ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

Volume 3, Special Issue 3, March 2014

2014 International Conference on Innovations in Engineering and Technology (ICIET'14)
On 21st & 22nd March Organized by

K.L.N. College of Engineering, Madurai, Tamil Nadu, India

Diversified Intrusion Detection with Various Detection Methodologies Using Sensor Fusion

K.Saleem Malik Raja, K.JeyaKumar

Department of ECE, Kamaraj College of Engineering and Technology, Virudhunagar, India

Department of ECE, Kamaraj College of Engineering and Technology, Virudhunagar, India

ABSTRACT— Various Intrusion Detection System (IDS) in the literature have shown that multiple classifier may be well versed in detecting the specific attack, but detecting different types of attack is low. In order to ensure high security this work focuses on multiple classifier fusion technique to increase detection rate. The primary role of classifier is to classify the correct and incorrect instance therefore multiple classifier design that is practical, and detects more attack by means of combining them is preferred here. To our best knowledge, this is the first design that considers multiple classifier in which all classifiers are different that detects both anomalies based and misuse based attacks. The dataset collected in a networking environment with the relatively high data density may contain attacks that assaults the system and thus violates system security. In this paper the operation of combining multiple classifiers that detects all categories of attack, from that improving the detection rate and true positive rate thereby reducing the false positive rate can be done. Decision based on threshold value and combining the classifiers result based on majority voting rule helps to increase the overall efficiency and accuracy in detecting the various categories of attack.

KEYWORDS— Intrusion Detection System (IDS), Fusion, Multiple Classifier, Majority Voting Rule.

I. INTRODUCTION

Intrusion detection system (IDS) is a type of security management system for computers and networks. An intrusion detection system obtains and analyzes information from various areas within a computer or to connected computers through network to identify possible security violation, which include both intrusion (attacks from outside the organization) and misuse (attacks from within the organization).

An intrusion detection system (IDS) is software that automates the intrusion detection and responds to the computer abuse. An intrusion prevention system (IPS) is the software that has all the capabilities of an intrusion detection system and can also attempt to stop possible incidents.Intrusion detection has various functions includes monitoring and analyzing both user and system activities, analyzing system configurations and vulnerabilities etc. Connection made at unusual times, repeated connection failure, unexpected change in network, unauthorized scans etc are the major signs of attack. Thus safeguarding a particular computer or computers connected to the network is becoming more difficult. Most IDS uses multiple detection methodologies [1], [2] to provide keen and accurate detection. The following are the two classes of detection methodologies they are *signature- based* and *anomaly-based detection.*

A. Signature-Based Detection

A signature is a pattern that corresponds to a known attack. Hence signature-based detection is the process of detecting already known attacks based on their signatures not in favor of observed events. Signature-based detection is very effective at detecting known attacks but largely ineffective at detecting previously unknown attacks

B. Anomaly-Based Detection

Anomaly-based detection is the process of comparing the normal behavior of the system to be protected against observed events. It results attack whenever a deviation occurs between observation at a particular instant and the normal behavioral profile. The profiles are developed by monitoring the characteristics of typical activity over a period of time. An important benefit of this method is that they can be very effective at detecting previously unknown attacks.

C. *IDS Implementation Methodologies*

The concept of Intrusion Detection System (IDS) is useful to detect, identify and track the intruders. An intrusion detection system monitors network or system activities for malicious activities or policy violations and produces reports to administrator. Thus the IDS are classified as Network based or Host based attacks. The network based attacks and host based attacks may be either misuse or anomaly based attacks. The *network based attacks* are detected from computer systems that are interconnected or intra connected whereas the *host based attacks* are detected only from a single computer system. The intrusion can be effectively detected using data mining or by using soft computing techniques. This paper focuses on detecting intrusion based on data mining methodologies. Data mining helps in to classify the attacks to measure the effectiveness of the system. The process of finding the hidden pattern in given datum is the process of Classification. It is easy to estimate the accuracy of the resulting predictive model, and to visualize fallacious predictions with the use of classification technique. The goal of classification is to accurately predict the target class for each case in the data. The following section clearly explains the related work and also the intrusion detection using data fusion that yields improved performance than other fusion methods [2]

