IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 7, July 2017

Analysis of Classification Techniques for Intrusion Detection System

Divya M S[1], Vinutha H P[2]

M.Tech Student, Computer Science and Engineering, BIET Davangere[1]

Assistant Professor, Computer Science and Engineering, BIET Davangere[2]

Abstract: Duplicate and unimportant features exist in dataset will cause a long-term problem in classification of network traffic. The existing duplicate features not only reduce the processing speed of classification but they also prevent the classifier from classifying the data, and also losses the trust of providing accurate decisions especially when working with huge collection of data. By considering all these drawbacks a novel system is designed, this system uses two algorithms FMIFS and FLCFS for feature selection and for the classification of data. Here KDD Cup 99 dataset is used for selecting and classifying of dataset. The LS-SVM classification algorithm is used by the two algorithms and it is evaluated for KDD dataset. The evaluation result shows the most relevant features for classification of the dataset and classifies the dataset by sorting out normal data and attacked data.

Keywords: Intrusion Detection System, FMIFS, FLCFS, Feature Selection, Classification, LS-SVM.

I. INTRODUCTION

Despite increasing awareness of network security, the existing solutions remain incapable of fully protecting internet applications and computer networks against the threats from ever-advancing cyber-attack techniques such as DoS attack and computer malware. Developing effective and adaptive security approaches, therefore, it has become more critical than ever before. The traditional security techniques, as the first line of security defence, such as user authentication, firewall and data encryption, are insufficient to fully cover the entire landscape of network security while facing challenges from ever-evolving intrusion skills and techniques. Hence, another line of security defence is highly recommended, such as Intrusion Detection System (IDS). Recently, an IDS alongside with anti-virus software has become an important complement to the security infrastructure of most organizations. The combination of these two lines provides a more comprehensive defence against those threats and enhances network security.

A significant amount of research has been conducted to develop intelligent intrusion detection techniques, which help achieve better network security. Bagged boosting-based on C5 decision trees and Kernel Miner are two of the earliest attempts to build intrusion detection schemes. Methods proposed in and have successfully applied machine learning techniques, such as Support Vector Machine (SVM), to classify network traffic patterns that do not match normal network traffic. Both systems were equipped with five distinct classifiers to detect normal traffic and four different types of attacks (i.e., DoS, probing, U2R and R2L). Experimental results show the effectiveness and robustness of using SVM in IDS.

However, current network traffic data, which are often huge in size, present a major challenge to IDSs. These “big data” slow down the entire detection process and may lead to unsatisfactory classification accuracy due to the computational difficulties in handling such data. Classifying a huge amount of data usually causes many mathematical difficulties which then lead to higher computational complexity. As a well-known intrusion evaluation dataset, KDD Cup 99 dataset is a typical example of large-scale datasets. This dataset consists of more than five million of training samples and two million of testing samples respectively. Such a large scale dataset retards the building and testing processes of a classifier, or makes the classifier unable to perform due to system failures caused by insufficient memory. Furthermore, large-scale datasets usually contain noisy, redundant, or uninformative features which present critical challenges to knowledge discovery and data modelling.

II. PROBLEM STATEMENT

Providing safety to the network data is become a day to day difficult problem. Detecting as well as employing a luminary safety methodologies is becomes a serious point. A replacement and ridiculous feature exist in dataset decreases the enactment and affects the dangerous problem in network traffic. It will also replicates on classifier when creating the exact conclusions, primarily when replicating vast amount of data. Network traffic analysis for intrusion detection system (IDS) delivers a finest classification algorithm which is benefit for intrusion detection system to detect the threats in the dataset.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering

