Store the accuracy in VA[i]
            3. Store classification time in VT[i]
        end for
End
```

4.3 Algorithm 2: To identify Feature Selection for given fitness threshold value.

Input: Accuracy vectors VA[], α, β and Maximum threshold value MaxFV, D_1,and Feature Numbers in variance indexed order FN[].

Output: Subset of 41 features that are selected SA[];

```
Step 1 :
        Set R to 41
        foreach i=41 to 1
            /* calculate
            FV[i]=Alpha*(VA[i])+beta*(41-R)/41
            R=R-1
        end for
Step 2 :

        foreach  i=41 to 1
            if FV[i]>=MaxFV then
                  for j= I to 1
                      SA[j]=Fn[j]
                  end for
            end if
        end for
end
```

V. Result Analysis

This model is developed in windows 7 operating system and Java 1.6 on Intel core i5 processor, with 4 GB RAM. In order to investigate the performance of VIFS a fitness values based procedure is developed and executed. The experimental results about classification accuracy, computational time and fitness values are presented in the table IV and the following graphs, when implementing the algorithms 1 and 2.

Table IV: Accuracy, computational time, fitness value by applying IKPDS with VIFS, eliminated feature number and number of selected features where α= 0.8 and α= 0.9.

Round	Eliminated feature number	number of features Selected	Classification Accuracy	Computational time in min	Fitness Values α= 0.8	Fitness Values α= 0.9
1	--	41	0.9965	8.614	0.7972	0.8968
2	20	40	0.9965	8.614	0.8021	0.8993
3	21	39	0.9965	8.243	0.807	0.9017
4	6	38	0.9965	8.255	0.8118	0.9041
5	13	37	0.9965	8.433	0.8167	0.9066
6	16	36	0.9965	8.278	0.8216	0.9090
7	5	35	0.9965	8.244	0.8265	0.9114
8	9	34	0.9964	8.256	0.8313	0.9139
9	11	33	0.9964	8.252	0.8362	0.9163
10	19	32	0.9964	8.235	0.8411	0.9187
11	18	31	0.9964	8.246	0.8459	0.9211
12	17	30	0.9964	8.242	0.8508	0.9236
13	7	29	0.9964	8.246	0.8557	0.9260
14	15	28	0.9964	8.245	0.8606	0.9284
15	10	27	0.9951	8.231	0.8644	0.9297
16	14	26	0.9951	8.222	0.8693	0.9322
17	1	25	0.9950	8.199	0.8741	0.9346
18	8	24	0.9949	8.200	0.8789	0.9369
19	22	23	0.9949	8.187	0.8838	0.9393
20	37	22	0.9948	7.874	0.8886	0.9417
21	24	21	0.9949	7.618	0.8935	0.9442

22	30	20	0.9950	7.551	0.8984	0.9467
23	35	19	0.9945	7.110	0.903	0.9487
24	23	18	0.9927	5.438	0.9064	0.9495
25	3	17	0.9899	4.527	0.909	0.9494
26	31	16	0.9894	3.993	0.9135	0.9514
27	2	15	0.9837	3.952	0.9138	0.9488
28	40	14	0.9813	3.855	0.9168	0.9490
29	36	13	0.9665	3.610	0.9098	0.9382
30	41	12	0.9638	3.461	0.9126	0.9382
31	27	11	0.9633	3.429	0.917	0.9402
32	28	10	0.9623	3.383	0.9211	0.9417
33	4	9	0.9428	3.289	0.9104	0.9266
34	32	8	0.9279	2.750	0.9033	0.9156
35	33	7	0.8892	2.015	0.8773	0.8832
36	29	6	0.8641	1.638	0.862	0.8630
37	38	5	0.8579	1.367	0.862	0.8599
38	39	4	0.8549	1.251	0.8645	0.8597
39	25	3	0.8541	1.169	0.8687	0.8613
40	26	2	0.8572	1.099	0.876	0.8666
41	34	1	0.8294	1.077	0.8587	0.8440

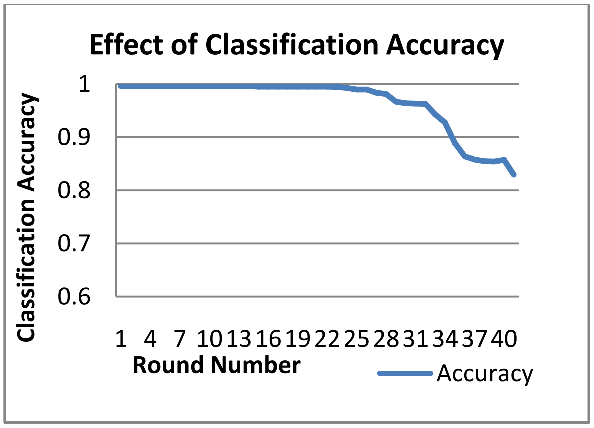


Figure 2: Effect of the Classification accuracy when applying IKPDS with VIFS.

