International Journal of Computational Intelligence Research
ISSN 0973-1873 Volume 13, Number 8 (2017), pp. 1963-1976
© Research India Publications
http://www.ripublication.com

A Novel Approach for Intrusion Detection System Using feature Selection algorithm

Ananda Kumar K S*, [1]Arpitha K, [2]Latha M N and [3]Sahana M

**Asst.Professor, Department of Information Science and Engineering*
[1, 2 & 3]Department of Information Science and Engineering
RajaRajeswari College of Engineering, Bangalore, India

Abstract

Abundance and unnecessary features in the large amount of data have caused a problem in data traffic classification which in turn slowdowns the classification process. Not only does this it also not allow the classifier for making extract decisions, which play a major role in big data. This system uses an algorithm based on mutual information which in turn selects the optimal features for classifications analytically, since it can handle linear and non linear features. Its efficiency can be evaluated in the network detection system. An Intrusion Detection System (IDS) named Least Square Support Vector Machine is fabricated using the feature selected by the algorithm. The performance of LSSVM-IDS can be obtained using three kinds of dataset namely KDD Cup 99, NSL-KDD and Kyoto 2006 dataset. The results show that algorithm contributes more critical features for the LSSVM-IDS to accomplish better exactness and lower computational cost.

Keywords: Big data, Classifier, Intrusion Detection System, Performance, Support Vector Machines.

I. INTRODUCTION

In spite of expanding consciousness of system security, the current arrangements stay unequipped for completely ensuring web applications and PC systems against the dangers from perpetually progressing digital assault procedures, for example, DOS assault and PC malware. Not with standing extending nature with framework security, the current courses of action remain unequipped for totally guaranteeing web applications and PC frameworks against the threats from attack techniques, for example DoS, strike and PC malware. Creating successful and versatile security approaches, in this manner, has turned out to be more basic than at any other time. The customary security strategies, as the principal line of security guard, for example, client confirmation, firewall and information encryption, are deficient to completely cover the whole scene of system security challenges from ever-developing interruption aptitudes and procedures [1]. Thus, a different line of security safeguard is exceptionally suggested, for example, Intrusion Detection System (IDS). As of late an IDS close by with hostile to infection programming has progressed towards becoming an imperative supplement to the security framework. The blend of these two lines gives a more far reaching resistance against those dangers improves system security. A lot of research has been led to create keen interruption recognition methods, which help accomplish better system security. Packed away boosting-based on C5 choice trees [2] and Kernel Miner [3] are two of the most punctual endeavors to assemble interruption location plans. Techniques proposed in previous research have adequately associated machine learning procedures, for instance, Support Vector Machine (SVM), to mastermind compose action plans that don't organize customary framework movement. Both frameworks were outfitted with five particular classifiers to identify ordinary movement, four distinct sorts of attacks (i.e., Denial of service, examining, User to root, Root to local). Exploratory outcomes demonstrate the viability and power of utilizing SVM in IDS.

To achieve better accuracy and lower computational cost in intrusion detection system using feature selection algorithm. Repetitive and immaterial components in information have brought about a long haul issue in system movement characterization. These components back off the procedure of order as well as keep a classifier from settling on exact choices, particularly when adapting to huge information.

II. RELATED WORKS

Existing arrangements stay unequipped for completely ensuring web applications and PC systems against the dangers from constantly progressing digital assault methods, for example, DOS assault and PC malware. Current system movement information, which are regularly tremendous in size, show a noteworthy test to IDSs. These "huge information" back off the whole recognition prepare and may prompt unacceptable arrangement precision because of the computational challenges in taking care of such information. Arranging a gigantic measure of information more often than not causes numerous scientific troubles which then prompt higher computational intricacy. Vast scale datasets for the most part contain loud, excess, or uninformative elements which introduce basic difficulties to learning disclosure and information demonstrating.

Chandrasekhar, K. Raghuveer et al suggested that Intrusion recognition is not yet a flawless innovation. The chance to make a few imperative commitments to the field of interruption recognition utilizing information mining Concepts [05]. In this framework, framework has proposed another strategy by using information mining methods, for example, neuro-fluffy and spiral premise bolster vector machine (SVM) for the interruption discovery framework. Their proposed method has four noteworthy strides in which, initial step is to play out the Fuzzy C-implies bunching. At that point, neuro-fluffy is prepared, to such an extent that each of the information point is prepared with the comparing neuro-fluffy classifier related with the bunch. In this manner, a vector for SVM grouping is framed and in the fourth step, order utilizing spiral SVM is performed to recognize interruption has happened or not. Informational index utilized is the KDD glass 99 dataset and framework has utilized affectability, specificity and precision as the assessment measurements parameters. Our system could accomplish better precision for a wide range of interruptions. It accomplished around 98.94 precision in the event of DOS assault and achieved statures of 97.11 exactness if there should arise an occurrence of PROBE assault.

