Data Preparation and Reduction Technique in Intrusion Detection Systems: ANOVA-PCA

Mohammed Nasser Mohammed *mo_hd80@yahoo.com*
Faculty of Engineering/ Information Technology
University of Aden
Aden, Yemen

Mussa Mohamed Ahmed *mussa_m7@yahoo.com*
Faculty of Engineering/ Electronics and Communication
University of Aden
Aden, Yemen

Abstract

Intrusion detection system plays a main role in detecting anomaly and suspected behaviors in many organization environments. The detection process involves collecting and analyzing real traffic data which in heavy-loaded networks represents the most challenging aspect in designing efficient IDS.

Collected data should be prepared and reduced to enhance the classification accuracy and computation performance.

In this research, a proposed technique called, ANOVA-PCA, is applied on NSL-KDD dataset of 41 features which are reduced to 10. It is tested and evaluated with three types of supervised classifiers: k-nearest neighbor, decision tree, and random forest. Results are obtained using various performance measures, and they are compared with other feature selection algorithms such as neighbor component analysis (NCA) and ReliefF. Results showed that the proposed method was simple, faster in computation compared with others, and good classification accuracy of 98.9% was achieved.

Keywords: IDS, Supervised Classifiers, NOVA-PCA.

1. INTRODUCTION

With the growing usage of the Internet, it is important to develop effective network intrusion detection systems (NIDS) that identify existing attack patterns and recognize new intrusions. Network Intrusion Detection Systems monitor Internet traffic to detect malicious activities including but not limited to denial of service attacks, network accesses by unauthorized users, attempts to gain additional privileges and port scans. NIDS achieve this goal by inspecting all the incoming packets, outgoing or local traffic to find suspicious patterns [1].

The old and most used dataset in IDS is KDD CUP 99, however this dataset have some defects which are analyzed in [2]. The new version of KDD dataset, NSL-KDD, is publicly available for researchers [3]. Although, the dataset still suffers from some of the problems discussed by McHugh [4] and may not be a perfect representative of existing real networks, because of the lack of public datasets for network-based IDS, we believe it still can be applied as an effective benchmark dataset to help researchers compare different intrusion detection methods [5].

Data preparation is an initial and important preprocessing stage. NSL-KDD dataset is grouped, prepared, and transformed into a suitable and more informative form so that data modeling or classification could be more efficient. The aim is to mine the data and sort out the most valuable parts from the less important ones.

The remaining sections are organized as follows: related works are introduced in section2, in section3, NSL-KDD dataset is described by showing all the features and output labels and their types, data preparation and grouping are discussed in section4. Section5 introduces the proposed method which is used to analyze and reduce the size of the dataset. In section6, experimental results and outcomes from several classification algorithms are discussed and compared with previously related feature selection techniques such as NCA and ReliefF. Section7 includes conclusion and some notes.

2. RELATED WORKS

Most of the researches in IDS focused on reducing the size of the data before feeding it to a classifier which is called a preprocessing stage. It's essential stage to reduce the computation time and enhancing the detection rate while reducing the false positive rate. In 2012 Amrita and P. Ahmed [6] made a survey on latest feature selection methods, including filter and wrapper methods.

D. H. Deshmukh, T. Ghorpade, P. Padiya [7] used a feature selection algorithm called fast correlation based filter to choose a subset from the 41 feature set, they examined three classifier types, Naive Bayes, Hidden Naive Bayes, and NBTREE hybrid classifier. Continuous features were discretized by equal width discretization method, since the three classifiers are best adapted with discrete features. They introduced best accuracy of 94.6% using NBTREE classifier.

B. Angre, A. Yadav [8] used NSL-KDD dataset. Almost zero-value features were removed and Z-score normalization was applied. 29 features out of 41 were selected and applied as input to ANN classifier. They trained and tested ANN with different architectures, different number of neurons in hidden layer.

Y. Bouzida, F. Cuppens, N. Cuppens-Boulahia and S. Gombault [9] used KDD99 dataset. They used PCA technique to project high dimensional feature space into low dimensional space by investing the correlation between the features. As a consequence dataset was reduced. Two experiments were done. One used the full dataset and the other used the transformed one. Discrete features were transformed to numeric values using dummy variables before applying PCA. Results showed noticeable improvement in computation speed using PCA.

G. Meena, R. R. Choudhary [10] compared the classification performance of J48 and Naive byes classifiers. They used KDD99 and NSL-KDD dataset to train and test both classifiers. Better accuracy was obtained from J48 but Naive byes exhibited simplicity and less time-consuming than J48.

K. Ibrahimi, M. Ouaddane [11] used PCA to remove redundant features then applied LDA, linear discriminant analysis, to remove unrelated features, and select the features which maximizes the discrimination between output classes. 90% detection rate was obtained.

Muttaqien and T. Ahmad [12] introduced a hybrid feature selection method, combination of filter and wrapper technique. Filtering was done using gain ratio as a ranking metric followed by forward hill climb search algorithm to find the optimal feature set. They proposed feature transformation method to convert the resulted feature set into one dimensional distance feature. Classification was implemented using KNN classifier with accuracy of 97.42%.

N. Elssied, O. Ibrahim and A. Osman [13] used their proposed technique in feature selection. They used ANOVA to rank the features, and SVM as a classifier.

W. Yang, K. Wang, W. Zuo [14] proposed an algorithm in feature selection. It's based on nearest neighbor feature weighting, it performed better than the state-of-art methods in most cases.

In a previous work [15], a data preparation method was implemented. We used a combination of Sample mean and ANOVA techniques to obtain a small scaled version of data. Different classifiers were examined using the reduced dataset. It showed that there was a beneficial but with a loss of about 5% of accuracy. This paper extended the previous work, improved the accuracy and used a 5-category classification instead of two.

3. DATASET DESCRIPTION

NSL-KDD99 dataset which is an improved version of DARPA KDD CUP99 dataset, evolved in 1998. It's an old dataset and many researchers claimed that it didn't reflect recent real network traffic [16], but there is a survey in 2010 to 2015 showed that there are 142 researches in machine learning and IDS still used KDD99 dataset to compare and evaluate the performance of different classification methods [5]. NSL-KDD has advantages over KDD CUP99 which are stated in [17] and [11].