II.RELATED WORK

Alexander Hoffmann and B Sick [1] generated meta-alerts with a setback of typically few seconds after observing the first alert belonging to a new attack instance but in detection layer when misclassification occurs then the misclassified datum reaches alert processing layer and finally reaches reaction layer, thus wrongly assigned false alert or wrongly assigned true alert got generated. This may happen when cluster wrongly split or several clusters are wrongly grouped in to one. Ciza Thomas and N Balakrishnan [2] in improvement in intrusion with advancement in sensor fusion the number of inputs given to find each type of attack class is less. Hence on increasing the number of inputs to each attack class may reduce the detection. Different classes of intrusion detection system such as signature based and anomaly based are not incorporated for the purpose of better fusion output. Kamran and Abbass et al in their work on Learning Classifier System [3] uses Genetic Algorithm to find known attack but processing time increases exponentially with a raise in the number of rules for signature generation. The use of data fusion and cost minimization is presented by Devi Parikh and Tsuhan reduces the cost of classification errors but not the error rate itself [4]. Su-Yun Wua, Ester Yen b proposed an approach to detect intrusion based on data mining algorithms [5], detecting the attacks based on number of data arrived decides the algorithm efficiency. Average percentage of attack as well as normal data using C4.5 and SVM shows consistent detection. Therefore with the help of heterogeneous IDS the detection can be improved, but with the help of two classifiers alone the performance of the IDS cannot be evaluated. Wang et al. [6] present the superiority of the data fusion technology applied to IDSs. Giacinto et al. [7] proposed an approach to intrusion detection based on the fusion of multiple classifiers where the number of classifier depends on the number of features which increases processing complexity. Giorgio Giacinto [8] Didaci et al. [9] attempt the formulation of the intrusion detection problem as a pattern recognition task using data fusion approach based on multiple classifiers. Even though many classifiers are used, in dataset preprocessing extracting the intrinsic, content and traffic feature itself is a burden and the choice of finding the corresponding classifier for corresponding feature increases intricacy. Bass [10] in their work uses fusion in distributed IDS. Thus the detection engine is evaluated using the real network traffic.

III.DIVERISIFIED IDS MODELING

The IDS is said to be effective when the detection rate is high and should possess low false positive rate, to be effective IDS should have the competence of detecting all time attacks arrived. For the purpose of detecting the new attacks different classifiers that detects all time attacks shows improvement in every stage. This paper presented here includes optimizing the IDS by that means shows improved performance than what has been proposed so far in the literature. IDS maps the input data(X) into normal or an attack. When normal datum is detected it alerts Zero (0) or it alerts One (1). Thus it is represented as

$$\text{IDS: } X \rightarrow \{0, 1\} \qquad (1)$$

A. Problem Statement

The problem definition is defined in the following steps:

- The data set used so far is DARPA / KDD cup 1999 which is an old dataset and using it for the classifiers to detect availed type of attack by IDS, it may be effective and may not be broad in detecting new type of attacks arrived. These datasets contain duplicate records and the detection improvement can be varied by removing them therefore NSL KDD is used here.
- Detecting the intrusion using data mining or other methods by using ‘n’ number of classifiers was done. Using the same classifier many times may improve in detecting same type of attack but using different classifiers to detect anomaly as well as misuse was not in use yet with high detection in each attack class.

- One of the properties of all the sensor fusion algorithms is its ability to discover new features that are not explicit in the input. In particular, it learns to represent transitional features that are helpful for learning the target function. With the increasing eventuality of cyber attacks, constructing cogent intrusion detection models with high accuracy and real-time performance are indispensable. Hence we use Data mining techniques for effective features.

B. Dataset

Dataset upload process is used to search the dataset to select and retrieve from one particular location. In this process we use NSL KDD dataset to segregate the traffic. The traffic may be normal, Denial Of Service (DOS), probe, User to Root (U2R) and also Root to Local (R2L). After getting the dataset we have to know the traffic details about the attacks. The length of the data can be measured by using the rows in each dataset. The probability truth value is normally zero. If the value can be change we have to decide particular traffic was attacked. The input dataset has 41 features having class appended as a next feature. Hence the whole input data set is represented as $X= \{x_1^j, x_2^j, x_3^j \ldots x_n^j\}$ Where x_1, x_2 , x_n represents the number of records present and j represents the feature of each input record. The NSL KDD data set is downloaded and separated according to each layer. So we test the instance of NSL KDD Dataset to find the improvement in detection rate that is because of training it before.