S. Mukkamala, A. H. Sung, A. Abraham et al proposed Soft figuring procedures are progressively being utilized for critical thinking. This framework addresses utilizing an outfit approach of various delicate processing and hard figuring strategies for interruption location. Because of expanding episodes of digital assaults, building successful interruption discovery frameworks are fundamental for ensuring data frameworks security, but then it remains a slippery objective and an awesome test. Framework concentrated the execution of Artificial Neural Networks (ANNs), Support Vector Machines (SVMs) and Multivariate Adaptive Regression Splices (MARS). Framework demonstrates that a group of ANNs, SVMs and MARS is better than individual methodologies for interruption discovery [6].

III. METHODOLOGY

An intrusion detection system (IDS) is a device or software application that monitors a network or systems for malicious activity or policy violations. Any detected activity or violation is typically reported either to an administrator or collected centrally using a security information and event management (SIEM) system.

A. Intrusion Detection Framework on Least Square Vector Machine

The framework of the proposed intrusion detection system is depicted in figure 1. The detection framework is comprised of four phases: (1) data collection (2) data preprocessing (3) classifier training, and (4) attack recognition.

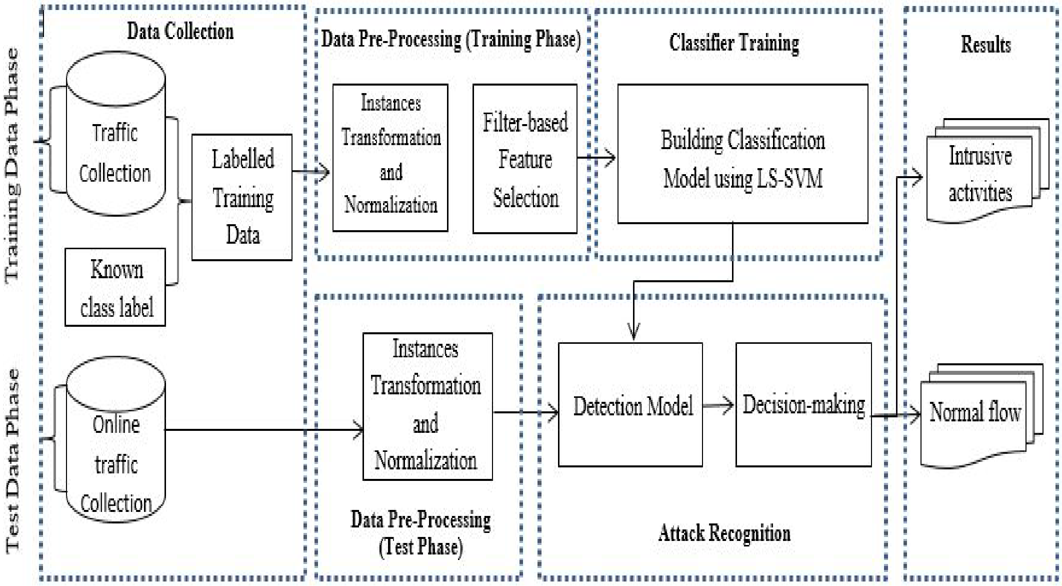

Training Data Phase
Test Data Phase
Data Collection
Traffic Collection
Labelled Training Data
Known class label
Online traffic Collection
Data Pre-Processing (Training Phase)
Instances Transformation and Normalization
Filter-based Feature Selection
Classifier Training
Building Classification Model using LS-SVM
Results
Intrusive activities
Normal flow
Instances Transformation and Normalization
Data Pre-Processing (Test Phase)
Detection Model
Decision-making
Attack Recognition

Fig 1: The framework of the LS-SVM-based Intrusion Detection System.

Data Collection

In data collection we collect a data from the KDD Cup 99 dataset where data is collected based on two factors design and effectiveness of IDS.

Data Preprocessing

In data preprocessing we have stages which are explained below

Data Transferring

Here in data transferring ever symbolic feature in dataset is converted to integer type. For example, the KDD CUP 99 dataset contains both symbolic feature and integer type, these symbolic features include TCP,UDP further it is replaced with integer type.

Data Normalization

There are 3 types of normalization step. In first one we delete all the duplicate data and unwanted data. In second step we generalize the some of the field values. In third step we put zero for the entire field which do not contain any value or which are empty. With help of this step comparison becomes easy.

Feature Selection

In Feature Selection the values which system have got are compared with trained dataset and only some features are selected based on the algorithm flexible mutual information based feature selection and flexible linear correlation coefficient based feature selection.

Attack recognition

In this there are two main steps, in first step the system takes the data and compare with the trained dataset and recognitions if the data is attacked or normal data. If the data is attack then it will undergo the second step, in which the attacked data is classified according to which type of attack is occurred by comparing it with the trained dataset.

