IOSR Journal of Computer Engineering (IOSRJCE)
ISSN: 2278-0661, ISBN: 2278-8727 Volume 6, Issue 2 (Sep-Oct. 2012), PP 60-67
www.iosrjournals.org

An Approach of Automatic Data Mining Algorithm for Intrusion Detection and Prevention System

[1]Mr.A.Siles Balasingh, [2]Mr.N.Surenth
[1, 2] Asst.Professor, School of Computer Science and Engineering, St.Joseph University in Tanzania,

Abstract: *The Network security system plays a major role in the communication world, here every one's need security and assurance for their communication. Normally the hacking, Intrusion software's are using threats, malicious to enter in to the network and they are keep recording or monitoring the communication for this action the hackers may use following techniques Rexd, Rsh, Defult account/Null password, Generic Recon Mechanisms, NIS (Formerly yp), and Rpc Portmapper. The normal firewalls can address only insider threats. The main focus of our research is applied on two different platforms, the first one is processing on TCP-Dump portion using KDDCUP99 data set and mining, the second one is finding of best algorithm for data mining Intrusion on UNIX. From the available kddcup'99 dataset two subsets are taken with the record size of 997 and 11438. For UNIX User data all instances are taken. The classification rate and the false negative rate are used as the performance criteria with 3 fold cross validation. It is found that PART, SMO, Hyper pipes, Filtered Classifier, Random forest, Naïve Bayes Updateable, KStar are giving high classification rate with low false positive rate. For UNIX User data ZeroR is giving high performance. Real time data is also applied to finalize the best algorithm under each category of classifier. Later online implementation has to be done.*

I. Introduction

In order to enforce high protection in network communication there are huge number of software tools are available in the market. A few software's aimed to detect the intruder who escapes the protection of screening of IP address and some other software's attempted to stop Intruder and make alert network administrator (Figure no.1 shows). However completely removing breaches of security as they appear, at present, unrealistic. We can now try to detect intrusion attempts so that action may be taken to repair the damage later. There are many categories of network intrusions. Examples include SMTP (Send Mail) attacks, guessing passwords, IP Spoofing, buffer overflow attacks, multi scan attacks, Denial of Service such as ping-of- death, SYN flood, etc,. Intrusion Detection is the act of detecting actions that attempt to compromise the confidentiality, integrity or availability of the resource. Intrusion detection System (IDS) inspects all inbound and outbound network traffic. When intrusive activity occurs, IDS let you know about that by making an alarm. It can generate false positives or false negatives. False positive occurs when an alarm is generated for a normal activity. False negative occurs when no alarm is there for an abnormal activity. Misuse detection is different from anomaly detection under IDS categories. In misuse detection it analyzes the information that it gathers and then it compares to large databases of attack signatures. In anomaly detection it monitors network segments to compare their state to normal baseline and look for anomalies. Misuse Detection is a particularly difficult problem because the extensive vulnerabilities of computer systems and the creativity of attackers.