Actually NSL-KDD data was divided into two main parts; training and testing datasets. The training dataset was used throughout the research. For generalization, testing dataset was used. The training dataset consists of 125973 network flow data, 41 features and one class which is marked as normal or attack. In the train dataset there are 24 different attack types. Dataset for testing contains additional new 14 attacks. In this way, testing is done with unknown attacks as well. Attacks are grouped into four categories, DOS, PROBE, R2L , and U2R. see Table1, Its shows all attack categories, including normal one. Detailed explanation for each category is found in [2] and [18].

Category	**Train Data**	**Train Data 20%**	**Test Data**
NORMAL	67343	13449	9710
DOS	45927	9234	7460
PROBE	11656	2289	2421
R2L	995	209	2885
U2R	52	11	67
Total Observations	125973	25192	22543

TABLE 1: Distribution of Data in NSL-KDD Dataset [18].

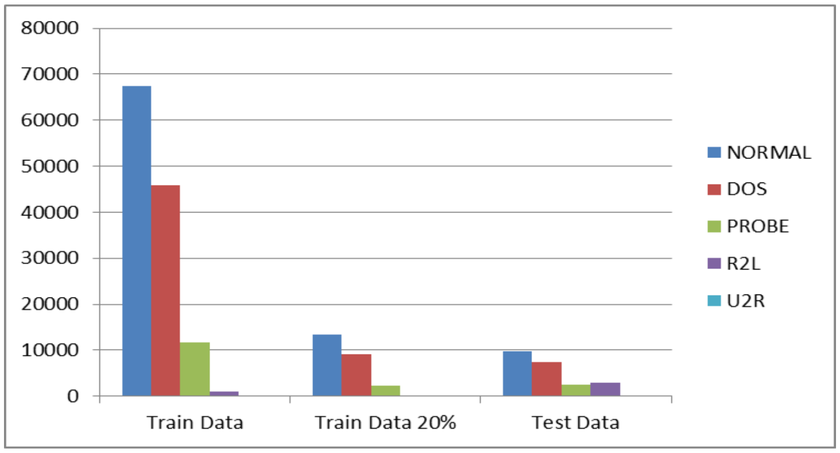


FIGURE 1: Graphical Representation of Distribution of Data [18].

The table shows how the total observations in each dataset are distributed among the output categories. Obviously shown that most of observations are acquired by normal and attacks of DOS and PROBE. The numerical data in Table1 can be represented graphically in Figure1, showing the relative data size for each category in every dataset. R2L and U2R have less number of records so less knowledge will have been acquired in training phase. As a consequence poor classification accuracy was predicted compared to others.

4. DATASET PREPARATION

In NSL_KDD dataset, there are 39 attacks that are categorized into four category groups. Table2 shows the names of the attack class and their corresponding class category group. The observations are labeled by their attack name, and by the use of this table, each observation was mapped by their corresponding category group rather than the name of the specific attack. This made the dataset easier to interpret and study.

NSL-KDD has 41 features which are categorized into:

1) Basic features
2) Content features
3) Traffic Features:
   - Same Service traffic features
   - Same host traffic features

The features are showed and explained in detail in [18]. They are sorted into two types, continuous and discrete features. Table3 stated each feature, their types and range of values. As you see one feature is omitted since it has zero values along all the dataset, and there are 33 continuous features and 7 discrete ones.

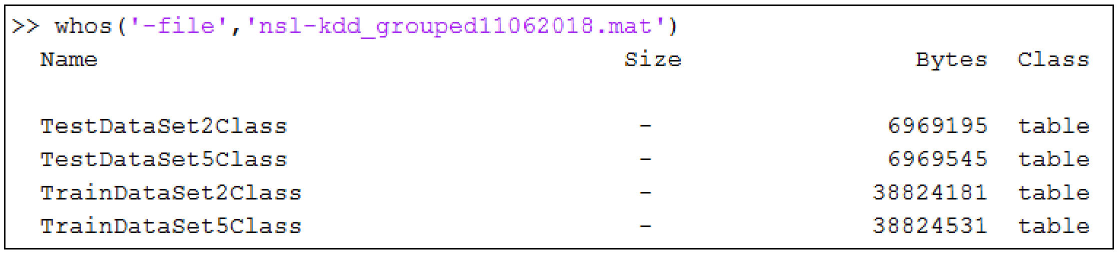
Class	Group	Group_no	Class	Group	Group_no
Satan	Probe	1	buffer_overflow	u2r	3
Saint	Probe	1	Sqlattack	u2r	3
Ipsweep	Probe	1	loadmodule	u2r	3
Nmap	Probe	1	Xterm	u2r	3
Portsweep	Probe	1	Rootkit	u2r	3
Mscan	Probe	1	Ps	u2r	3
Back	Dos	2	Perl	u2r	3
Smurf	Dos	2	guess_passwd	r2l	4
Processtable	Dos	2	Multihop	r2l	4
Land	Dos	2	Xlock	r2l	4
Teardrop	Dos	2	httptunnel	r2l	4
Worm	Dos	2	ftp_write	r2l	4
Neptune	Dos	2	warezmaster	r2l	4
apache2	Dos	2	Xsnoop	r2l	4
Pod	Dos	2	Sendmail	r2l	4
Udpstorm	Dos	2	Imap	r2l	4
Mailbomb	Dos	2	warezclient	r2l	4
			snmpguess	r2l	4
			Named	r2l	4
			Phf	r2l	4
			Spy	r2l	4
			snmpgetattack	r2l	4
			Normal	normal	5

TABLE 2: Attacks Names and Grouping Numbers.