IV. RESULTS AND DISCUSSIONS

A. Experimental Setup

Currently there are only few public datasets available for intrusion detection evaluation. Among them we have taken the data from the KDD CUP 99 and further it is pre-processed. In this step we have data transformation and data normalization. In data transformation step we need to change all the alphabetical values into numerical values because we can’t compare the alphabetical values with numeric. In data normalization we remove all unwanted data from the data set. After normalization next come the feature selection where the result clearly demonstrates that the classification performance of IDS is enhanced by the feature selection step. Moreover, the algorithm FMIFS shows results in terms of low computational cost and high classification results. This is done using the two different algorithms as mentioned above. After building the complete intrusion detection system we have to check its accuracy using the KDD-CUP 99 dataset .There are different scenarios present to calculate the efficiency and many more parameters of the project such as precision, F-measure etc. The whole GUI of the project contains many phases in the form of button each step that is being carried out and the main GUI of the whole project is embedded in the final GUI that is the confusion matrix GUI, the features obtained in the feature selection step are classified based on the labels of class that is already being classified using the trained data set. The final result that is final GUI of the classified data set its accuracy, precision, F-measure using the values of TP (true positive), TN(true negative),FP(false positive),FN(false negative). Then based on the results obtained for each classes accuracy, F-measure, precision are calculated for each attack found using the above mentioned parameters, later based on the values you get and the values already got by the trained data set is compared and they are classified as number of wrong predictions got and the number of correct predictions done. Based on this the overall efficiency of our builded intrusion detection system using an efficient algorithm known as Filter-Based Feature selection algorithm. The GUI screen shot used in the next page will be able to explain the main result obtained from the overall project. The project contains one basic GUI for the ease use of the user authenticator and later the GUI is constructed according to the ease of the user. The user has no strain of giving any input everything is automatically calculated and results are finalized, which will be helpful for the user to have smooth full use.

B. Performance Evaluation

Several experiments have been conducted to calculate the performance and effectiveness of LSSVM-IDS. For this purpose, the accuracy rate, detection rate, false positive rate and F-measure metrics are applied. The accuracy metric, detection rate and false positive rate are defined by

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

$$\text{Detection Rate} = \frac{\text{TP}}{\text{TP+FN}}$$

$$\text{False Positive Rate} = \frac{\text{FP}}{\text{FP+TN}}$$

Where True Positive (TP) is the number of actual attacks classified as attacks, True Negative (TN) is the number of actual normal records classified as normal ones, False Positive (FP) is the number of actual normal records classified as attacks, and False Negative (FN) is the number of actual attacks classified as normal records.

C. Results and Discussions

The characterization execution of the interruption discovery display joined with FMIFS, MIFS (b =0:3), MIFS (b = 1) furthermore, FLCFS and the model utilizing all components in light of the three datasets are appeared in Table 2 and Fig. 2. The outcomes unmistakably exhibit that the grouping execution of an IDS is upgraded by the component determination step. Also, the proposed include choice calculation FMIFS appears promising outcomes as far as low computational cost and high grouping outcomes. Table 2 compresses the arrangement consequences of the diverse choice techniques as to location rates, false positive rates and precision rates. It demonstrates plainly that the identification display consolidated with the FMIFS has accomplished an exactness rate of 99.79, 99.91 and 99.77 percent for KDD Cup 99, NSL-KDD and Kyoto 2006+, separately, and fundamentally beats every single other strategy. What's more, the proposed location show consolidated with FMIFS appreciates the most elevated recognition rate and the least false positive rate in correlation with other consolidated location models. The proposed highlight choice calculation is computationally productive when it is connected to the LSSVM-IDS. The building (preparing) and test times by the discovery show utilizing FMIFS contrasted and the recognition show utilizing all elements. The figure demonstrates that the LSSVM-IDS + FMIFS performs superior to LSSVM-IDS with every one of the 41 includes on all datasets. There are noteworthy contrasts when performing probes KDD Cup 99 and NSL-KDD and a slight contrast on Kyoto 2006+ dataset by correlation with the two previously mentioned models.

Table 5.1: Performance classification for all attacks based on KDD Cup 99 data set

	KDD Cup 99		
	DR	FPR	Accuracy
LSSVM-IDS + FMIFS	99.46	0.13	99.79
LSSVM-IDS + MIFS (β = 0.3)	99.38	0.23	99.70
LSSVM-IDS + MIFS (β = 1)	89.26	0.34	97.63
LSSVM-IDS + FLCFS	98.47	0.61	98.41
LSSVM-IDS + All features	99.16	0.97	99.19

Table 5.1 summaries the classification results of the different selection methods in regard to detection rates, false positive rates and accuracy rates. It shows clearly that the detection model combined with the FMIFS has achieved an accuracy rate of 99.79, 99.91 and 99.77 percent for KDD Cup 99 and NSL-KDD. In addition, the proposed detection model combined with FMIFS enjoys the highest detection rate and the lowest false positive rate in comparison with other combined detection models.