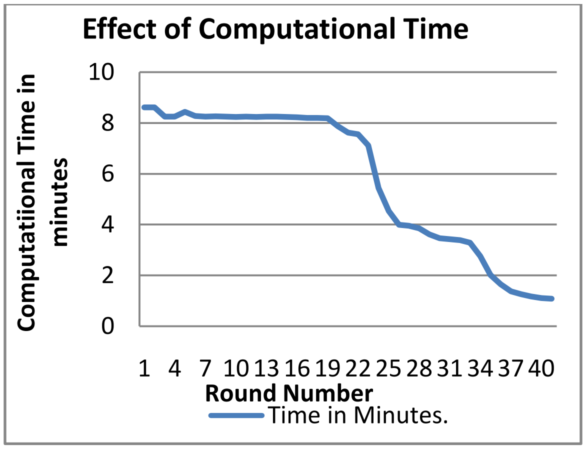


Figure 3: Effect of the Computational time when applying IKPDS with VIFS.

The features that are considered for the VIFS, and the feature that is eliminated at each round are presented in Appendix I for the better view of the elimination process.

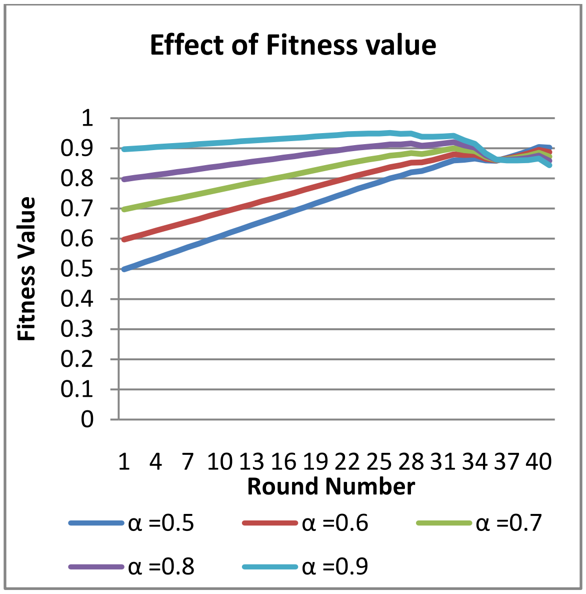


Figure 4: Effect of the fitness value when applying IKPDS with VIFS for 0.5 ≤ α ≤ 0.9 values .

From the table IV, Figure 2, Figure 3 and Figure 4 the following observations are identified to compare with 41 features set at round 1by applying IKPDS.

- The classification accuracy has no variation upto third decimal place until the execution of 14th round with 13 eliminated features, the next variation is observed at the execution of the 24th round with 23 eliminated features. The next major variation is observed upto second decimal place when the execution of the 32ndround with eliminated features 31. The loss of accuracy slowly increases up to execution of the 32nd round then it follow significantly increasing.
- The gain in computational time, after the execution of the 14th round is 0.4 minutes, at 24th round it is 3.2 minutes and at 32ndit is 5.3 minutes. It is significantly increases when the rounds and number of eliminated features increases where the number of selected features decreases.