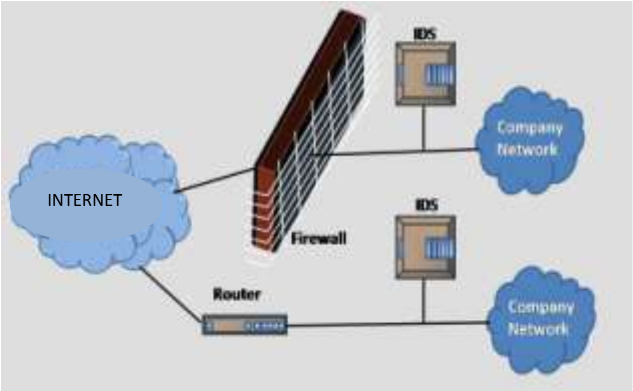


Fig No.1 shows the Intruder Detection system

Pattern matching systems such as rule-based expert systems, transition analysis and genetic algorithms (GA) are the dir works (ANN) and statistical methods and data mining techniques are used for anomaly detection. This paper applies each of the Data-Mining algorithms under each classifier category by using the weka tool which is freely available on different sets of data with 997 and 11438 records of the IDS dataset taken in our experiment normally known as KDDCUP'99 dataset. Then it applies each of the Data-Mining algorithms under different classifier categories on the Unix User data and Real time data. The performance of each Data-Mining algorithm is evaluated on each dataset. The criteria for performance evaluation that we have taken mainly are the classification rate, false positive rate alone. Based on their values under each category of classifiers the best performing algorithm is only taken into consideration. The value of classification performance for that best under each of the classifier category in Weka is only tabulated for each subset of records out of the dataset that we are using here. The rest of the paper is organized as follows; Section 2 describes the related work in detail. Section 3 deals with the datasets used .Section 4 gives the system model. Section 5 gives the implementation. Section 6 describes the results. Section 7 deals with the issues faced and finally Section 8 deals with the conclusion.

Intrusion detection Technology is an effective approach to dealing with network security. Misuse Detection uses well defined patterns of the attack that exploit weaknesses in system and application software to identify intrusions. These patterns are encoded in advance and used to detect intrusions normally. A simulation study [1] was performed to assess the performance of the comprehensive set of Data-Mining algorithms on the KDDCUP’99 intrusion detection dataset. Simulation results demonstrated that for a given classifier some algorithms performed better. Our work differs from that where for only 9 algorithms they had performed the simulation. Here we have applied accurately 70 algorithms on the same dataset which is 8 times the number that they had used. Their work would not offer much promise for detecting some type of attacks within the misuse detection context. A hybrid neural network and C4.5 model has been proposed where the neural network has been well performing for some types of attacks and C4.5 is promising for some other types of attacks. Both are combined [2] to work on their strong point alone.

An approach that uses the representation of a bag of system calls [3] in system call sequence is proposed. It has been shown by them that this representation us very suitable for well known attacks and trivially modified attacks. If the attacker is known of IDS the approach will fail as they mentioned in the paper. Artificial Neural Networks [4] are provided where stated that the most current approaches are less successful in detecting attacks and the ANN lets the limited, nonlinear, and incomplete data sources to be identified and classified. The approach uses the analytical strengths of neural networks as they mentioned. But normally the classification by neural networks will take larger amount of time for larger dataset. The performance and accuracy is also low.

A Genetic Algorithm based approach [5] is presented for generating efficient Rules for Misuse Detection. It involved only five relevant features for each attack category for that purpose. It incorporates different costs for misclassifying attacks in its fitness function to yield rules that are cost sensitive. The resulting rules generate an alert about an attack and its category. The rules generated are too short and amenable to Misuse Detection System (MDS).The tabulated results for 50 iterations with 200 rules covering all attacks and five iterations on DoS attacks highlighted the advantage of reduced and relevant feature set..

II. Dataset Used

The KDDCUP’99 IDS dataset is used in our work. This dataset is the only publicly available bench mark dataset for intrusion detection. It was created using simulated traffic and attacks. It is used in the “3rd International Knowledge Discovery and Data mining tools Competition” in conjunction with “KDD’99 The 5th International Conference on Knowledge Discovery and Data mining” where the task is to build a network intrusion model capable of distinguishing between “bad” connections, called intrusions or attacks and “good” normal connections. This contains a standard set of data to be audited, including a wide variety of intrusions simulated in a military network environment. It contains nearly 40 lakh records with 42 attributes. The right most attribute describe the “good” or “bad” behavior of each of the records. Based on these categorical attributes, each record in the 10% corrected dataset of the KDDCUP’99 IDS dataset collection is classified into any one of the resultant classes.

There are totally 17 classes. The attributes in the KDDCUP’99 contains numerical as well as non- numerical values. The dataset is normally a multivariate type. Out of all input attributes 6 attributes are of non numeric type. That is they contain only text data. All other 35 are of numeric type. That means they contain numbers as their values possibly.

A Set 1

From the above mentioned dataset a subset of 997 records are chosen. These are taken as set1 in our experiment. The records for this subset are chosen randomly without repetition from the KDDCUP’99 dataset as a whole such that the subset’s records will contain instances from each of the 17 classes. For each of the records all attributes are used for classification. The overall classification process involves a minimum of 6 steps as mentioned in the section that follows.