C. Modeling the Classifier

Intrusion detection naturally lends itself into a data-fusion scenario where it is beneficial to combine information from multiple sources. Our proposed algorithm to do so, here we are classifying attacks into four major categories such as Probe, DOS, U2R, R2L. This is taken based on the attacks that arrived in network. The proposed system architecture shown in Figure 1 gives the detailed view of the entire detection process. At first the classifiers should able to detect the anomaly based attack and misused based attacks as well. The following Table I shows the classifier type and the type of attack it can able to detect.

TABLE I
CLASSIFIER AND ITS DETECTION TYPE

Classifier Name	Anomaly Detection	Misuse Detection
Bayes Net	Yes	Yes
IB k	No	Yes
J 48	No	Yes
SVM	Yes	No

Each of the classifiers used here can able to detect anomaly and misuse detection. Based on these criteria of detecting both the categories the different classifiers have been taken for processing. Here we use BayesNet, IBk, J48 and SVM [11] which takes the NSL KDD 20 % dataset individually to detect the four categories of attack on training and at testing the data.

BayesNet classifier learns from training data the conditional probability of each attribute M_i given the class label L given Classification is then done by applying Bayes rule to compute the probability of the particular instance of $M_1, M_2 \ldots M_n$, and then predicting the class with the highest posterior probability. This computation is possible by making a strong independence assumption all the attributes M_i are conditionally independent given the value of the class L. Independence specifies probabilistic independence, that is, M is independent of N given L whenever Pr(M/N,L) = Pr(M/L) for all possible values of M,N and L, whenever $P_r(L) > 0$.

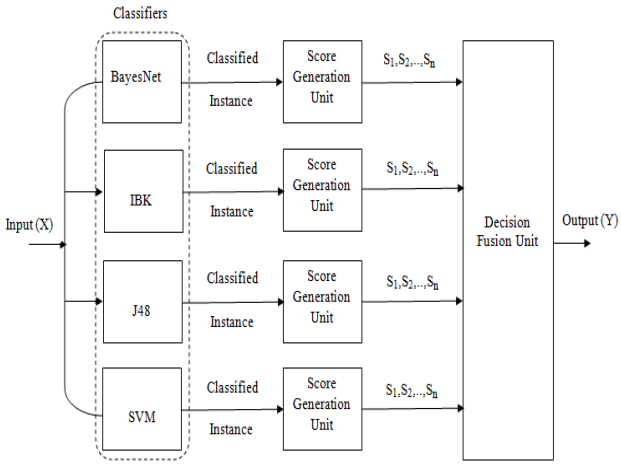

Fig. 1: Diversified Intrusion Detection Architecture

In IBk classification is carried out in two steps. The first is one of learning/training and the second is prediction / classification. For the IB k classifiers, the algorithm stores the feature vectors and class labels during training hence it's a memory based method. In the classification phase, an unlabeled vector is classified by assigning the label which is more frequent among the k training samples nearest to that query point (majority voting) which will dominate the prediction of the new vector as they tend to come up in the k nearest neighbours when the neighbours are computed due to their large number. By weighing the classification and taking the distance from the test point to each of its k nearest neighbor into account detection occurs.

A Decision Tree (J48) or a classification tree helps to learn a classification function which concludes the value of a dependent attribute (variable) given the values of the independent (input) attributes. This verifies a problem known as supervised classification because the dependent attribute and the counting of classes (values) are given. J48 is a program that creates a decision tree based on a set of labeled input data. The decision trees formed by J48 can be used for

classification and for this cause J48 is often referred to as a statistical classifier.

SVM algorithm is based on finding the hyper plane that gives the largest minimum distance to training data. This classifier clusters the data based on the features that are provided or the classes to classify each input datum. Classification is based on the help of the decision boundaries for the given number of data. Separating the hyper plane (decision boundary) can be given as

$$W * xn + b = 0 \quad (2)$$

Where $W = \{w_1, w_2, .., w_n\}$ is the Weight vector n represents the number of attributes and *b* is a scalar which is referred as bias. In the input data $X=\{x_1,x_2,\ldots,x_n\}$ $x_1,x_2,..x_n$ specifies each record input having its own attribute values and a record x_n in X is given as $x_n=\{x_1^j,x_2^j,..x_n^j\}$ having $x_1^j,x_2^j,..x_n^j$ specifies the attribute value of its record $x_{n.}$ As a result the all the classifiers will give the confusion matrix that could show the attack which was classified correctly and also the misclassified instances.