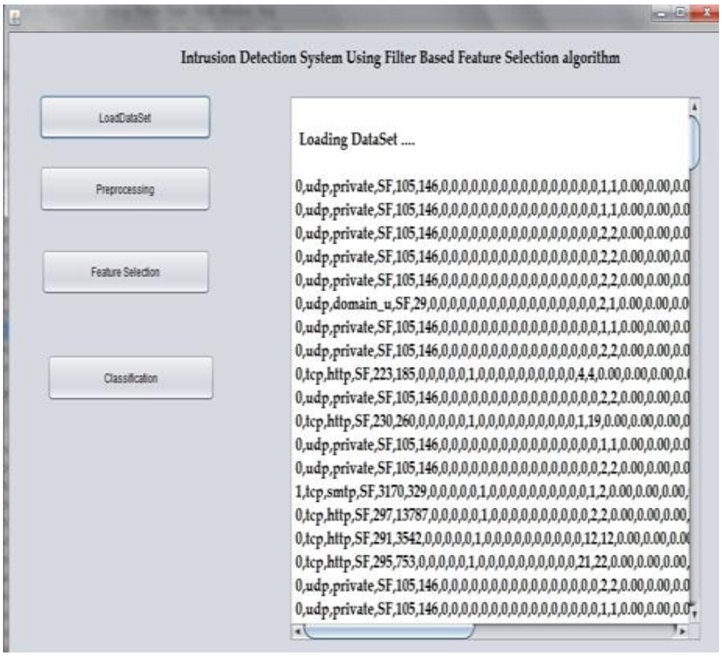
The proposed feature selection algorithm is computationally efficient when it is applied to the LSSVM-IDS. Table 5.1 shows the building (training) and test times consumed by the detection model using FMIFS compared with the detection model using all features. The above table shows that the LSSVM-IDS + FMIFS perform better than LSSVM-IDS with all 41 features on all datasets.

Table 5.2: Accuracy, Building Time (min) and Test Time (min)for All Different Classes on the Corrected Labels of the KDD Cup 99 Dataset Using LSSVM-IDS + FMIFS are Compared with Those Using PLSSVM + MMIFS

Class Name	Model	Accuracy (%)	Building time (min)	Testing time (min)
Normal	LSSVM-IDS+FMIFS PLSSVM+MMIFS	98.39 99.1	7.92 25	5.51 11
DoS	LSSVM-IDS+FMIFS PLSSVM+MMIFS	98.93 84.11	10.06 19	4.5 8
Probe	LSSVM-IDS+FMIFS PLSSVM+MMIFS	99.57 86.12	13.04 35	8.49 13
U2R	LSSVM-IDS+FMIFS PLSSVM+MMIFS	99.66 99.47	0.47 23	0.32 10
R2L	LSSVM-IDS+FMIFS PLSSVM+MMIFS	90.08 98.70	1.06 5	0.44 4
Overall	LSSVM-IDS+FMIFS PLSSVM+MMIFS	97.33 93.50	6.51 21.4	3.85 9.20

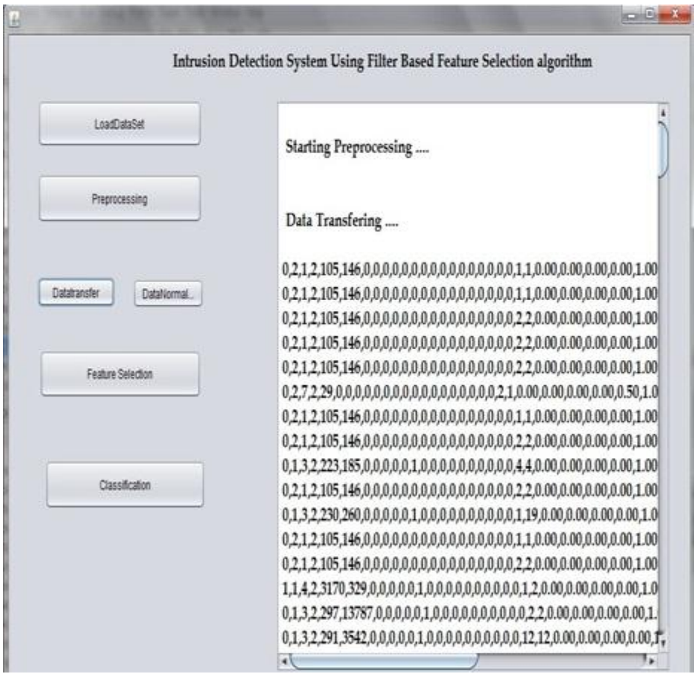
From Table 5.2, it shows that the proposed system reduces the building time and test time very considerably for all categories. It is clear from the table that LSSVM-IDS + FMIFS has better accuracy in DoS, Probe and U2R classes, while the PLSVM + MMIFS produces a better accuracy rate when applied to Normal and R2L class. Moreover, the table shows that LSSVM-IDS + FMIFS out performs the PLSSVM + MMIFS model in the overall performance.

The below figure 2 shows this is the very first step of the implementation, in this step we load the data set which is collected from the KDD cup.