B Set 2

From the same dataset, another subset of 11438 records is chosen. This subset is considered as set2. Here we have taken instances randomly from the main dataset. It will cover all 41 attributes but 6 different classes alone when given for a classification algorithm that is applicable for this dataset. Here too for each record all attribute values are used for classification. Then UNIX User dataset is used in our paper which contains two input attributes and one output attribute. The history and session are the two input attributes. The history attribute is numeric and the session attribute is nominal. History attribute specifies the line number with respect to the class attribute. The session attribute contains enriched command line argument given by the user in a UNIX terminal. The dataset contains 9100 instances whereas an instance will be classified as it belongs to any one of the nine users based on the two input attributes. User0, User1, User2, User3, User4, User5, User6, User7, User8, User9 are the possible values of the output attribute. Then Real time data from seven Red Hat Linux Users has been used in our paper. The data has two attributes namely the line number and the enriched command line data. Based on the command line each instances are assumed as it is given by the masquerader or a normal user.

III. SYSTEM MODEL

In this paper the Detection System is modeled using the standard available KDDcup'99 dataset and Unix User data. Two subsets of this standard version of KDDcup'99 dataset is used with 3 fold cross validation. This system model is developed with the objective of classifying the intrusions correctly with minimal false alarms. The system model is depicted in Fig. 2.

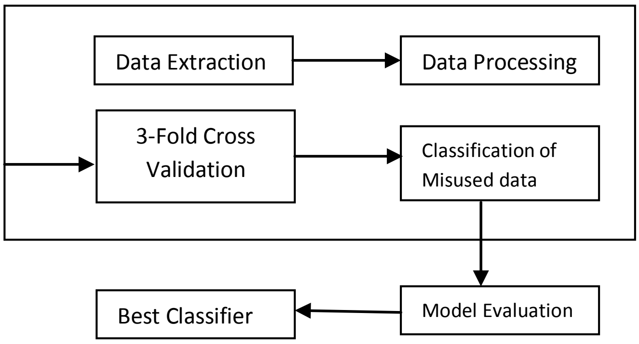


Fig.2.Misuse Classification System Model

A. Data Set

The KDDcup'99 dataset has around 3, 00,000 records with 18 classes. 17 attacks and 1 normal are represented with 42 attributes. The last column of the attributes specifies which class it belongs to either normal or attack. It is a freely available and universally applied for mainly IDS related research work. In this paper 10% corrected part of the dataset is used. Here the UNIX User dataset is used which has 9100 instances involving 3 attributes. From the Real time dataset all instances are applied totaling 1375 instances.

B. Data Extraction

From the full KDDcup'99 dataset a subset is extracted and used in this research. From the whole dataset a random subset of 997 records with all 42 attributes is selected and this subset is named as set1. Another random subset of 11,438 records with all 42 attributes is selected and this subset is named as set2. From the UNIX User dataset all instances involving all attributes are extracted. From the Real time command files all records are used.

C. Data Preprocessing

The extracted data is preprocessed such that it can be accepted as input to the classifier. The input files are prepared for both the sets of random collection. Every attribute is defined using it's data type and values for the attributes is also specified. The final input files are now ready to be fed as input to the classifier. For UNIX User data and all other data sets the files are converted into csv or arff files. For Real time data the files are separately maintained for enriched and truncated command line formats.

D. 3 - fold Cross Validation

3-fold cross validation is done for each set1 and set2 data individually. There the set1 and set2 instances are divided into so many numbers of samples and 3 samples are used for testing and the remaining samples are used for training the respective classifiers.

E. Classification of Misuse data

The input files are given to the classifier with all the attributes one by one. The classifier is chosen from the category of classifiers. The type of cross validation is specified and the dependent class is selected. The classifier evaluation options like output model, output class-stats, output entropy evaluation measures, output confusion matrix, output predictions, store predictions for visualization and cost sensitive evaluation are given. Then the classifier starts classifying the data and displays the result. The final classification will give the number of records that are correctly classified and the number of records that are incorrectly classified. The error values are displayed. The confusion matrix is also drawn. The performance measures Precision, Recall and F-Measure are calculated from the results.

1) CORRECT

There exist 2 kinds of instances. Actually the instance is said to be correctly classified if its class category is same as what is predicted by the classification algorithm.

2) INCORRECT

An instance is called an incorrectly classified instance if it is predicted as it belongs to some other class by the classification algorithm.