- When α increases from 0.5 to 0.9 the fitness value also increases. The round number is increasing the fitness values also increases, then it will decreases. For differ values of α the knee of the fitness curve corresponding round number or fitness value is stopping criteria for feature elimination process.
- For α= 0.5 and α=0.6 the highest fitness values are occurred when the execution of the 34th round with number of eliminated features 33 and number of selected features are 8 with a gain in computational time is 5.86minutes.
- When α= 0.7 and α=0.8 the highest fitness values are occurred when the execution of the 32nd round with number of eliminated features 31 and number of selected features 10 with a gain in computational time is 5.23minutes.
- Whereas for α= 0.9 the highest fitness values is occurred when the execution of the 26th round with number of eliminated features 25 and number of selected features 16 with a gain in computational time is 4.25minutes.
- However the loss in accuracy is small quantity between α=0.9 and α=0.8 i.e., 0.027 (0.034 - 0.007).

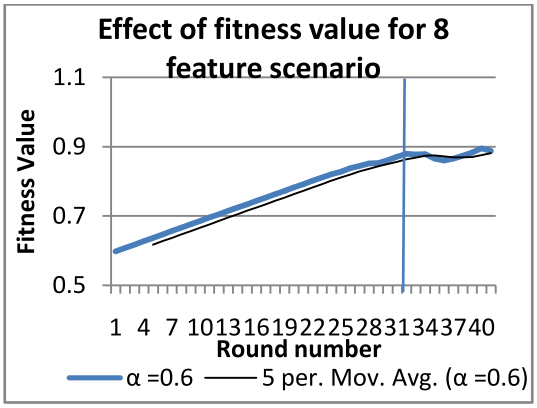
From the above observations the following table is formulated with round numbers, difference in loss of accuracies, and gain in computational times along with fitness values for 0.5 ≤ α ≤ 0.9.

Table V: Detailed explanation of the above observations with numerical values and maximum fitness threshold values showed in parenthesis.

Round	**1**	**14**	**26**	**32**	**34**
Number of features selected	41	28	16	10	8
Loss in classification accuracy	0	0	0.007	0.034	0.069
Gain in computational time	0	0.37	4.25	5.23	5.86

Fitness value	**□=0.5**	0.4982	0.6567	0.7996	0.8592	**0.8664**
	□=0.6	0.5979	0.7246	0.8375	0.8798	**(0.8787)**
	□=0.7	0.6975	0.7961	0.8755	**0.9004**	0.8910
	□=0.8	0.7972	0.8606	0.9135	**(0.9211)**	0.9033
	□=0.9	0.8968	0.9260	**(0.9514)**	0.9417	0.9156

From the table V there are three scenarios that are identified based on the values of the loss of accuracy, gain in computational time and number features selected. They are mainly 8 feature scenario (0.5≤ α ≤0.6), 10 feature scenario (0.7≤ α ≤0.8), and 16 feature scenario (α = 0.9). In 8 feature scenario highest fitness value is occurred at 34th round and maximum fitness threshold value is 0.8787. For 10 feature scenario highest fitness value is occurred at 32nd round and maximum fitness threshold value is 09211. Whereas for 16 feature scenario highest fitness value is occurred at 26th round and maximum fitness threshold value is 0.9514. These maximum fitness threshold values are used in algorithm 2 to get feature subsets for the respective three scenarios. These stopping criteria values are shown in soft braces in table V. The effect of fitness value for 10, 8 and 16 featured scenarios are shown using line graph in figures 5, 6 and 7 respectively.



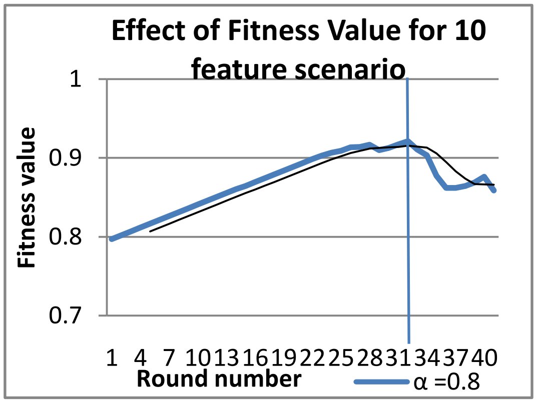


Figure 5: Effect of fitness value along with smoothed 5 points moving average curve when α=0.6.