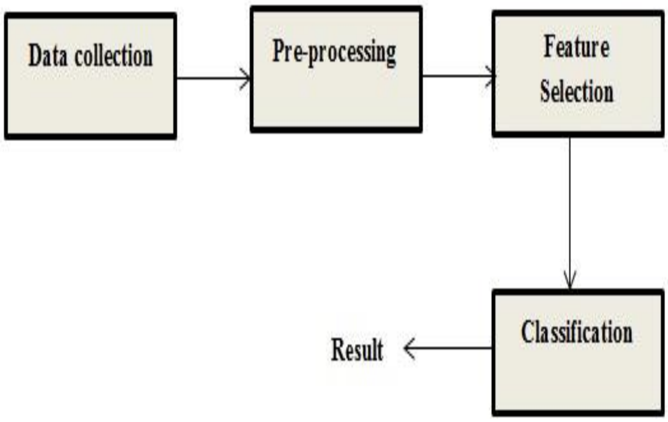
ISO 3297:2007 Certified

Vol. 6, Issue 7, July 2017

III. PROPOSED SYSTEM

In this paper we have conduct complete experiments on KDD Cup 99 IDS dataset in addition to the dataset used. And uses two feature selection algorithms for selecting and classifying the dataset. The result is helpful for the IDS system for identifying the threats in the dataset.

IV.METHODOLOGY



Data Collection

Data collection is the first and a critical step to intrusion detection. The type of data source and the location where data is collected from are two determinate factors in the design and the effectiveness of an IDS. Here we are taking a KDD dataset

Data Pre-processing

The data obtained during the phase of data collection are first processed to generate the basic features such as the ones in KDD Cup 99 dataset. This phase contains two main phases

Data transferring

The trained classifier requires each record in the input data to be represented as a vector of real number. Thus, every symbolic feature in a dataset is first converted into a numerical value.

Data normalisation

Data normalisation is a process of scaling the value of each attribute into a well-proportioned range, so that the bias in favour of features with greater values is eliminated from the dataset.

Feature selection

Even though every connection in a dataset is represented by various features, not all of these features are needed to build IDS. Therefore, it is important to identify the most informative features of traffic data to achieve higher performance. Here we are using two algorithms for feature selection

1. Flexible mutual information based feature selection

To remove the burden of setting an appropriate value for β as it is required in Battiti's MIFS, Kwak's MIFS-U and Amiri's MMIFS, a new variation of MIFS is proposed in this section. This new feature selection approach suggests an enhancement to the feature selection criterion involved in the computation of Step 4 of Battiti's MIFS algorithm.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 7, July 2017

Equation 1 below shows a new formulation of the feature selection criterion involved, which is intended to select a feature from an initial input feature set that maximizes I(C; fi) and minimizes the average of redundancy MRs simultaneously.

$$GMI = \underset{fi \in F}{\operatorname{argmax}} \left(I(C; fi) - \frac{1}{|S|} \sum_{fs \in S} MR \right) \quad (1)$$

where I(C; fi) is the amount of information that feature fi carries about the class C. MR, in, is the relative minimum Redundancy of feature fi against feature fs and is defined

$$MR = \frac{I(fi; fs)}{I(C; fi)} \quad (2)$$

GMI has the following properties:

1) If (GMI = 0), then the current feature fi is irrelevant or unimportant to the output C because it cannot provide any additional information to the classification after selecting the subset S of features. Thus, the current candidate fi is removed from S.

2) If (GMI > 0), then the current feature fi is relevant or important to the output C because it can provide some additional information to the classification after selecting the subset S of the feature. Thus, the current candidate fi is added into S.

3) If (GMI < 0), then the current feature fi is redundant to the output C because it can cause reduction in the amount of MI between the selected subset S and the output C. It is worth noting that the second term in Equation 1, which measures the redundancy among features, is larger than the first term, which measures the relevance between feature fi and the output class. Thus, feature fi is removed from S.

The selection process of FMIFS is demonstrated in Algorithm below.

Algorithm 1: Flexible mutual information based feature Selection

Input: Feature set F = { fi, i=1…n}
Output: S **-** the selected feature subset
Begin
Step1. Initialization: set S = ø
Step2. Calculate I(C; fi) for each feature, i=,……, n
Step3. nf = n; Select the feature fi such that:
$\underset{fi}{\operatorname{argmax}}(I(C; fi))$, i = 1,……, nf ,
Then, set F $\leftarrow$ F \ {fi }; S $\leftarrow$ S U {fi }; nf = nf -1.
Step4. while F ≠ ø **do**
Calculate GMI in (4) to find fi where i € {1,2, ..., ; nf};
nf = nf **-** 1;
F $\leftarrow$ F\{fi};
if (GMI > 0) **then**
S $\leftarrow$ S U{ fi}.
end
end
Step 5. Sort S according to the value of GMI of each selected feature.
return S