F. Model Evaluation

Here the built model is evaluated. In this stage several options are available in weka. The model alone can be displayed. For each class too it can be shown. Entropy evaluation measures can be seen. Confusion matrix can be generated. Predictions can be stored. These predictions may be used in visualization. Cost sensitive evaluation can be made. Predictions can be made available for display. For cost sensitive evaluation random seed has to be set and for detailed evaluation cost matrix editor can be tuned.

G. Best Classifier

The classification algorithms that are applicable on both of these subsets under each classifier category are evaluated. The per-class stats and the entropy evaluation measures related to each algorithm on each subset of the data are noted down. Based on these measures the best performing algorithm under each classifier category is tabulated. Classification rate, False Positive Rate and Time to build the model are mainly used as the performance measures for comparisons.

V. Implementation

Weka tool is used here. The version that we are using is weka-3.4.13. It contains simulation results for all the Data-Mining algorithms. Here for all 70 algorithms we have given the set1 and set2 but for only 46 algorithms the dataset is applicable. ZeroR, OneR, PART, Conjunctive Rule, Ridor , Decision Table, NNge, JRIP, Hyper Pipes, VFI, IB1, IBk , KStar, LWL, Decision Stump, J48, Rep Tree, Random Tree, User Classifier, Random Forest, LMT, NB-Tree, AdaBoostM1, Attribute Selected Classifier, Bagging, Classification Via-Regression, CV-Parameter Selection, Filtered Classifier, Decorate, Grading, Logi-Boost, Multi Boost AB, Multi-Class Classifier, Multi Scheme, Ordinal Class Classifier, RacedI ncrement Logit Boost, Random Committee, Stacking, StackingC, Vote, BayesNet, NaiveBayes,

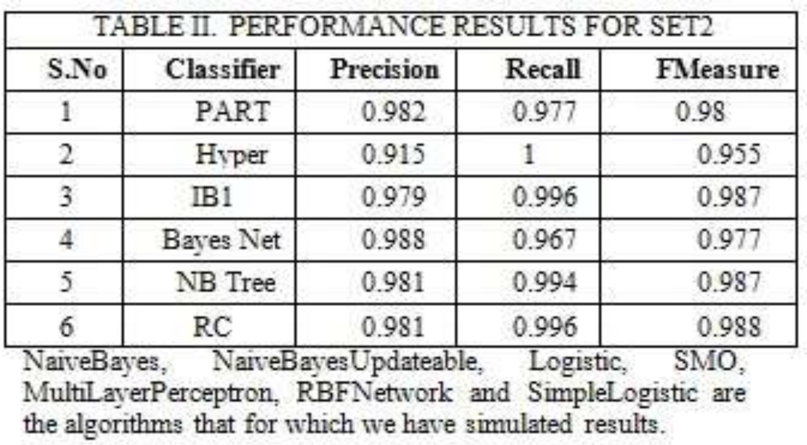


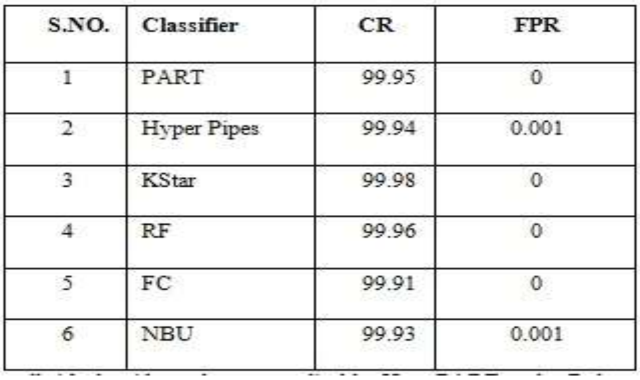
TABLE II. PERFORMANCE RESULTS FOR SET2

S.No	Classifier	Precision	Recall	FMeasure
1	PART	0.982	0.977	0.98
2	Hyper	0.915	1	0.955
3	IB1	0.979	0.996	0.987
4	Bayes Net	0.988	0.967	0.977
5	NB Tree	0.981	0.994	0.987
6	RC	0.981	0.996	0.988

NaiveBayes, NaiveBayesUpdateable, Logistic, SMO, MultiLayerPerceptron, RBFNetwork and SimpleLogistic are the algorithms that for which we have simulated results.