Figure 6: Effect of fitness value along with smoothed 5 points moving average curve when α=0.8.

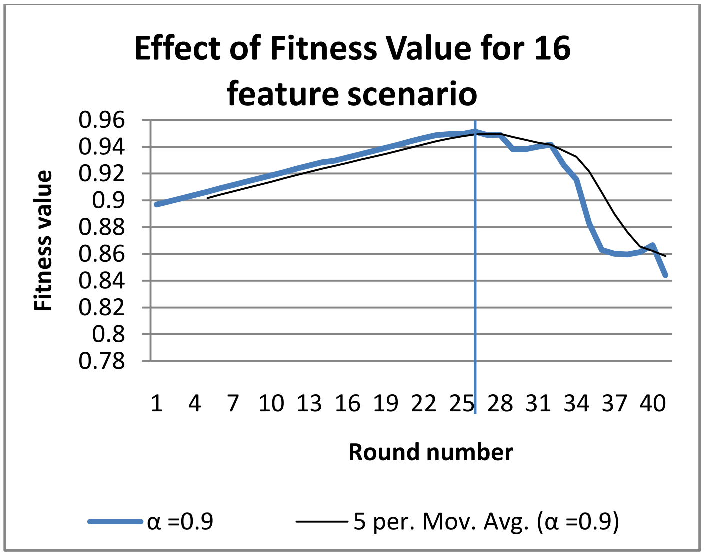


Figure 7: Effect of fitness value along with smoothed 5 points moving average curve when α=0.9.

There is a need to identify the trade of between three scenarios for respective fitness threshold values are implemented for stopping criteria of least variance feature elimination process. To establish the trade between these three scenarios, the loss in accuracy and gain in computational time are used to find the normalized percentages for three different scenarios are calculated and shown in the following Figure 8.

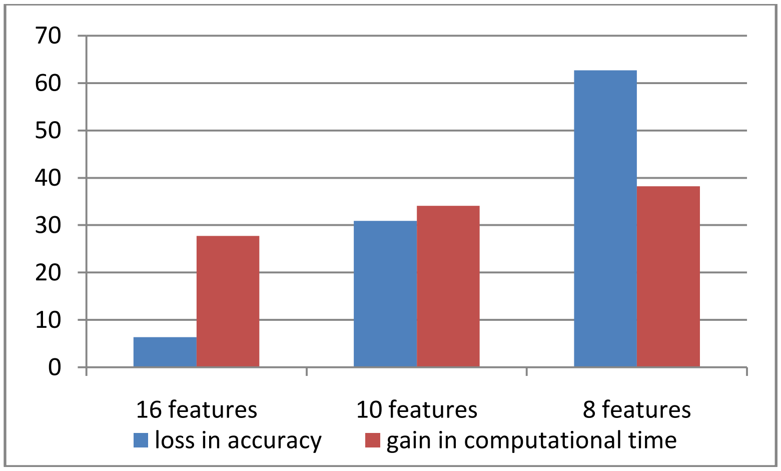


Figure 8: The loss in accuracy and gain in computational time for three different scenarios

In the 8th featured scenario, The loss in accuracy is double when compared with the 10 featured scenario and ten times greater than 16 feature scenario. But the gain in computational time is not very much difference between scenarios when compare to loss of accuracy i.e., the difference between 16 feature scenario and 10 feature scenario is double the difference between 10 feature scenario and 8 feature scenario. From these findings to suggest that the 16 feature scenario is better for those applications that needs high detection rates even though it takes more computational time. For faster detection 10, 8 featured scenarios are preferable even though there is a loss in classification accuracy is high. The following table VI shows the features selected for the given three scenarios.