Intrusion Detection System Using Filter Based Feature Selection algorithm
LoadDataSet
Preprocessing
Feature Selection
Classification

```
Loading DataSet ....

0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.0
0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.0
0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.0
0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.0
0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.0
0,udp,domain_u,SF,29,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0.00,0.00,0.0
0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.0
0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.0
0,tcp,http,SF,223,185,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,4,4,0.00,0.00,0.00,0.
0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.0
0,tcp,http,SF,230,260,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,19,0.00,0.00,0.00,0
0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.0
0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.0
1,tcp,smtp,SF,3170,329,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,2,0.00,0.00,0.00,
0,tcp,http,SF,297,13787,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,
0,tcp,http,SF,291,3542,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,12,12,0.00,0.00,0.0
0,tcp,http,SF,295,753,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,21,22,0.00,0.00,0.00,
0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.0
0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.0
```

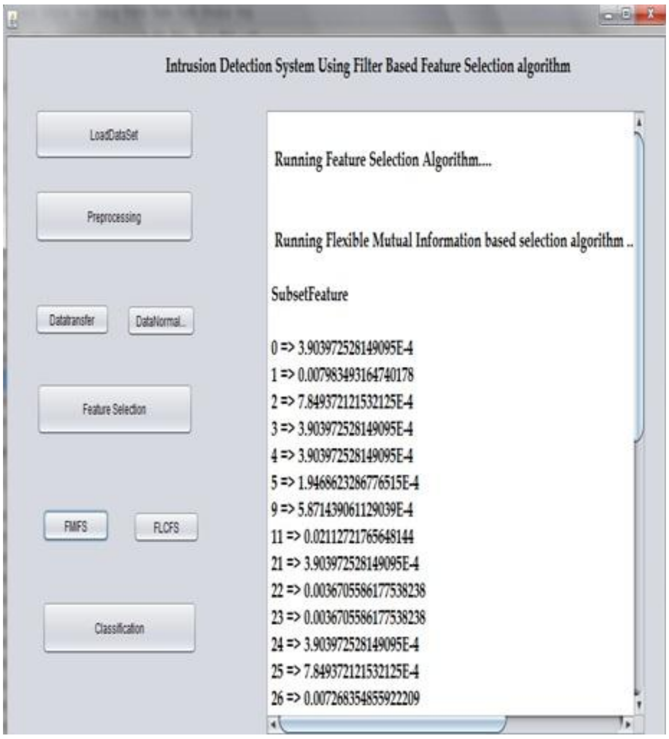
Fig 2: Loading the data set



```
Starting Preprocessing ....


Data Transfering ....

0,2,1,2,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00
0,2,1,2,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00
0,2,1,2,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00
0,2,1,2,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00
0,2,1,2,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00
0,2,7,2,29,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0.00,0.00,0.00,0.00,0.50,1.0
0,2,1,2,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00
0,2,1,2,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00
0,1,3,2,223,185,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,4,4,0.00,0.00,0.00,0.00,1.00
0,2,1,2,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00
0,1,3,2,230,260,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,19,0.00,0.00,0.00,0.00,1.0
0,2,1,2,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00
0,2,1,2,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00
1,1,4,2,3170,329,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,2,0.00,0.00,0.00,0.00,1.0
0,1,3,2,297,13787,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.
0,1,3,2,291,3542,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,12,12,0.00,0.00,0.00,0.00,1
```

Fig 3: Data Preprocessing

In the above figure 3 there are two steps data transferring and data normalization. In data transformation step we need to change all the alphabetical values into numerical values because we cant compare the alphabetical values with numeric. For example if it is tcp change it 1 and if it is udp change to 1. There are 3 types of normalization step. In first one we delete all the duplicate data and unwanted data. In second step we generalize the some of the field values. In third step we put zero for all the field which do not contain any value or which are empty. With help of this step comparison becomes easy.