NSL-KDD dataset contains two main text files which can be retrieved from their Internet site [3]:

- KDDTrain+.txt
- KDDTest+.txt

All observations in both files are labeled by their corresponding attack type. They were imported into Matlab environment and saved in a Matlab format file to speedup computation. Both training and testing observations are grouped into two groups, normal and anomaly. And the two datasets are formed which are called TrainDataSet2Class and TestDataSet2Class. They are used in our experiments in case of two categories. In similar way, the observations are grouped into two other new datasets having five output categories which are called TrainDataSet5Class and TestDataSet5Class. Figure2 gives the result of Matlab 'whos' command which explores the data tables.



```
>> whos('-file','nsl-kdd_grouped11062018.mat')
  Name                                   Size                   Bytes  Class

  TestDataSet2Class                        -                  6969195  table
  TestDataSet5Class                        -                  6969545  table
  TrainDataSet2Class                       -                 38824181  table
  TrainDataSet5Class                       -                 38824531  table
```

FIGURE 2: Matlab Dataset Tables (Result of whos command)

No.	Feature	type	Range
1	'duration'	'real'	''
2	'protocol_type'	'discrete'	'"tcp","udp", "icmp"'
3	'service'	'discrete'	'"aol", "auth", "bgp", "courier", "csnet_ns", "ctf", "daytime", "discard", "domain", "domain_u", "echo", "eco_i", "ecr_i", "efs", "exec", "finger", "ftp", "ftp_data", "gopher", "harvest", "hostnames", "http", "http_2784", "http_443", "http_8001", "imap4", "IRC", "iso_tsap", "klogin", "kshell", "ldap", "link", "login", "mtp", "name", "netbios_dgm", "netbios_ns", "netbios_ssn", "netstat", "nnsp", "nntp", "ntp_u", "other", "pm_dump", "pop_2", "pop_3", "printer", "private", "red_i", "remote_job", "rje", "shell", "smtp", "sql_net", "ssh", "sunrpc", "supdup", "systat", "telnet", "tftp_u", "tim_i", "time", "urh_i", "urp_i", "uucp", "uucp_path", "vmnet", "whois", "X11", "Z39_50"'
4	'flag'	'discrete'	'"OTH", "REJ", "RSTO", "RSTOS0", "RSTR", "S0", "S1", "S2", "S3", "SF", "SH"'
5	'src_bytes'	'real'	''
6	'dst_bytes'	'real'	''
7	'land'	'discrete'	'0 , 1'
8	'wrong_fragment'	'real'	''
9	'urgent'	'real'	''
10	'hot'	'real'	''
11	'num_failed_logins'	'real'	''
12	'logged_in'	'discrete'	'0 , 1'
13	'num_compromised'	'real'	''
14	'root_shell'	'real'	''
15	'su_attempted'	'real'	''
16	'num_root'	'real'	''
17	'num_file_creations'	'real'	''
18	'num_shells'	'real'	''

19	'num_access_files'	'real'	''
20	'is_host_login'	'discrete'	'0 , 1'
21	'is_guest_login'	'discrete'	'0 , 1'
22	'count'	'real'	''
23	'srv_count'	'real'	''
24	'serror_rate'	'real'	''
25	'srv_serror_rate'	'real'	''
26	'rerror_rate'	'real'	''
27	'srv_rerror_rate'	'real'	''
28	'same_srv_rate'	'real'	''
29	'diff_srv_rate'	'real'	''
30	'srv_diff_host_rate'	'real'	''
31	'dst_host_count'	'real'	''
32	'dst_host_srv_count'	'real'	''
33	'dst_host_same_srv_rate'	'real'	''
34	'dst_host_diff_srv_rate'	'real'	''
35	'dst_host_same_src_port_rate'	'real'	''
36	'dst_host_srv_diff_host_rate'	'real'	''
37	'dst_host_serror_rate'	'real'	''
38	'dst_host_srv_serror_rate'	'real'	''
39	'dst_host_rerror_rate'	'real'	''
40	'dst_host_srv_rerror_rate'	'real'	''
41	'class'	'discrete'	'"normal", "anomaly"'

TABLE 3: Feature Types and Ranges.

5. PROPOSED METHOD

As mentioned before, NSL-KDD dataset for intrusion detection systems is used. Flow of data is illustrated in Figure3. The data is passed through three phases, during each phase it is transformed from a form to another.

In data preparation phase, the input is NSL-KDD dataset in its original form. It has many records and contains numerical and categorical types, in addition, the features are of variant scale, because of all we need to prepare it first. More detailed operations on data preparation phase was pictured in Figure4. The categorical features are transformed into numerical values using dummy variables method [9,Section 3.1]. Dummy rule is more suitable since all categorical features are of nominal values. 40 features in NSL-KDD are transformed to get a dataset of 118 numeric features. Then they are normalized using Z-standard and Min-Max methods [15,Section 4.3]. Uniform sampling is used to reduce the number of observations which is controlled by the user.

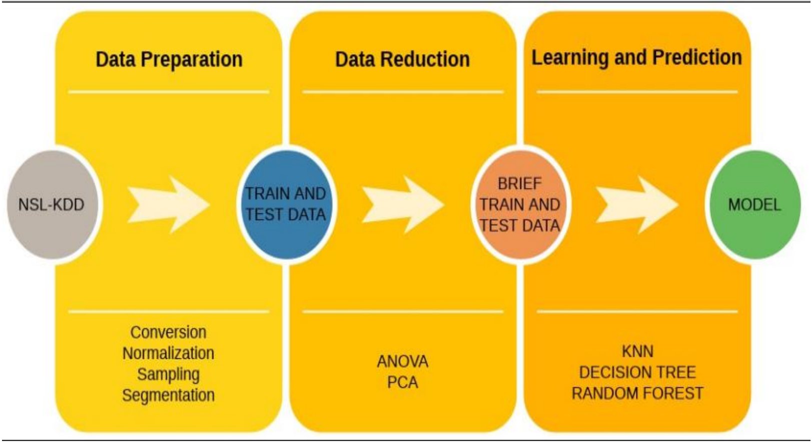


FIGURE 3: Data Flow for a Proposed Technique.