Table VI: Selected Features for 16, 10 and 8 feature scenarios

	16 featured Scenario	**10 featured scenario**	**8 featured scenario**
Features Selected	protocol_type, Flag, logged_in, serror_rate, srv_serror_rate, rerror_rate, srv_rerror_rate, same_srv_rate, dst_host_count, dst_host_srv_count, dst_host_same_srv_rate, dst_host_same_src_port_rate, dst_host_serror_rate, dst_host_srv_serror_rate, dst_host_rerror_rate, dst_host_srv_rerror_rate	Flag, logged_in, serror_rate, srv_serror_rate, same_srv_rate, dst_host_count, dst_host_srv_count, dst_host_same_srv_rate, dst_host_serror_rate, dst_host_srv_serror_rate	logged_in, serror_rate, srv_serror_rate, same_srv_rate, dst_host_srv_count, dst_host_same_srv_rate, dst_host_serror_rate, dst_host_srv_serror_rate

VI. Conclusion

In this paper an unsupervised least variance feature eliminated feature selection algorithm is proposed and implemented on a benchmark NSLKDD data set for NIDS. To investigate the VIFS performance the classification accuracy is used by applying IKPDS. In this feature selection process a fitness value is evaluated for identifying the stopping criteria of the least variance feature elimination process. Different threshold fitness values are evaluated and classified three scenarios based on quality of accuracy parameter (α) and number of features selected (β). From these experimental results the trade between these three scenarios is evaluated and draw conclusions based on loss of classification accuracy and gain in computational time. For application of NIDS with the importance of attacks detection rate the 16 feature scenario is suggested whereas the importance of the faster detection of attacks, 10 and 8 feature scenarios are suggested. This VIFS is threshold choice base feature subset generation process with compromising in accuracy and gain in computational times.

In future work different supervised/unsupervised feature selection algorithms compared with VIFS on different types of classifications for significance of the VIFS.

References

[1]. B. BasaveswaraRao, K. Swathi, Fast kNN Classifiers for Network Intrusion Detection System, 2015, working paper

[2]. Yu-Long Qiao, Jeng-Shyang Pan, Sheng-He Sun, Improved Partial Distance Search for K Nearest-neighbor Classification, 2004 IEEE.

[3]. S. Chebrolu, A. Abraham, and J.P. Thomas, “Feature Deduction and Ensemble Design of Intrusion Detection Systems.” International Journal of Computers and Security, Vol 24, Issue 4,(June 2005), 295-307.

[4]. C. H. Tsang and S. Kwong.Multi-agent intrusion detection system in industrial network using ant colony clustering approach and unsupervised feature extraction.In IEEE International Conference on Industrial Technology (ICIT ’05), pages 51– 56. IEEE Press, 14-17 Dec. 2005.

[5]. Zainal, Anazida, MohdAizainiMaarof, and SitiMariyamShamsuddin. "Feature selection using rough-DPSO in anomaly intrusion detection."Computational Science and its Applications–ICCSA 2007.Springer Berlin Heidelberg, 2007.512-524.

[6]. MahbodTavallaee, EbrahimBagheri, Wei Lu, and Ali A. Ghorbani, A Detailed Analysis of the KDD CUP 99 Data Set, Proceedings of the 2009 IEEE symposium on computational intelligence in Security and Defense Applications(CISDA 2009)

[7]. “Nsl-kdd data set for network-based intrusion detection systems.” Available on: http://nsl.cs.unb.ca/KDD/NSLKDD.html, March 2009. W. Wang, X. Zhang, S. Gombault, and S. J. Knapskog, “Attribute normalization in network intrusion detection,” in Proceedings of the 10th International Symposium on Pervasive Systems, Algorithms, and Networks (I-SPAN’09), pp. 448–453, IEEE, Kaohsiung City, Taiwan, December 2009.

[8]. Ahmad I, Abdullah AB, Alghamdi AS, Hussain M (2011b). Optimized intrusion detection mechanism using soft computing techniques, Telecommun. Syst., 48(1-2):1-9.