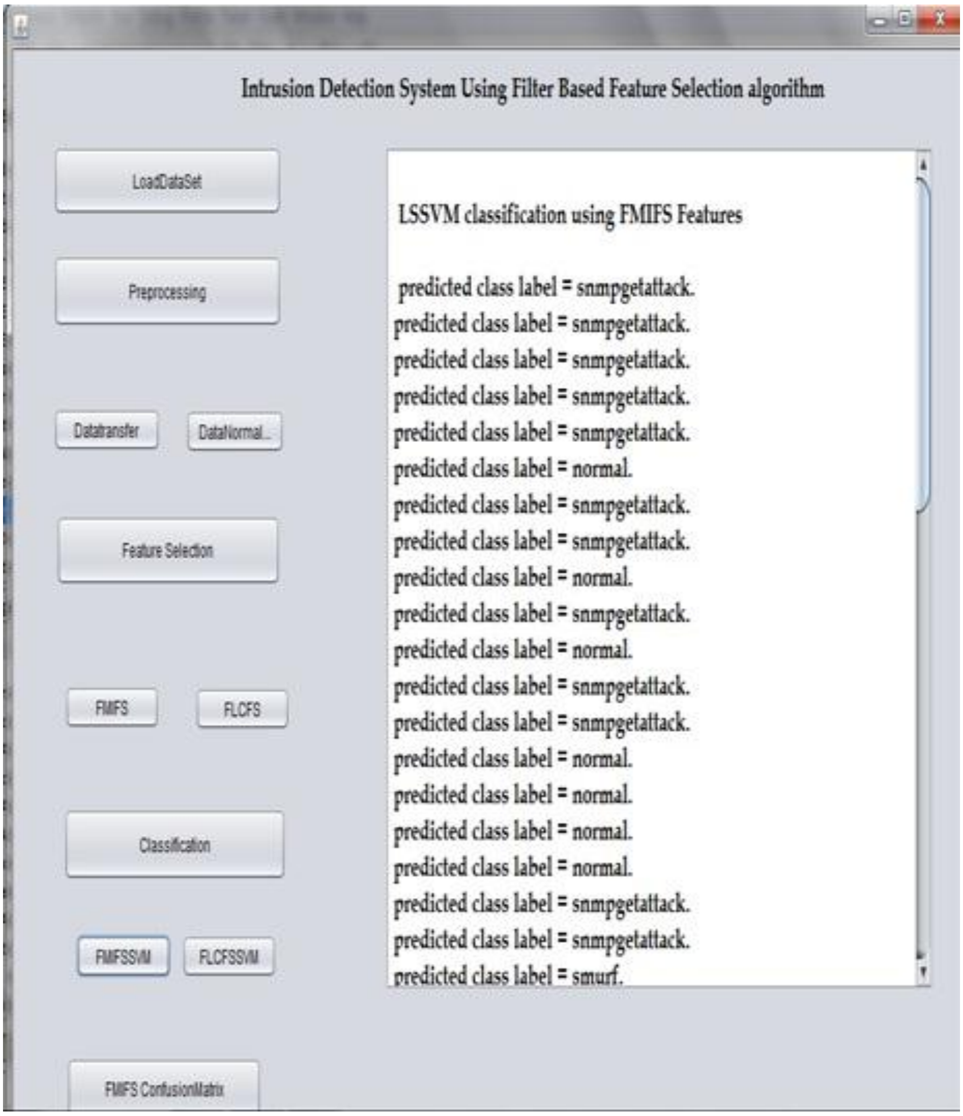


Intrusion Detection System Using Filter Based Feature Selection algorithm

```
Running Feature Selection Algorithm....

Running Flexible Mutual Information based selection algorithm ..

SubsetFeature

0 => 3.903972528149095E-4
1 => 0.007983493164740178
2 => 7.849372121532125E-4
3 => 3.903972528149095E-4
4 => 3.903972528149095E-4
5 => 1.9468623286776515E-4
9 => 5.871439061129039E-4
11 => 0.021127217656481 44
21 => 3.903972528149095E-4
22 => 0.0036705586177538238
23 => 0.0036705586177538238
24 => 3.903972528149095E-4
25 => 7.849372121532125E-4
26 => 0.007268354855922209
```

Fig 4: Feature selection

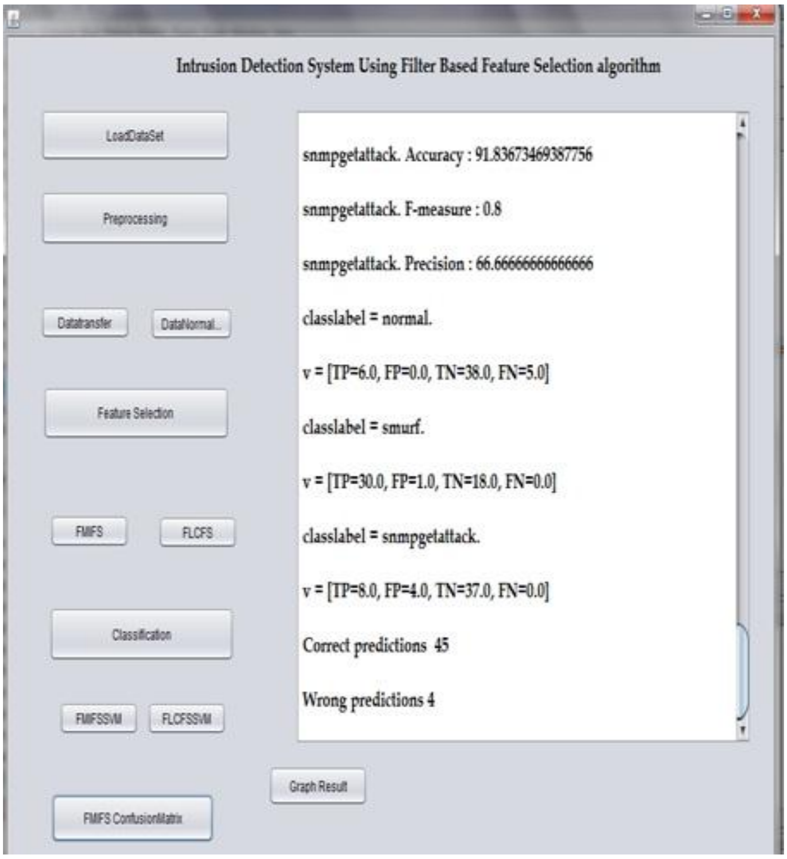
In feature selection types there are 2 types flexible mutual information based feature selection and feature selection based on the linear correlation coefficient. In flexible mutual information based feature selection we select the features based on the mutual information with the help of mathematical formula. In this step of feature selection we select the sub feature from already selected features and reduce the no of features which helps us to classify further. In order to demonstrate the flexibility and effectiveness of flexible mutual information based feature selection against feature selection based on linear dependence measure we use Feature selection based on the linear correlation coefficient and it is one of the measures used to find the relationship between two random variables.



```
LSSVM classification using FMIFS Features

predicted class label = snmpgetattack.
predicted class label = snmpgetattack.
predicted class label = snmpgetattack.
predicted class label = snmpgetattack.
predicted class label = snmpgetattack.
predicted class label = normal.
predicted class label = snmpgetattack.
predicted class label = snmpgetattack.
predicted class label = normal.
predicted class label = snmpgetattack.
predicted class label = normal.
predicted class label = snmpgetattack.
predicted class label = snmpgetattack.
predicted class label = normal.
predicted class label = normal.
predicted class label = normal.
predicted class label = normal.
predicted class label = snmpgetattack.
predicted class label = snmpgetattack.
predicted class label = smurf.
```