The last operation in data preparation phase is data segmentation. The normalized dataset is segmented into two sections. The sampled data contains the training dataset while the left data is kept as a testing dataset. Both datasets are fed as input to data reduction phase, see Figure5. The training dataset is analyzed by ANOVA method. As a result, features are ranked according to their contribution in discriminating the output classes, higher rank features have more importance and so on. In this phase, the most important features are identified and used for further processing. A statistical variable, F-value, is estimated for each feature using the following relation:

$$F = \frac{SS_T/(k-1)}{SSE/(n-k)} \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (1)$$

Where k – 1 and n – k are the degrees of freedom, SSE is the sum of square variation within the data of same class or category, and SS_T is the sum of square variation between the data of different classes. If the variation SSE is small compared to the variation SS_T, then we would expect to observe a large value of F statistic and have an evidence that there is a significance difference between classes using the tested feature otherwise the feature is useless [19].

Based on the results of analysis, feature selection operation selects a feature subset from the whole features, features of greatest rank values are selected from 118 available features. The selected features exhibit a high correlation, as it is obviously seen from the correlation matrix image snapshot of a subset of 32 features, see Figure6.

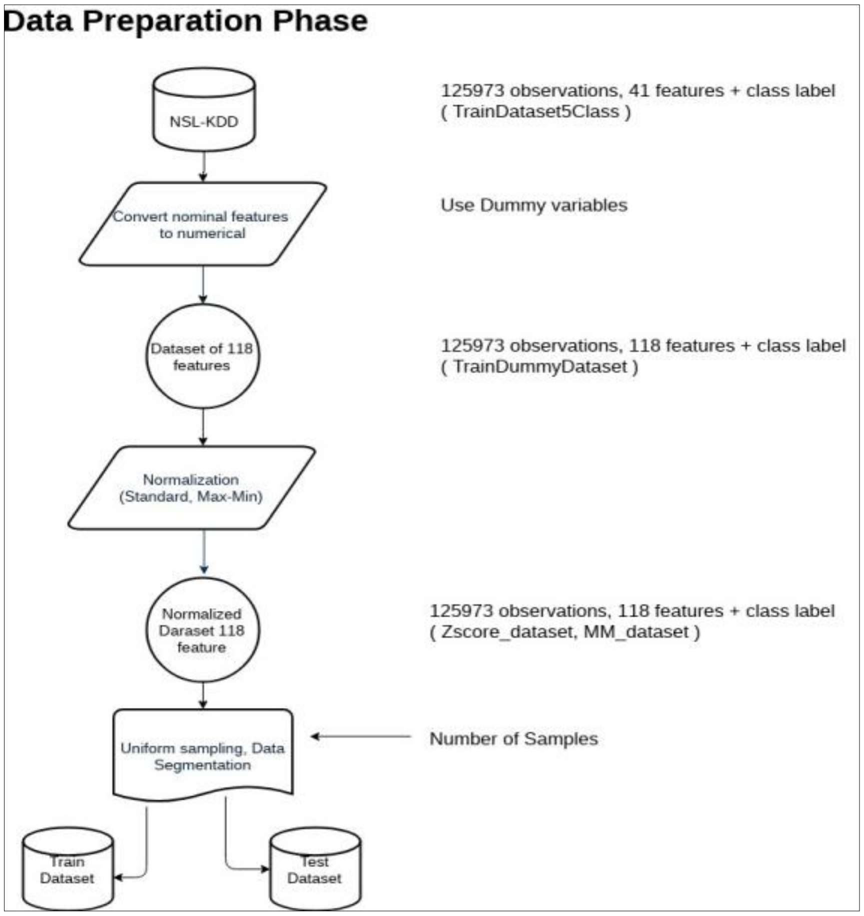

Data Preparation Phase
NSL-KDD
125973 observations, 41 features + class label (TrainDataset5Class)
Convert nominal features to numerical
Use Dummy variables
Dataset of 118 features
125973 observations, 118 features + class label (TrainDummyDataset)
Normalization (Standard, Max-Min)
Normalized Daraset 118 feature
125973 observations, 118 features + class label (Zscore_dataset, MM_dataset)
Uniform sampling, Data Segmentation
Number of Samples
Train Dataset
Test Dataset

FIGURE 4: Data Preparation Flowchart.

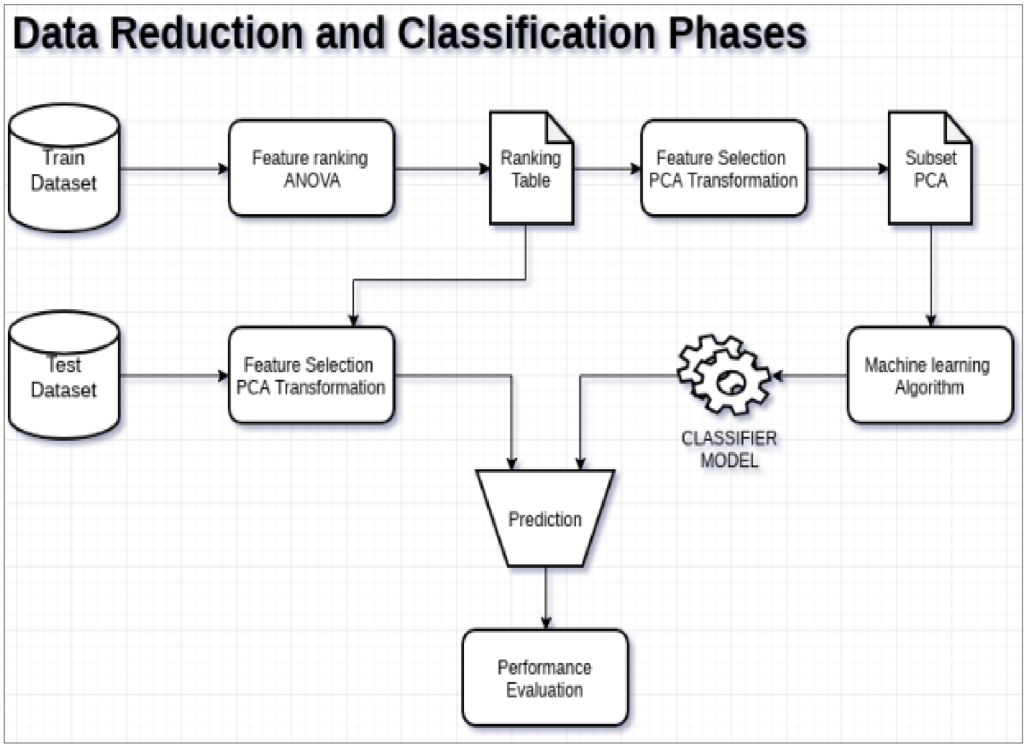

Data Reduction and Classification Phases
Train Dataset
Feature ranking ANOVA
Ranking Table
Feature Selection PCA Transformation
Subset PCA
Test Dataset
Feature Selection PCA Transformation
CLASSIFIER MODEL
Machine learning Algorithm
Prediction
Performance Evaluation

FIGURE 5: Data Reduction and Classification Flowchart.