[9]. ShafighParsazad, EhsanSaboori, Amin Allahyar “Fast Feature Reduction in Intrusion Detection Datasets” MIPRO 2012, Pp 1023-1029.

[10]. Li, Yinhui, et al. "An efficient intrusion detection system based on support vector machines and gradually feature removal method." Expert Systems with Applications 39.1 (2012): 424-430.

[11]. A. Dastanpour and R. A. R. Mahmood, "Feature selection based on genetic algorithm and Support Vector machine for intrusion detection system," in The Second International Conference on Informatics Engineering & Information Science (ICIEIS2013), 2013, pp. 169-181.

[12]. H. F. Eid, A. E. Hassanien, T.-h. Kim, and S. Banerjee, "Linear Correlation-Based Feature Selection for Network Intrusion Detection Model," in Advances in Security of Information and Communication Networks, ed: Springer, 2013, pp. 240-248.

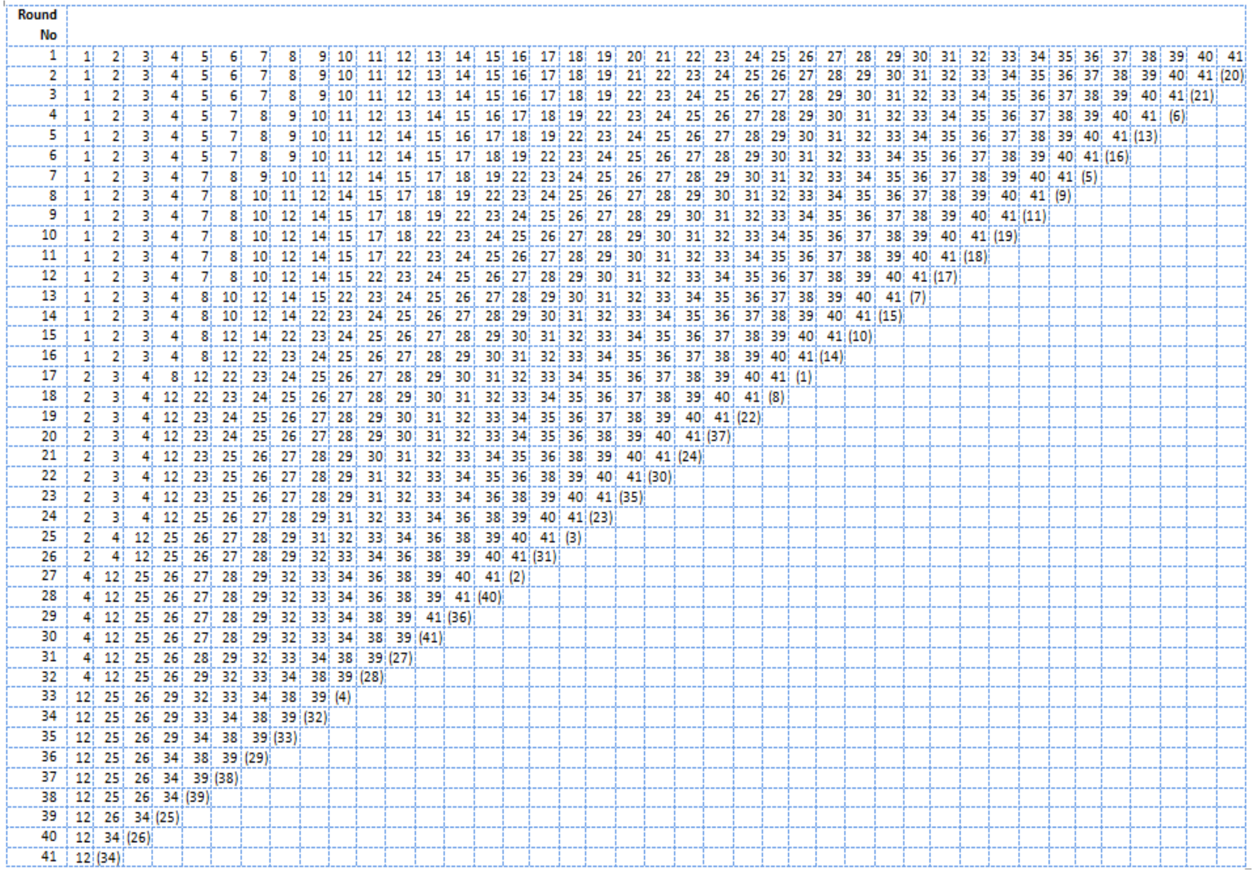
[13]. Hee-suChae, Byung-oh Jo , Sang-Hyun Choi , Twae-kyung Park, “Feature Selection for Intrusion Detection using NSL-KDD” in Recent Advances in Computer Science, 2013.