Fig 5: The GUI of the LSSVM classifier labeling based on the feature selection based on the mutual information

The above figure 5 consists of the class labeling based on the attack by using the algorithm. Feature selection based on the mutual information between the two random variables. By using this algorithm we get features and these features are labeled based on mutual information. The above figure 5 mainly focuses on the class labeling based on the feature selection based on the correlation .This the next step of selecting the features based on the mutual information here many features will be taken reducing these no of features we use this algorithm and the reduced no of features we have got from this algorithm we classify the attack based on the label.



```
snmpgetattack. Accuracy : 91.83673469387756

snmpgetattack. F-measure : 0.8

snmpgetattack. Precision : 66.66666666666666

classlabel = normal.

v = [TP=6.0, FP=0.0, TN=38.0, FN=5.0]

classlabel = smurf.

v = [TP=30.0, FP=1.0, TN=18.0, FN=0.0]

classlabel = snmpgetattack.

v = [TP=8.0, FP=4.0, TN=37.0, FN=0.0]

Correct predictions  45

Wrong predictions 4
```

Fig 6: GUI of the output.

This fig 6 describes the final GUI of the overall system which consists of real class label, predicted class label as well which in turn consists of the type of the attack the data has gone through .Then it consists of the class label attack here example the attack is smurf which is nothing but distributed denial of service attack .The accuracy ,F-measure, precision of the respective attack and these three parameters are calculated for snmp get attack and for normal class also thus the efficiency can be calculated. There are many formulas and equations used to calculate the accuracy ,F-measure and precision and moreover all these parameter are calculated with the help of the small parameter like TP(true positive),TN(true negative),FP(false positive),FN(false negative) whose specification is mentioned further.

The above fig 7 contains the graphical representation of the output obtained from this system. The above graph is a bar chart that is plotted between the detection rate and attack type like DOS, R2L and also the normal class. Here the prediction of the detection for different types of attack are pre calculated using trained data and that is compared with the output of the graph obtained from our system. Thus the accuracy of our system can be obtained.