D. Score Generation

The Score generation Unit (SGU) has a major aspect of generating the original class of each input record in the dataset and its classification result after passing the classifier output to SGU. Since four classifiers generate results four confusion matrix output, by using Knowledge Flow (KF) to generate scores and also to get the actual class and classification. We are providing 42 features as input to training data including class label. Once these features are given as inputs to Score Generation Unit (SGU) then it results an extra column that specifies the classification based on distribution score. Hence the input data that is represented as X is now represented as $X=\{x_1^k,x_2^k,x_3^k\ldots x_n^k\}$ Where $x_1, x_2, x_{n,}$ represents the number of records present and k represents new column appended to the already availed feature of each input record. From this result obtained from four classifiers we have taken classes and classification label alone for making decision and fusion.

E. Effect of Setting Threshold

As the SGU that processed the classifiers output generates four files separately combining them is the tedious process. Reading the class and classification result helps to take the decision by the sensor. Class and Classification label is a string and thus converting those into binary is done which is used to perform OR operation. OR operation is done by comparing the class with the classification value. This will in turn rise to detect whether the given input is a normal or an attack. If it is normal then the representation is '0' if the given datum is an attack then '1' will be saved in log file. As a result four log files are obtained indicating the given number of records is normal or an attack in binary form. Unifying these four classifiers in a single data set using data consolidation helps to generate a file from n files. Combining is done based on Majority Voting Rule. The final output (Y) which specifies whether the individual datum is an attack or normal. In general detection is done by setting the threshold value T and is given by

$$\text{Sensor Detection} = \begin{cases} \text{Normal} & s < T \\ \text{Attack} & \text{Otherwise} \end{cases} \quad (3)$$

IV. Result And Discussion

A. Test setup

The test setup for the experimental evaluation consisted of Pentium machines with windows operating systems. For a good protection, a combination of shallow and deep sensors is necessary and for the purpose of fusing, we have incorporated data consolidation. It is important to mention that the proposed architecture can be generalized beyond the data set or the IDSs that are used in fusion. If a system is evaluated on the DARPA data set [12] [13] and KDD'99 [14] classifier learning by Elkan [15] cannot maintain anything more in the perspective of its performance on the real network traffic. Therefore NSL KDD data set can be considered as the baseline of any research. The NSL KDD [17] dataset contains the number of records. The number of records taken for training the classifier and for testing is given in Table II. With the classifiers used such as BayesNet, IBk, J48 and SVM classification is done. By giving more records during testing than number of record input during training helps in evaluating the IDS.

TABLE II
NSL KDD Input Dataset

Attack Type	Number of inputs taken for training	Number of inputs taken for testing
DOS	2625	4340
NORMAL	2152	2152
PROBE	1097	2402
R2L	2196	2734
U2R	35	67
Total	**8105**	**11695**

B. Evaluation Metrics

Let TP is the number of attacks correctly predicted as an attack, FN is the number of attacks that are detected as normal, TN is the number of normal traffic packet / connections that are correctly classified as normal and FP be the number of normal traffic packet/connections that are incorrectly detected as an attack. The commonly used IDS evaluation metrics are the precision and recall. From the attack classes what fraction of test data actually detected as an attack is the precision or specificity measure and it is given by

$$\text{Precision} = TP / (TP + FP) \quad (4)$$

Recall or Sensitivity is the measure of what fraction of attack class was correctly detected and is given by

$$\text{Recall} = TP / (TP + FN) \quad (5)$$

There exists some tradeoff between the precision and recall as the number of detection increases the recall will increase and the precision expected to get decrease and this is based on threshold value. F-Score is the harmonic mean of recall and precision, it's a score that balance between the precision (P) and recall (R) and it is given by

$$F - \text{Score} = (2 * P * R) / (P + R) \quad (6)$$

Overall Accuracy of IDS can be evaluated by the help of TP FP TN and FN and it is calculated in Table III, it is given as

$$\text{Overall Accuracy} = (TP + TN) / (TP + FP + TN + FN) \quad (7)$$

Thus in diversified intrusion detection after fusion for various threshold (T) the true positive, true negative, false positive, false negative after training and testing results as shown in the Table IV and Table V respectively . Another dimension in evaluating the performance of the classifier can be viewed as detecting the normal as a normal record. Detecting an attack as another type of attack not as an exact type will result that the attack is classified as an attack. Hence the individual classifiers TN rate (in percentage) is calculated here.