2. Feature Selection Based on Linear Correlation Coefficient

In order to demonstrate the flexibility and effectiveness of FMIFS against feature selection based on linear dependence measure, we substitute MI by Linear Correlation Coefficient (LCC) in Algorithm 1. LCC is one of the most popular dependence measures evaluating the relationship between two random variables. Whilst LCC is fast and accurate in measuring the correlation between random linearly dependent variables, it is insensitive to nonlinear correlations. Given the two same random variables U and V of the same type, the correlation coefficient between these two variables is defined in Equation 3 Below.

$$corr(X;Y) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}} \quad (3)$$

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 7, July 2017

The value of an corr (U;V) falls in a definite closed interval between [-1,1] . A value close to either -1 or 1 indicates a strong relationship between the two variables. A value close to 0 infers a weak relationship between them. Algorithm 2 shows our proposed algorithm based on LCC, and this algorithm is named Flexible Linear Correlation Coefficient based Feature Selection (FLCFS). Algorithm 2 is designed to select a feature that maximizes Gcorr in Equation (4) and to eliminate irrelevant and redundant features.

$$Gcorr = \underset{fi \in F}{\arg\max} \left((corr(C;f) - \frac{1}{|s|} \sum_{fi \in S} corr(fi; fs) \quad \cancel{corr(C; fi}) \right) \quad (4)$$

The selection process of FLCFS is demonstrated in Algorithm below.

Algorithm 2: Flexible Linear Correlation Coefficient based Feature Selection

Input: Feature set F = { fi, i=1…n}
Output: S - the selected feature subset
Begin
Step1. Initialization: set S = ø
Step2. Calculate corr (C; fi) for each feature, i = 1,… n
Step3. nf = n; Select the feature fi such that:
argmax(I(C; fi)), i = 1,……, nf ,
fi
Then, set F ← F\ {fi }; S ← S U {fi }; nf = nf -1.
Step4. while F ≠ ø **do**
Calculate Gcorr in (7) to find fi where i € {1,2, ..., ; nf};
nf = nf - 1;
F ← F\{fi};
if (Gcorr > 0) **then**
S ← S U{ fi}.
end
end
Step 5. Sort S according to the value of Gcorr of each selected feature.
return S

Classifier Training

Once the optimal subset of features is selected, this subset is then taken into the classifier training phase where LS-SVM is employed. Since SVMs can only handle binary classification problems and because for KDD Cup 99 five optimal feature subsets are selected for all classes, five LS-SVM classifiers need to be employed. Each classifier distinguishes one class of records from the others.

For example the classifier of Normal class distinguishes Normal data from non-Normal (All types of attacks). The DoS class distinguishes DoS traffic from non-DoS data (including Normal, Probe, R2L and U2R instances) and so on. The five LS-SVM classifiers are then combined to build the intrusion detection model to distinguish all different classes.

Algorithm3:Intrusion0detection0based0on0LS-SVM

Input: LS-SVM Normal Classifier, selected features (normal class), an observed data item x
Output: Lx - the classification label of x
begin
Lx   classification of x with LS-SVM of Normal class
if Lx == "Normal" **then**
Return LX
else
do: Run Algorithm 4 to determine the class of attack
end
end

DOI10.17148/IJARCCE.2017.6748

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 7, July 2017

Algorithm 4: Attack classification based on LS-SVM

Input: LS-SVM Normal Classifier, selected features (normal
class), an observed data item x
Output: Lx **-** the classification label of x
begin
Lx   classification of x with LS-SVM of DoS class
if Lx==”DoS” **then**
Return LX
else
Lx   classification of x with LS-SVM of Probe class
if Lx == “Probe” **then**
Return LX
else
Lx   classification of x with LS-SVM of R2L class
if Lx == “R2L” **then**
Return LX
else
Lx == “U2R”;
Return LX
end
end
end
end