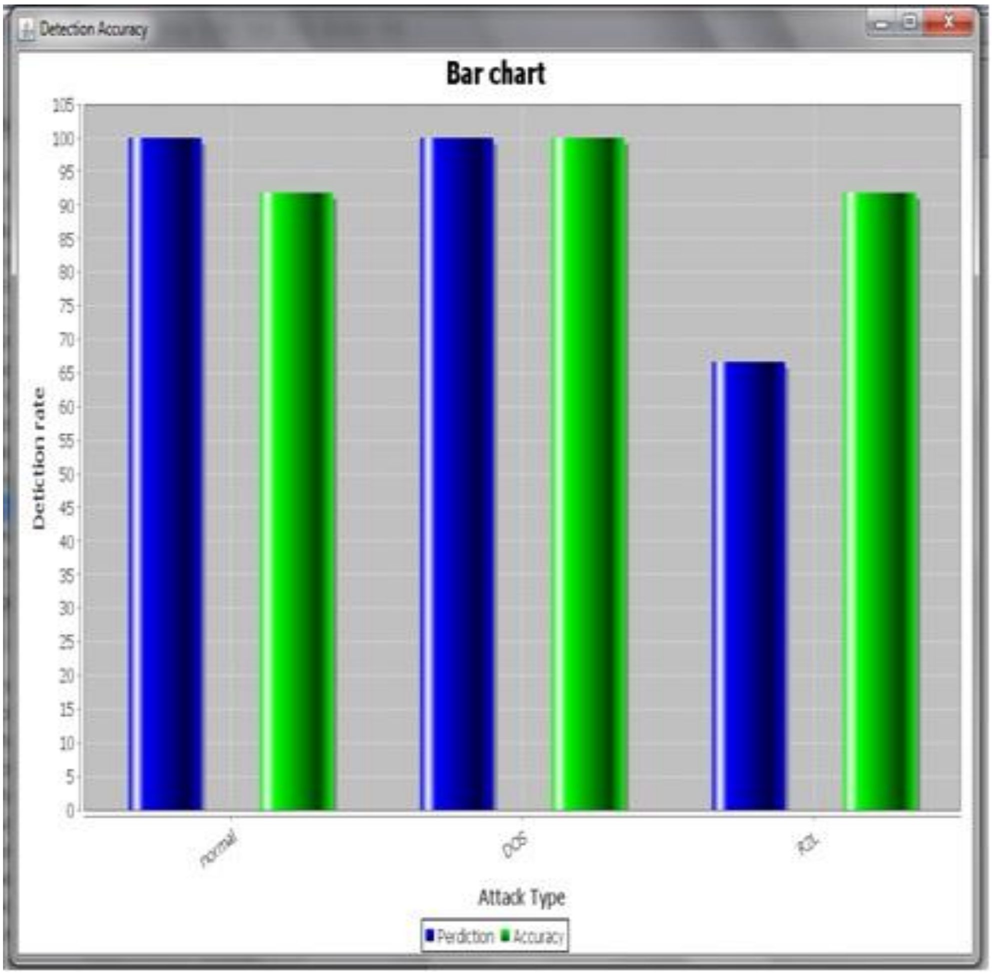


Fig 7: Graphical representation of the output.

V. CONCLUSION

The two main components to build an IDS are robust classification and feature selection .As proposed an algorithm namely Flexible Mutual Information Feature Selection (FMIFS) supervised by Filter Based feature selection algorithm .FMIFS modifies the Battitis algorithm which redundancy among the features and eliminated the redundancy used in MIFS and MMIFS. There is no pre described procedure to select value. FIMS+LSSVM is used to build an IDS. The proposed LSSVM-IDS+FMIFS has been evaluated here with the help of KDDCUP 99 data set .But we get many other datasets like NSL-KDD and Kyoto 2006+datasets for evaluation The corrected set of data of KDD cup 99 data set are tested on normal, DOS and probe classes .The performance is evaluated in the term of accuracy, detection rate, False positive and F-measure. Finally, based on the experimental results achieved on KDD CUP 99.So the result of the system is achieved promising performance in detecting intrusions in the network.

ACKNOWLEDGEMENTS

The authors would like to express sincere thanks for encouragement and constant support provided by the Management RRGI, Dr. Balakrishna, Principal; Mr. Ananda Kumar, HOD; Dept of ISE, RajaRajeswari College of Engineering, Bangalore-74, India during this research work.

REFERENCES

[1] S. Pontarelli, G. Bianchi, and S. Teofili, “Traffic-aware design of a high-speed FPGA network intrusion detection system,” IEEE Trans. Comput., vol. 62, no. 11, pp. 2322–2334, Nov. 2013.

[2] B. P fahringer, “Winning the KDD99 classification cup: Bagged boosting,” SIGKDD Explorations Newslett., vol. 1, no. 2, pp. 65–66, 2000.

[3] S. Mukkamala, A. H. Sung, and A. Abraham, “Intrusion detectionusing an ensemble of intelligent paradigms,” J. Netw. Comput.Appl., vol. 28, no. 2, pp. 167–182, 2005.

[4] C. Grosan, C. Martin-Vide, A. Abraham, "Evolutionary Design of Intrusion Detection Programs", International Journal of Network Security, vol. 4, pp. 328-339, 2007.

[5] A. N. Toosi and M. Kahani, “A new approach to intrusion detectionbased on an evolutionary soft computing model using neurofuzzyclassifiers,” Comput. Commun., vol. 30, no. 10, pp. 2201–2212, 2007.

[6] Z. Tan, A. Jamdagni, X. He, P. Nanda, L. R. Ping Ren, and J. Hu,“Detection of denial-of-service attacks based on computer visiontechniques,” IEEE Trans. Comput., vol. 64, no. 9, pp. 2519–2533,Sep. 2015.

[7] A. M. Ambusaidi, X. He, and P. Nanda, “Unsupervised featureselection method for intrusion detection system,” in Proc. Int.Conf. Trust, Security Privacy Comput. Commun., 2015, pp. 295–301.

[8] A. M. Ambusaidi, X. He, Z. Tan, P. Nanda, L. F. Lu, and T. U.Nagar, “A novel feature selection approach for intrusion detectiondata classification,” in Proc. Int. Conf. Trust, Security Privacy Comput.Commun., 2014, pp. 82–89.