VI. RESULTS

First of all the set 1 data is given to the weka tool and after preprocessing, for each category the best performing algorithm is noted down. Here the data is given to each algorithm and their performance results are tabulated. Table I gives for each of the best algorithm under each classifier category their respective classification rates and false positive rates. When considering classification rate (CR) and FPR the IB1 algorithm under lazy category is performing well out of all 46 algorithms that are applicable. Here PART under Rules category and Bayes Net under Bayes category are having the same classification rates. When we consider the FPR value alone the Bayes Net under Bayes category is performing well across all 7 categories of classification algorithms. Where NB Tree under Trees category, Random Committee (RC) under Meta category and Multi Layer Perception (MLP) under Functions category are in the same level with the same False Positive Rates.



S.NO.	Classifier	CR	FPR
1	PART	99.95	0
2	Hyper Pipes	99.94	0.001
3	KStar	99.98	0
4	RF	99.96	0
5	FC	99.91	0
6	NBU	99.93	0.001

The set 2 data is given to the tool and after preprocessing, for each category the best performing algorithm is noted down. This selection is based on the classification rate (i.e. the percentage of correctly classified instances) and false alarm rate mainly. The results are shown in Table II below. PART under Rules category, Hyper Pipes under Miscellaneous category, KStar under Lazy category, Random Forest (RF) under Trees category, Filtered Classifier (FC) under Meta category, Naïve Bayes Updateable under

Bayes category and SMO under functions category are the best performing algorithms. Out of these KStar is the best from the Table II below which considers only False Positive Rate (FPR) and Classification Rate (CR) (i.e. the number of correctly classified versus incorrectly classified instances) alone. Out of all algorithms KStar under Lazy category is performing well. When we consider classification rate alone also it is performing as the best one. The SMO under functions category and Random Forest under Trees category are in the same level. If we consider only the False Positive Rate alone other than Hyper Pipes under the miscellaneous category and Naïve Bayes Updateable (NBU) under the Bayes category all 5 other best algorithms are performing well with a needed False alarm rate that is expected for a perfect intrusion detection system. From the above two tabulations we can finalize that the PART classifier under Rules category of classifiers and Hyper Pipes under Miscellaneous classifier category are performing well when we consider the classification rate and the false alarm rate i.e. false positive rate.

The False Positive Rate (FPR) is the proportions of examples that are classified as X but truly belong to different class out of those that are not belonging to class X. The Precision is the proportion of the examples which truly have class X among all those which were classified as class X. The Recall is the proportion of the examples that are classified as of class X among all those that are in class X.

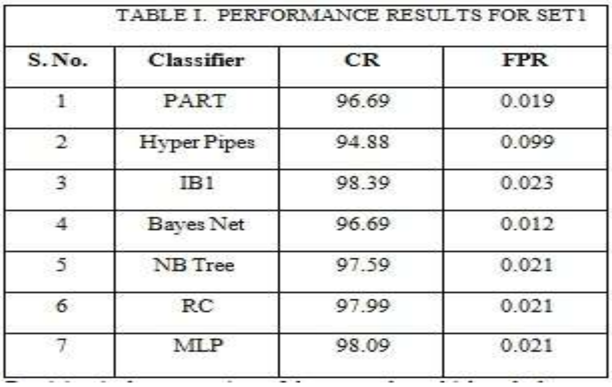


TABLE I. PERFORMANCE RESULTS FOR SET1

S. No.	Classifier	CR	FPR
1	PART	96.69	0.019
2	Hyper Pipes	94.88	0.099
3	IB1	98.39	0.023
4	Bayes Net	96.69	0.012
5	NB Tree	97.59	0.021
6	RC	97.99	0.021
7	MLP	98.09	0.021

When we consider the precision value the Bayes Net under Bayes category of classifiers is performing as the best one.

TABLE V. PERFORMANCE RESULTS FOR UNIX USER DATA

S.No	Classifier	CR	FPR
1	CVParameter Selection	26.64	0.266
2	ZeroR	50.85	0.045
3	VFI	26.64	0.266
4	User Classifier	26.64	0.266

There NB Tree of Tree category, Random Committee of Meta and Multi Layer Perceptron of Functions category of classifiers are in the same level. The F-Measure is simply 2*Precision* Recall / (Precision + Recall), as a combined measure of precision and recall. There when we consider the value of Recall the Hyper Pipes under Miscellaneous category is the best classification algorithm. Here NB Tree of Trees category and Multi Layer Perceptron of Functions category are with the same Recall value. There also IB1 of Lazy category and Random Committee of Meta category are in the same level when we consider the Recall value alone. The results for the best performing algorithms for set 1 under each category of classifier are provided with their respective values for precision, recall, measure and cost as shown in Table III as given below.