TABLE III
OVERALL ACCURACY OF DIVERSIFIED IDS WITH FUSION

After fusion having	Train Dataset	Test Dataset
Threshold T >= 2	0.9721	0.9546
Threshold T > 2	0.9976	0.9930

TABLE IV
TP FP TN FN RESULT AFTER FUSION OF TRAIN DATASET

Train Dataset				
	TP	**FP**	**FN**	**TN**
At T>=2	0.9288	0.0497	0.0041	0.9502
At T>2	0.9829	0.0018	0.0027	0.9981

TABLE V
TP FP TN FN RESULT AFTER FUSION OF TEST DATASET

Test Dataset				
	TP	**FP**	**FN**	**TN**
At T>=2	0.9412	0.0803	0.0080	0.9160
At T>2	0.9765	0.0088	0.0050	0.9911

Each classifier when given input will classify the type of attacks based on the attributes of each type of class. By means of fusion with varying threshold the evaluation metrics measure can be improved and below Table VI and Table VII is the proof for better result than the individual classifiers result.

TABLE VI
TRAIN DATA WEIGHTED AVERAGE OF EVALUATION METRICS

Classifier	TP Rate	FP Rate	Precision	Recall	F Score
BayesNet	0.957	0.010	0.965	0.957	0.960
IBk	1.000	0.000	1.000	1.000	1.000
J48	0.994	0.002	0.994	0.994	0.994
SVM	0.943	0.019	0.943	0.942	0.942
T>=2	0.928	0.049	0.949	0.995	0.967
T>2	0.982	0.001	0.998	0.997	0.997

TABLE VII
TEST DATA WEIGHTED AVERAGE OF EVALUATION METRICS

Classifier	TP Rate	FP Rate	Precision	Recall	F Score
BayesNet	0.919	0.018	0.931	0.919	0.922
IBk	0.996	0.001	0.996	0.996	0.996
J48	0.985	0.004	0.985	0.985	0.985
SVM	0.929	0.024	0.927	0.929	0.926
T>=2	0.941	0.080	0.921	0.991	0.954
T>2	0.976	0.008	0.991	0.994	0.992

The following is the comparison chart of each classifiers evaluation metrics. Each bar in Figure 2 and in Figure 3 from top to bottom represents F Score, Recall, Precision, FP Rate and TP Rate respectively.