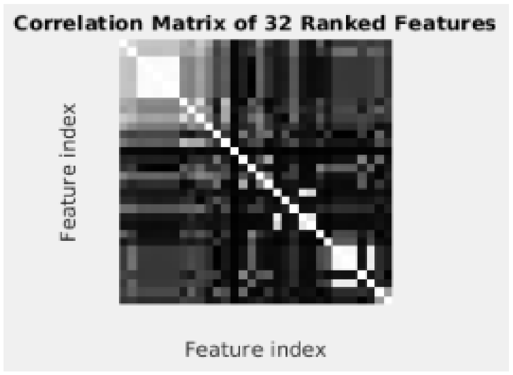


FIGURE 6: Simulated Correlation Matrix Image of 32 Features.

White and light gray spots means high correlation factor as shown in upper left and lower right corners, which means first features are correlated to each other, and last indexed features are also correlated to each other. We can invest this phenomena by using PCA tool to transform data into another feature domain, where most of the data is focused in a fewer number of non-correlated features called principle components, more information about PCA method in [20].

A subset of 5 to 10 PCA components are sufficient to train a machine learning algorithm, Figure4. Several machine learning algorithms can be used as a result a classifier model is invented. At this point the training stage is finished and testing stage is initiated by selecting the highest ranked features from the testing dataset. Since classifier model is working on PCA domain, the selected features are being transformed else, and transformed data with classifier model are used to drive the prediction process to evaluate the performance of the classifier.

6. EXPERIMENT AND DISCUSSION

6.1 Evaluation and Performance Measures

A dataset used for performance evaluation is called a test dataset. It should contain the correct labels (observed labels) for all data instances, as in Figure7(a). These observed labels are used to compare with the predicted labels for performance evaluation after classification [21].

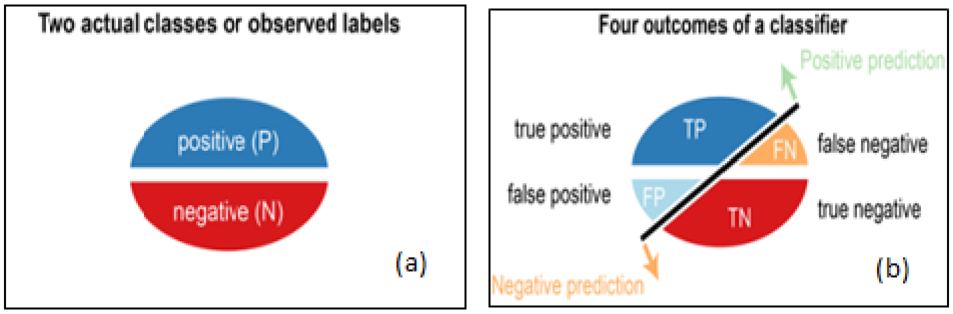


FIGURE 7: (a) Observed labels (Correct Instances); **(b)** Predicted labels after classification [21].

Classification process outputs four types of predicted labels for each class, see Figure7(b):

- True positive (TP): Correct positive predictions
- False positive (FP): Incorrect positive predictions.
- True negative (TN): Correct negative predictions.
- False negative (FN): Incorrect negative predictions.

The following performance measures are estimated from above quantities [21]:

Accuracy: Accuracy is calculated as the number of all correct predictions divided by the total number of the dataset. The best accuracy is 1, whereas the worst is 0.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad \text{..............} \ (2)$$

Error rate = Accuracy -1.

Sensitivity: is calculated as the number of correct positive predictions divided by the total number of positives. It is also called recall or true positive rate (TPR). The best sensitivity is 1, whereas the worst is 0.

$$Sensitivity(TPR) = \frac{TP}{TP+FN} \quad \text{...............} \ (3)$$

Specificity: is calculated as the number of correct negative predictions divided by the total number of negatives. It is also called true negative rate (TNR). The best specificity is 1, whereas the worst is 0.

$$Specifity(TNR) = \frac{TN}{TN+FP} \quad \text{................} \ (4)$$

False alarm rate: is called false positive rate (FPR), and calculated as the number of incorrect positive predictions divided by the total number of negatives. The best false positive rate is 0.

whereas the worst is 1. It can also be calculated as 1 – specificity.

$$False\ alarm\ rate\ (FPR) = \frac{FP}{FP + TN} \quad \text{.........} \ (5)$$

6.2 Feature Ranking Results

ANOVA is a statistical tool used to find out the features having significant differences. In our experiment 128 numerical features were ranked and sorted in descendant order. Using Matlab simulation, the features and their F-values are depicted in Figure8.

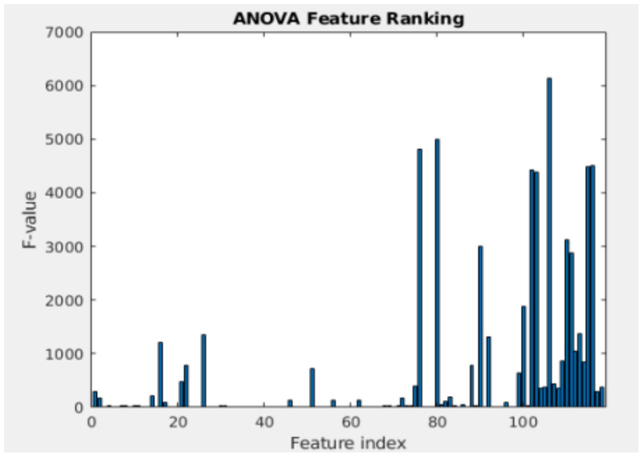


FIGURE 8: Simulated ANOVA Feature Ranking.