TABLE III. PERFORMANCE RESULTS FOR SET1

S.No	**Classifier**	**Precision**	**Recall**
1	PART	1	1
2	Hyper Pipes	0.99	1
3	IB1	1	1
4	Bayes Net	1	1
5	NB Tree	1	0.99
6	RC	0.99	1

When we consider the precision value the Bayes Net under Bayes category of classifiers is performing as the best one. There NB Tree of Tree category, Random Committee of Meta and Multi Layer Perceptron of Functions category of classifiers are in the same level. The F-Measure is simply 2*Precision* Recall / (Precision + Recall), as a combined measure of precision and recall. There when we consider the value of Recall the Hyper Pipes under Miscellaneous category is the best classification algorithm. Here NB Tree of Trees category and Multi Layer Perceptron of Functions category are with the same Recall value.

There also IB1 of Lazy category and Random Committee of Meta category are in the same level when we consider the Recall value alone. The results for the best performing algorithms for set 1 under each category of classifier are provided with their respective values for precision, recall, measure and cost as shown in Table III as given below. The same for the best performing algorithms under each category for the set2 is given in Table IV as below. Here the other values like Fmeasure, Precision, Recall and Cost are provided for those algorithms. Here other than Hyper Pipes of Miscellaneous category and Naïve Bayes Updateable under Bayes category are performing well when we consider the Precision values alone. When we consider the Recall values all other than the Filtered Classifier of category Meta classifiers are performing well. When we consider the Fmeasure values, all other than the Filtered Classifier of Meta category and Naïve Bayes Updateable of Bayes category of classifiers are well equipped with greatest performance.

There are also several issues that are being faced by us in our experimentation and evaluation of classifiers. The following section deals with those issues.

TABLE IV: PERFORMANCE RESULTS FOR SET 2

Finally the UNIX User dataset is given to the tool and the results are noted down. Here we found that some classifiers are not applicable on this dataset which are found to be applicable on IDS dataset in our work. Table V given below gives the result of the best classifiers that are applicable for the UNIX User dataset. With respect to the Classification Rate (CR) and False Positive Rate (FPR) we have found that ZeroR of Rules category is the best performing algorithm than the other best performing classifier algorithms in their respective categories. The VFI is the best under Miscellaneous and User Classifier is the best under trees category and CVParameter Selection is the best under Meta category.

Table VI given below gives the resultant values for other measures for these algorithms that are applied to the UNIX User dataset. Table VII given below provides the algorithms that are best performing under each classifier category for enriched command line formatted real time data. Here the OneR algorithm under Rules category of classifier is the best performing one when compared to all other category of algorithms.

TABLE VI. PERFORMANCE RESULTS FOR UNIX USER DATA

S.No	**Classifier**	**Precision**	**Recall**	**FMeasure**
1	CV Parameter Selection	0.071	0.266	0.112
2	ZeroR	0.071	0.266	0.112
3	VFI	0.68	0.509	0.541
4	User Classifier	0.071	0.266	0.112

TABLE VI. PERFORMANCE RESULTS FOR ENRICHED REAL TIME DATA

S.No	**Classifier**	**CR**	**FPR**
1	OneR	93.09	0.269
2	Hyper Pipes	92.65	0.299
3	IBK	93.07	0.270
4	HNB	93.06	0.275
5	Random Tree	93.08	0.279
6	Bagging	93.01	0.289

Table VIII given below lists the results about the high performance algorithms under each of the classifier category that are applied on truncated real time data. Here also OneR is found to be the best performing algorithm overall. When considering Classification Rate (CR) alone Bayes Net of Bayes category is found as best. When False Positive Rate (FPR) alone is considered, RBF Network of Functions category is found to be the best performing one.