V. RESULTS

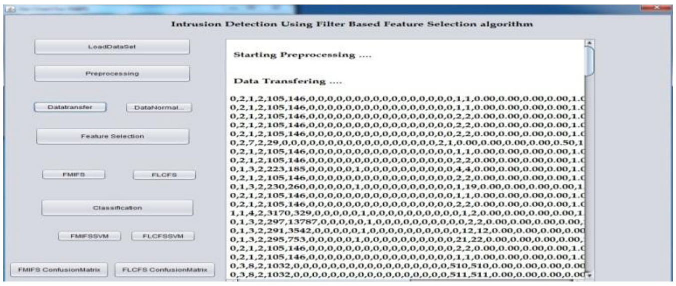


Fig (a)

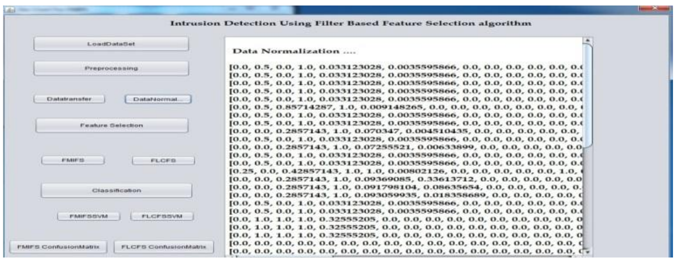


Fig (b)

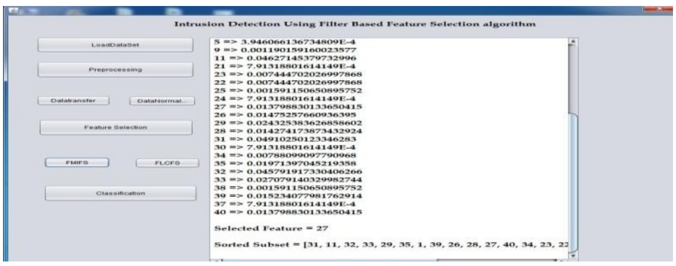


Fig (c)

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified

Vol. 6, Issue 7, July 2017

Fig (d)

Fig (e)

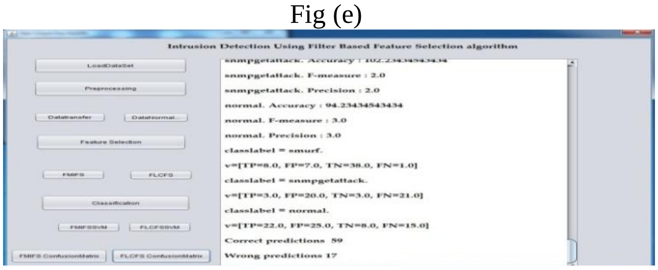


Fig (f)

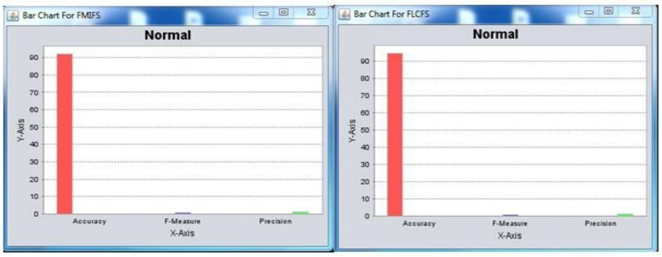


Fig (g)

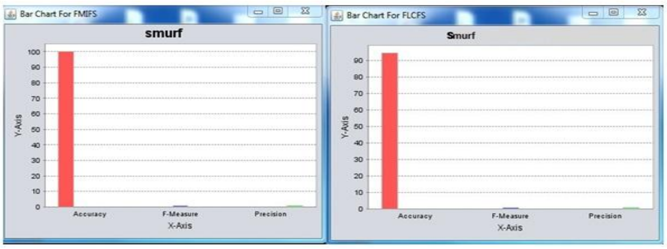


Fig (h)

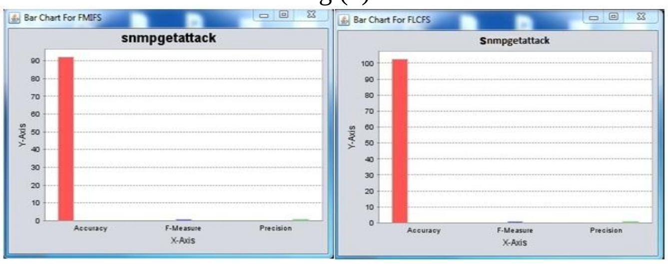


Fig (i)

Here we can observe the results of this system, fig (a) and fig (b) shows the data transferring and normalization. Fig(c) and fig (d) shows the FMIFS and FLCFS feature selection. Fig (e) and fig (f) shows the classification of data using FMIFS and FLCFS algorithm. Fig (g), fig (h) and fig (i) displays the graph showing the results.