The 32 highest F-value features are listed down in Table4.

Feature	F-value	Feature	F-value
flag_RSTR	6136	dst_host_diff_srv_rate	856
same_srv_rate	4989	dst_host_count	853
flag_SF	4822	dst_host_srv_diff_host_rate	790
flag_S0	4517	hot	785
dst_host_srv_serror_rate	4481	service_ftp_data	728
dst_host_serror_rate	4431	service_private	631
serror_rate	4389	is_guest_login	475
srv_serror_rate	3128	service_ftp	433
dst_host_srv_count	2993	diff_srv_rate	393
logged_in	2883	flag_RSTR	374
dst_host_same_srv_rate	1879	dst_host_srv_rerror_rate	370
count	1366	srv_rerror_rate	364
dst_host_same_src_port_rate	1346	rerror_rate	358
service_http	1312	srv_diff_host_rate	303
root_shell	1217	dst_host_rerror_rate	290
service_eco_i	1047	protocol_tcp	211

TABLE 4: The first highest 32 feature ranking as resulted from simulation.

6.3 PCA Feature Transformation

From 128 features, 32 features were filtered and others discarded. The filtered features were further transformed into 10 features using PCA tool. 90% of data variance was perceived as shown in figure9.

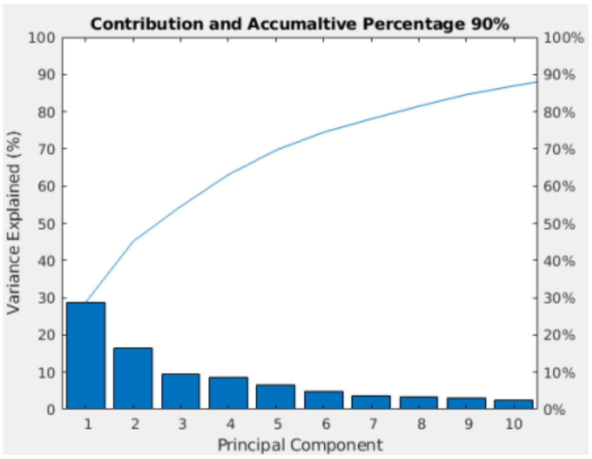


FIGURE 9: 90% PCA Explanation for 10 PCAs.

6.4 Machine Leaning and Evaluation

in our experiment, three types of machine learning algorithms are used:

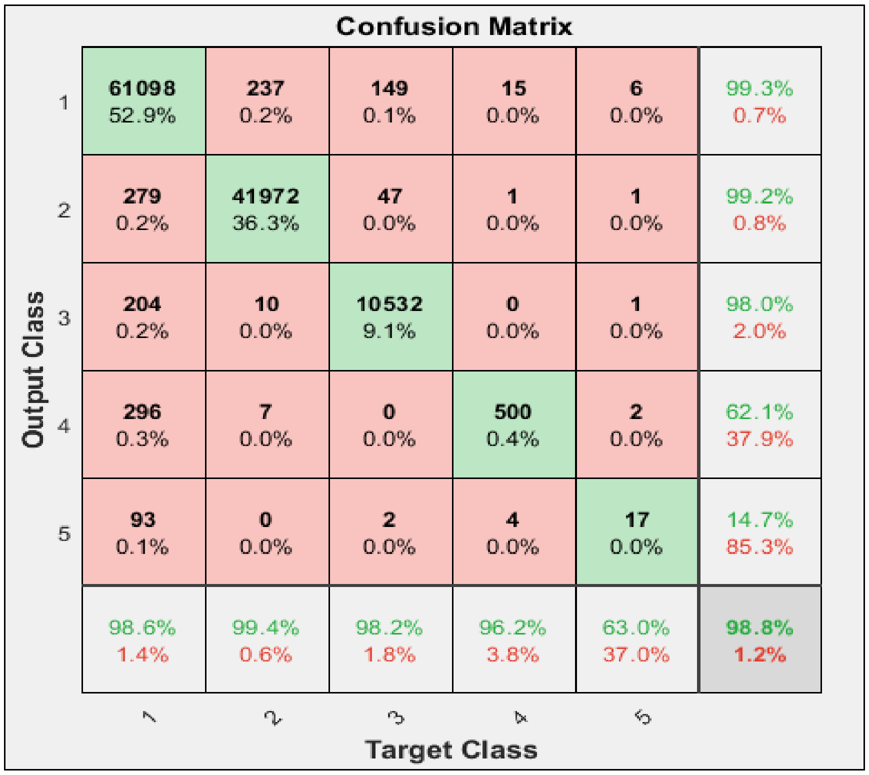
- KNN (k=1)
- Decision Tree
- Ensemble (Random Forest)

As shown in Table5, KNN and Random Forest gave better results compared to decision tree. Processing speeds, number of observations per second, for training and prediction phases were estimated, the training phase included ANOVA and PCA stages. KNN classifier outperformed the others regarding training speed and most of performance parameters. Five category classification was used, and Figure10 showed the confusion matrix of KNN classifier model. Ten PCAs training dataset was used to train the classifier, and its 10 x 10 scatter matrix showed the distribution of training observations in 2D view for the five categories, see Figure11.

KNN Algorithm K=1			
Classes	**Sensitivity**	**Specificity**	**False Alarm rate**
Normal	98.81%	99.14%	0.86%
Dos	99.32%	99.53%	0.47%
Probe	97.71%	99.81%	**0.19%**
U2R	95.57%	99.79%	0.21%
R2L	57.14%	99.94%	0.06%
Accuracy	98.90%		
ANOVA-PCA +Training Speed	1155 Obs/s		
Prediction Speed	142152 Obs/s		
Decision Tree			
Classes	**Sensitivity**	**Specificity**	**False Alarm rate**
Normal	97.99%	98.68%	1.32%
Dos	98.74%	99.05%	0.95%

Probe	95.91%	99.62%	**0.38%**
U2R	90.54%	99.66%	0.34%
R2L	17.24%	99.92%	0.08%
Accuracy	98.00%		
ANOVA-PCA +Training Speed	1092 Obs/s		
Prediction Speed	758403 Obs/s		
Random Forest			
Classes	**Sensitivity**	**Specificity**	**False Alarm rate**
Normal	98.95%	98.86%	1.14%
Dos	98.97%	99.64%	0.36%
Probe	97.88%	99.79%	**0.21%**
U2R	94.23%	99.81%	0.19%
R2L	40.74%	99.96%	0.04%
Accuracy	98.80%		
ANOVA-PCA +Training Speed	366 Obs/s		
Prediction Speed	16184 Obs/s		

TABLE 5: Machine Learning Algorithm Performance Results.