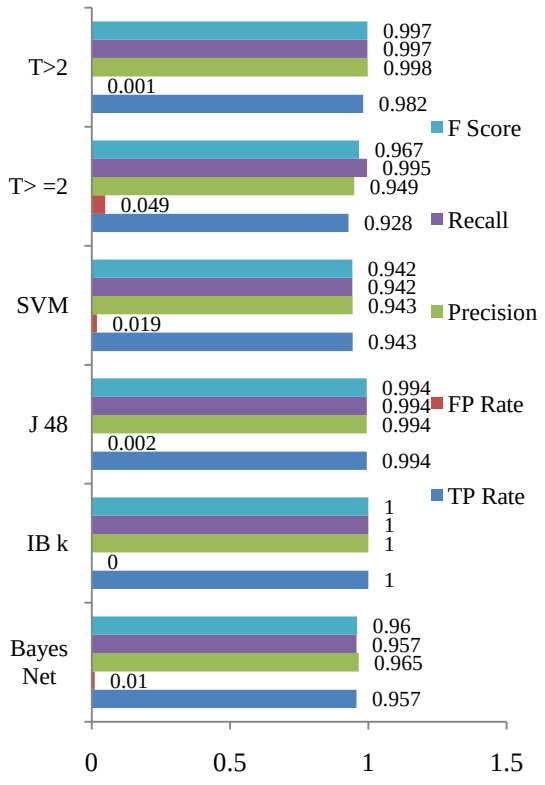

Fig. 2: Weighted Average of the Classifiers Train dataset

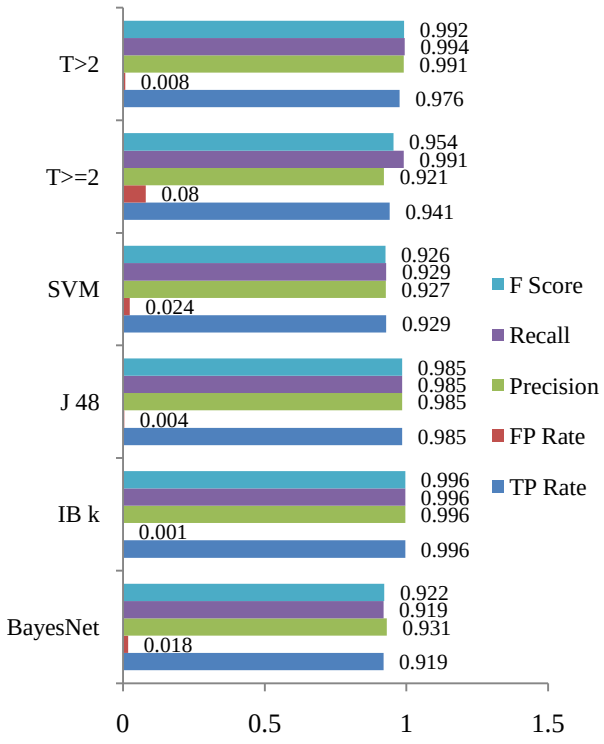

Fig. 3: Weighted Average of the Classifiers Test dataset

The Table VIII shows that the TN rate on combing and decision making based on the threshold yields improvement than averaging the four classifiers TN result. In Table IX detection rate of each type of attack shows that this IDS is effective in detecting all types of attack which yields high security to the users. It can be clearly viewed that the classifier of the same type can able to detect the U2R attack with a minimum amount only [2]. Hence using fusion IDS the harmful attacks can also be detected with greater amount which shows a dramatic increase in detection rate.

TABLE VIII
NORMAL DETECTION RATE BEFORE AND AFTER CONSOLIDATION

Classifier	Training Data	Test Data
Bayes Net	89.63%	87.96%
IBk	100.0%	100.0%
J48	99.62%	96.93%
SVM	86.29%	72.58%
With T>=2	95.02%	91.96%
With T>2	99.81%	99.11%

Examining the each attack class input detection rate before consolidation with equal number of record input during training and at testing shown in Table IX and Table X. It is clearly viewed that averaging the four classifier results can even give a low detection rate of correctly classified instances which is improved using data fusion shown in Table XI

TABLE IX
EACH ATTACK CLASS WITH NORMAL DATA AT TRAINIING

Class Label	SVM	IBk	J48	BayesNet	Average detection
DOS	99.04	100	99.50	97.63	99.04
Normal	94.42	100	99.90	97.95	98.07
Probe	97.26	100	98.81	98.81	98.69
Normal	94.42	100	100.0	96.18	97.65
R2L	95.53	100	99.59	99.04	98.54
Normal	97.67	100	99.62	95.16	98.11
U2R	65.71	100	68.57	82.85	79.28
Normal	99.76	100	100.0	96.23	99.00

TABLE X
EACH ATTACK CLASS WITH NORMAL DATA AT TESTING

Class Label	SVM	IBk	J48	BayesNet	Average detection
DOS	99.28	100.0	99.74	97.44	99.11
Normal	90.61	100.0	99.53	96.56	96.67
Probe	98.50	100.0	99.25	98.16	98.97
Normal	93.63	100.0	99.67	96.46	97.44
R2L	95.68	98.09	97.65	82.48	93.48
Normal	88.10	100.0	97.95	92.75	94.70
U2R	61.19	100.0	79.10	86.56	81.71
Normal	99.67	100.0	99.90	95.72	98.82

The following result shows improvement in individual type of attack, it is clearly shown that the individual classifier can able to detect the U2R attack with a percentage of 79 approximately during training and 81 during testing even after averaging the result obtained from each classifier as in Table IX and Table X. Hence using fusion IDS the amount of detection shows a dramatic increase in each type of attack class found in dataset than the existing methods. In specific improvement in the detection rate of U2R is obtained as 94% during training and 92 % during testing shown in Table XI.