 DOI10.17148/IJARCCE.2017.6748

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 7, July 2017

Table I: Shows results of Smurf attack classification

Smurff Attack			
	Accuracy	F-Measure	Precision
FMIFS	100.0	1.0	1.0
FLCFS	94.23	1.0	1.0

Table II: Shows results of Snmpgetattack classification

Snmpgetattack			
	Accuracy	F-Measure	Precision
FMIFS	91.83	0.8	0.66
FLCFS	102.23	2.0	2.0

Table III: Shows results of Normal classification

Normal			
	Accuracy	F-Measure	Precision
FMIFS	91.83	0.77	1.0
FLCFS	94.23	3.0	3.0

Table IV: Shows Confusion Matrix of FMIFS algorithm

	Smurf	**Snmpgetattack**	**Normal**
TP	3.0	8.0	6.0
FP	1.0	4.0	0.0
TN	18.0	37.0	38.0
FN	0.0	0.0	5.0

Table V: Shows Confusion Matrix of FLCFS algorithm

	Smurf	**Snmpgetattack**	**Normal**
TP	8.0	3.0	22.0
FP	7.0	20.0	25.0
TN	38.0	3.0	8.0
FN	1.0	21.0	15.0

VII.CONCLUSION

Proposed project for selecting and classifying the dataset will helps for the intrusion detection system. The intrusion detection system just reads the table of comparing the results of the algorithms and uses that algorithm in future work. Here in this project feature selection and classification are the two major components. FMIFS and FLCFS algorithms are used for feature selection and for classification along with LS-SVM. To conclude, the results given by the system is that the proposed system is expanded well surely performance in finding the attacks in the network attached systems. Overall FMIFS, FLCFS, LSSVM-IDS has achieved the best when matched with the extra state-of the-art models. Lastly by assessing the results of both the algorithms it is concluded that FLCFS is more efficient than FMIFS in providing the values.

REFERENCES

1. KDDcup99, "Knowledge discovery in databases DARPA archive,"1999. (https://archive.ics.uci.edu/ml/databases/kddcup99/)
2. NSL-KDD, NSL-KDD Data Set for Network-based Intrusion0Detection0Systems,Mar.02009. (http://iscx.cs.unb.ca/NSL-KDD/)
3. M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in Proceedings of the 2nd IEEE International Conference on Computational Intelligence for Security and Defense Applications, pp. 53–58, USA, 2009.
4. Z. Tan, A. Jamdagni, X. He, P. Nanda, L. R. Ping Ren, J. Hu, (2015) Anomaly based scheme for the detection of denial-of-service attacks based on computer vision techniques.
5. H. F. Eid, A. E. Hassanien, T.-h. Kim, S. Banerjee, Linear correlation-based feature selection for network intrusion detection model, in: Advances in Security of Information and Communication Networks, Vol. 381, Springer, 2013, pp. 240–248.
6. Muna M. Taher Jawhar and Monica Mehrotra, "Anomaly Intrusion Detection System using Hamming Network Approach," International Journal of Computer Science & Communication,Vol. 1, No. 1, pp. 165-169, January-June 2010.
7. A. M. Ambusaidi, X. He, P. Nanda, Unsupervised feature selection method for intrusion detection system, in: International Conference on Trust, Security and Privacy in Computing and Communications, IEEE, 2015.
8. Mohammed A. Ambusaidi, Member, IEEE, Xiangjian He*, Senior Member, IEEE, Priyadarsi Nanda, Senior Member, IEEE, and Zhiyuan Tan, Member, IEEE, "Building an intrusion detection system using a filter-based feature selection algorithm", IEEE TRANSACTIONS ON COMPUTERS,2016.