Confusion Matrix
Output Class
Target Class

Output Class \ Target Class	1	2	3	4	5	
1	61098 52.9%	237 0.2%	149 0.1%	15 0.0%	6 0.0%	99.3% 0.7%
2	279 0.2%	41972 36.3%	47 0.0%	1 0.0%	1 0.0%	99.2% 0.8%
3	204 0.2%	10 0.0%	10532 9.1%	0 0.0%	1 0.0%	98.0% 2.0%
4	296 0.3%	7 0.0%	0 0.0%	500 0.4%	2 0.0%	62.1% 37.9%
5	93 0.1%	0 0.0%	2 0.0%	4 0.0%	17 0.0%	14.7% 85.3%
	98.6% 1.4%	99.4% 0.6%	98.2% 1.8%	96.2% 3.8%	63.0% 37.0%	98.8% 1.2%

FIGURE 10: Confusion Matrix of KNN Model.

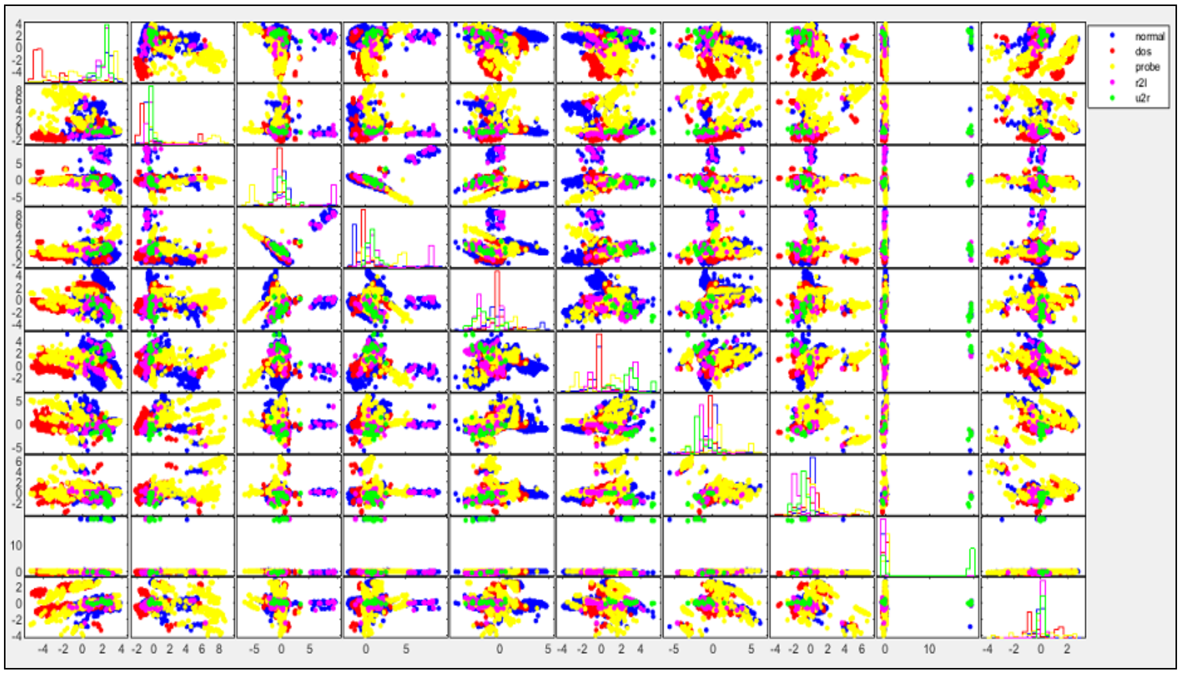


FIGURE 11: Scatter Distribution of Five Categories: NORMAL, DOS, PROBE, R2L, and U2R.

6.5 Comparison between ANOVA, NCA and ReliefF Methods

The proposed method was compared with two robust and familiar FS methods called Neighbor Component Analysis [14] and ReliefF [22]. Both methods are adopted in MATLAB release 2018a, therefore our dataset was used to check the performance and compare the results with what obtained before. FS method was followed by PCA, and 32 selected features are transformed to 10 PCAs, and KNN classifier. Accuracy and speed of ranking were the highlighted terms.

Results in Table6 showed that NCA played a little bit better in accuracy than ANOVA, but ANOVA ranking computation speed was faster.

FS Method	**Selected Features**	**Accuracy**	**Ranking Speed (observation/sec)**
ANOVA	32	98.77%	38130
NCA	32	99.13%	62
ReliefF	32	98.50%	89

TABLE 6: Comparison between ANOVA, NCA, and Relief.

6.6 Comparison with Other Related Research Methods

The results obtained from the research can be compared with other research results. The comparison includes the FS method, the type of classifier, the dataset, the number of selected features, The transformation method-if present, and the classification accuracy. When comparing the results with other approaches in Table7, we find that our experimental results are better than other approaches in terms of classification accuracy.

FS Method	Classifier	Dataset	Selected Features	Transform Method	Accuracy
ANOVA-PCA [Proposed Technique]	KNN	NSL-KDD	32	PCA (10)	98.80%
Fast Correlation Based Filter [7]	NBTREE	NSL-KDD	12		94.60%
Manual Selection [8]	ANN	NSL-KDD	29		81.20%
Hybrid FS Technique [12]	KNN	NSL-KDD	19	Sub-medoid (1)	97.42%
ANOVA [13]	SVM	SPAM BASE	52 out of 57		93.55%

TABLE 7: Comparison between Different Related Methods.