TABLE XI
ATTACK OF EACH TYPE DETECTED AFTER FUSION DURING TRAINING AND TESTING

Data set	Class	Total Inputs	Correctly Detected	Detection Percentage
TRAIIN DATASET	DOS	2625	2617	99.652
	NORMAL	2152	2148	99.814
	PROBE	1097	1091	99.450
	R2L	2196	2190	99.726
	U2R	35	33	94.285
	Resultant data	**8105**	**8079**	**99.679**
TEST DATASET	DOS	4340	4330	99.769
	NORMAL	2152	2133	99.117
	PROBE	2402	2396	99.750
	R2L	2734	2695	98.573
	U2R	67	62	92.537
	Resultant data	**11695**	**11616**	**99.324**

V. CONCLUSION

We have shown that by using multiple classifiers of different type that detects both anomalies as well as misuse based attack will improve the overall accuracy of IDS. Increase in number of classifiers and setting a threshold based on the number of classifiers used will improve the detection rate. This can be done by using the majority voting rule in decision and fusion unit. It can be viewed clearly that highly malicious attacks can even be detected more than already availed IDS [2].

ACKOWLEDGEMENT

The authors highly value the suggestions of the anonymous reviewers which in turn improved the quality of the paper.

REFERENCES

[1] Alexander Hofmann and B Sick , "Online Alert Aggregation with Generative Data stream Modeling" IEEE Transaction on Dependable and secure Computing March-April 2011

[2] Ciza Thomas and N Balakrishnan "Improvement in Intrusion Detectin with advances in sensor Fusion" IEEE Transaction on Information Forensics and security - September 2009

[3] Kamran Shafi Hussain A Abbass " An adaptive genetic based signature learning system for Intrusion detection" Elsevier-Experts system with application 2009

[4] Devi Parikh Tsuhan Chen "Data Fusion and Cost minimization for Intrusion Detection "- IEEE Transactions On Information Forensics And Security, Vol. 3, No. 3, Sep 2008

[5] Su-Yun Wua, Ester Yen b, " Data mining-based intrusion detectors" Elsevier machine learning methods in intrusion detection system 2008

[6] Y. Wang, H. Yang, X. Wang, and R. Zhang, "Distributed intrusion detection system based on data fusion method," in *Intelligent Control and Automation (WCICA)*, Hangzhou, China, June. 2004.

[7] G. Giacinto, F. Roli, and L. Didaci, "Fusion of multiple classifiers for intrusion detection in computer networks," *Pattern Recognit. Lett.*, vol.24, no. 12, pp. 1795–1803, Aug. 2003

[8] Giorgio Giacinto and Fabio Roli ," Intrusion Detection in Computer Networks by Multiple Classifier Systems" IEEE Security of Computer Networks 2002

[9] L. Didaci, G. Giacinto, and F. Roli, "Intrusion detection in computernetworks by multiple classifiers systems," in *Int. Conf. Pattern Recognition*,Quebec, Canada, 2002.

[10] T. Bass, "Multisensor data fusion for next generation distributed intrusion detection systems," in *IRIS Nat. Symp.*, Laurel, MD, 1999

[11] Ozgur Depren, Murat Topallar, Emin Anarim, M. Kemal Ciliz, "An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks"Elsevier 2005

[12] J. McHugh, "Testing intrusion detection systems:Acritique of the 1998and 1999 DARPA IDS evaluations as performed by Lincoln laboratory,"*ACM Trans. Inf. Syst. Security*, Nov.2000

[13] M. V. Mahoney and P. K. Chan, An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection Tech. Rep. CS-2003-02, unpublished.

[14] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[15] C. Elkan, "Results of the KDD'99 classifier learning," *SIGKDD Explorations*, vol. 1, no. 2, pp. 62–63, Jan. 2000

[16] J. Kittler and F. Roli (eds.), *Multiple Classifier Systems*,LNCS 2096, Springer, 2001.

[17] http://nsl.cs.unb.ca/NSL-KDD/NSLKDD.html