[14]. Pervez, Muhammad Shakil, and DewanMdFarid. "Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs." Software, Knowledge, Information Management and Applications (SKMA), 2014 8th International Conference on.IEEE, 2014.

[15]. Zhao, Xuehua, et al. "A two-stage feature selection method with its application." Computers & Electrical Engineering 47 (2015): 114-125.

Appendix I

List of feature Numbers after eliminating one feature(showed in braces) for each round



Round No																																										
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	
2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	(20)	
3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	(21)		
4	1	2	3	4	5	7	8	9	10	11	12	13	14	15	16	17	18	19	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	(6)			
5	1	2	3	4	5	7	8	9	10	11	12	14	15	16	17	18	19	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	(13)				
6	1	2	3	4	5	7	8	9	10	11	12	14	15	17	18	19	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	(16)					
7	1	2	3	4	7	8	9	10	11	12	14	15	17	18	19	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	(5)						
8	1	2	3	4	7	8	10	11	12	14	15	17	18	19	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	(9)							
9	1	2	3	4	7	8	10	12	14	15	17	18	19	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	(11)								
10	1	2	3	4	7	8	10	12	14	15	17	18	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	(19)									
11	1	2	3	4	7	8	10	12	14	15	17	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	(18)										
12	1	2	3	4	7	8	10	12	14	15	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	(17)											
13	1	2	3	4	8	10	12	14	15	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	(7)												
14	1	2	3	4	8	10	12	14	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	(15)													
15	1	2	3	4	8	12	14	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	(10)														
16	1	2	3	4	8	12	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	(14)															
17	2	3	4	8	12	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	(1)																
18	2	3	4	12	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	(8)																	
19	2	3	4	12	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	(22)																		
20	2	3	4	12	23	24	25	26	27	28	29	30	31	32	33	34	35	36	38	39	40	41	(37)																			
21	2	3	4	12	23	25	26	27	28	29	30	31	32	33	34	35	36	38	39	40	41	(24)																				
22	2	3	4	12	23	25	26	27	28	29	31	32	33	34	35	36	38	39	40	41	(30)																					
23	2	3	4	12	23	25	26	27	28	29	31	32	33	34	36	38	39	40	41	(35)																						
24	2	3	4	12	25	26	27	28	29	31	32	33	34	36	38	39	40	41	(23)																							
25	2	4	12	25	26	27	28	29	31	32	33	34	36	38	39	40	41	(3)																								
26	2	4	12	25	26	27	28	29	32	33	34	36	38	39	40	41	(31)																									
27	4	12	25	26	27	28	29	32	33	34	36	38	39	40	41	(2)																										
28	4	12	25	26	27	28	29	32	33	34	36	38	39	41	(40)																											
29	4	12	25	26	27	28	29	32	33	34	38	39	41	(36)																												
30	4	12	25	26	27	28	29	32	33	34	38	39	(41)																													
31	4	12	25	26	28	29	32	33	34	38	39	(27)																														
32	4	12	25	26	29	32	33	34	38	39	(28)																															
33	12	25	26	29	32	33	34	38	39	(4)																																
34	12	25	26	29	33	34	38	39	(32)																																	
35	12	25	26	29	34	38	39	(33)																																		
36	12	25	26	34	38	39	(29)																																			
37	12	25	26	34	39	(38)																																				
38	12	25	26	34	(39)																																					
39	12	26	34	(25)																																						
40	12	34	(26)																																							
41	12	(34)																																								