7. CONCLUSION AND FUTURE WORK

Data preparation is the process of transforming data into a form suitable for analysis and mining. NSL-KDD dataset is transformed into numerical and normalized data. ANOVA is a statistical method used to discover the most significant features so that unrelated features are discarded. Further reduction such us PCA was helpful since there found high correlation between selected features.

Our proposed method, ANOVA-PCA, was used to reduce the number of features of NSL-KDD from 41 to 10, and classification performance evaluation gave good results for all classifier types, and KNN classifier obtained the best results. R2L class had poor sensitivity or detection rate due to few number of observations contained in NSL-KDD. As compared with other FS algorithms, ANOVA obviously over perfumed NCA and ReliefF in term of computation speed. With a little bit degradation in classification accuracy.

The proposed work was a filter type FS technique which gave local optimal subset of features. In Future, a mix of filter and wrapper type FS techniques can be implemented to achieve more data reduction. For example, using of ANOVA method first to choose initial set of features followed by an efficient wrapper method to search for the optimal subset from the initial set may be helpful.

8. REFERENCES

[1] J.P. Nziga. ”Minimal Dataset for Network Intrusion via Dimensionality Reduction.” Sixth International Conference on Digital Information Management ICDIM, 2011.

[2] M. Tavallaee, E. Bagheri, Wei Lu, and Ali A. Ghorbani. “A Detailed Analysis of the KDD CUP 99 Data Set.” IEEE Symposium on Computational Intelligence in Security and Defense Applications, 2009.

[3] University of New Brunswick Canadian Institute for Cyber-Security “NSL-KDD Dataset.” Internet: https://www.unb.ca/cic/datasets/nsl.html, Nov. 21, 2018.

[4] D. H. Deshmukh, T. Ghorpade, P. Padiya. "Intrusion Detection System by Improved Preprocessing Methods and Naive Bayes Classifier using NSL-KDD99 Dataset.” International Conference on Electronics and Communication Systems (ICECS), 2014.

[5] A. Ozgur, H. Erdem. (2016, Apr 14). ”A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015.” PeerJ Preprints. Available: https://peerj.com/preprints/1954/

[6] Amrita & P Ahmed. (2012, Sep). “A Study of feature selection methods in Intrusion Detection System: A Survey”, International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR) , ISSN 2249-6831 , Vol.2, Issue 3, pp. 1-25. Available: https://www.researchgate.net/publication/317543749

[7] D. H. Deshmukh, T. Ghorpade, P. Padiya. ”Improving Classification Using Preprocessing and Machine Learning Algorithms on NSL-KDD Dataset” International Conference on Communication, Information & Computing Technology (ICCICT),2015 Jan. 16-17, Mumbai, India.

[8] B. Ingre, A. Yadav. ”Performance Analysis of NSL-KDD dataset using ANN” International Conference on Signal Processing and Communication Engineering Systems,2015, pp. 93-96.

[9] Y. Bouzida, F. Cuppens, N. Cuppens-Boulahia and S. Gombault, (2004, Jan) “Efficient Intrusion Detection Using Principal Component Analysis” https://www.researchgate.net/publication/267821847_Efficient_Intrusion_Detection_Using_Principal_Component_Analysis, 2004, Jan.

[10] G. Meena, R. R. Choudhary. “A Review Paper on IDS Classification using KDD99 and NSL-KDD Dataset in WEKA” International Conference on Computer, Communications and Electronics, July 01-02, 2017.

[11] K. Ibrahimi, M. Ouaddane. ”Management of Intrusion Detection Systems based-KDD99: Analysis with LDA and PCA” International Conference on Wireless Networks and Mobile Communications (WINCOM), Mar 17, 2017.

[12] I. Z. Muttaqien, T. Ahmad. “Increasing Performance of IDS by Selecting and Transforming Features” IEEE International Conference on Communication, Networks and Satellite, 2016 Dec 8-10, Surabaya, Indonesia

[13] N. Elssied, O. Ibrahim and A. Osman. (2014, Jan). “A Novel Feature Selection Based on One-Way ANOVA F-Test for E-Mail Spam Classification.” Research Journal of Applied Sciences, Engineering and Technology 7(3): pp. 625-638

[14] W. Yang, K. Wang, W. Zuo. (2012, Jan). "Neighborhood Component Feature Selection for High-Dimensional Data." Journal of Computers. Vol. 7, Number 1.

[15] M. N. Abdullah and M. M. Ahmed. “Dataset Analysis and Preprocessing for Intrusion Detection Using Machine Learning Techniques.” 3th Engineering Conference University of Aden- Faculty of Engineering, Mar 17-18, 2019, pp. 165-176.

[16] H. G. Kayacik, N. Zincir-Heywood. “Generating Representative Traffic for Intrusion Detection System Benchmarking.” 3rd Annual Communication Networks and Services Research Conference (CNSR'05) , May 16-18, 2005, pp. 4-5

[17] N. Paulauskas, J. Auskalnis. “Analysis of Data Pre-processing Influence on Intrusion Detection using NSL-KDD Dataset.” Open Conference of Electrical, Electronic and Information Sciences (eStream), Jun 17, 2017 IEEE

[18] L. Dhanabal, S.P. Shantharajah. “A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms”, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 4, Issue 6, Jun, 2015.

[19] W. L. Martinez, M. Cho. Statistics in Matlab A Primer. United Kingdom: Chapman & Hall/CRC, 2015, pp. 134.

[20] M. F. Elrawi, T. K. Abdelhamid, and A. M. Mohamed. (2013, July). “IDS IN TELECOMMUNICATION NETWORK USING PCA.” International Journal of Computer Networks & Communications. Vol.5, No.4. pp. 147-157.

[21] T. Saito, M. Rehmsmeier. (2015,Mar 4). "The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets.": PLOS ONE , Available:
https://drive.google.com/open?id=0Bx5AC3BOw_m7ODR2bzBlTTBtajg&authuser=0

[22] M. Robnik-Sikonja, and I. Kononenko, (2003). "Theoretical and empirical analysis of ReliefF and RReliefF." Machine Learning Journal, pp. 23-69.