TABLE VIII.
PERFORMANCE RESULTS FOR TRUNCATEDREAL TIME DATA

S.No	**Classifier**	**CR**	**FPR**
1	OneR	93.65	0.236
2	Hyper Pipes	93.01	0.348
3	IBK	93.65	0.236
4	Bayes Net	93.73	0.277
5	Random Committee	93.65	0.236
6	REP-Tree	93.73	0.288
7	RBF-Network	93.44	0.225

Table IX given below lists the other performance values of the best performing algorithms under each classifier category.

TABLE IX . PERFORMANCE RESULTS FOR ENRICHED REAL TIME DATA

S.No	**Classifier**	**CR**	**FPR**
1	OneR	93.65	0.236
2	Hyper Pipes	93.01	0.348
3	IBK	93.65	0.236
4	Bayes Net	93.73	0.277
5	Random Committee	93.65	0.236
6	REP-Tree	93.73	0.288
7	RBF-Network	93.44	0.225

Table X given below lists the other performance values of the best performing algorithms under each classifier category.

TABLE X. PERFORMANCE RESULTS FOR TRUNCATED REAL TIME DATA

S.No	**Classifier**	**Precision**	**Recall**	**FMeasure**
1	CV Parameter Selection	0.071	0.266	0.112
2	ZeroR	0.071	0.266	0.112
3	VFI	0.68	0.509	0.541
4	User Classifier	0.071	0.266	0.112

VII. ISSUES

When tried to simulate, for some algorithms it had taken more time that is not even better for a perfect intrusion detection system in reality. When the number of instances increased step by step, the simulation time to build the model has also seen increasing rapidly. At some stages for algorithms that take less time always too it is taking more than a half an hour. For some algorithms it is reporting that the heap size for the virtual machine memory is not enough for the computation of performance values. Data preprocessing is also difficult for record sizes above 10000. There for those larger amount of time is taken for even the preprocessing step in weka.

VIII. CONCLUSION

In our experiment we have evaluated the performance of Intruder Data Mining algorithms and compared the results using two sets of the KDDCUP'99 dataset. Thereby founded out under each of the 7 classifier categories, the best performing algorithm and also the best out of all categories based on the criteria such as FPR, Classification Rate mainly for individual and also both sets of data. In future by using weka tool we will test the real time data for each of the algorithm and note down the performance values such as Classification rate, False Positive Rate, Time Duration and Cost to build the model. Thereby we will find which algorithm is best performing in real time out of the 46 algorithms for which the KDDCUP'99 dataset is applicable.

References

[1] M. Sabhnani, G. Sarpen, "Application Of Machine Learning Algorithms to Kdd Intrusion Detection Dataset Within Machine Learning Context", In Proc. Int. Conf. Machine Learning Models, Technologies and Applications , vol. 1, pp. 209-215, 2003.

[2] Zhi Song Pan, Song Can Chen, Gen Bau Ho, Dao Qiang Zhang, "Hybrid Neural Network and C4.5 for Misuse Detection", In Proc. Second Int. Conf. Machine Learning And Cybernetics, Xi-an, Nov.2003.

[3] Dae-Ki Kang, Doug Fuller, Vasant Honavar, "Learning Classifiers For Misuse And Anomaly Detection Using A Bag Of System Calls Representation", In Proc. IEEE Workshop on Information Assurance and Security United States Military Academy, West Point, NewYork, 2005.

[4] James Cannady, "Artificial neural Networks for Misuse Detection", In Proc. Conf. National Information Systems Security, pp. 443-456, 1998.

[5] Saqib Ashfaq, M.Umar Farooq, Asim Karim, "Efficient Rule Generation For Cost Sensitive Misuse Detection Using Genetic Algorithms", In Int. Conf. Computational Intelligence and Security, vol.1, pp.282 – 285, Nov. 2006 .

[6] S. Mukkamala, G. Janoski, A H. Sung, (2002), "Intrusion Detection using Neural Networks and Support Vector Machines", Proc. IEEE Int. Joint Conf. on Neural Networks, IEEE Computer Society Press, pp.1702-1707.

[7] Anup K. Ghosh, Aaron Schwartzbard & Michael Schatz, "Learning Program Behavior Profiles for Intrusion Detection", Proc. Workshop on Intrusion Detection and Network Monitoring, 1999, USENIX Technical Program, pp. 51–62.

[8] Srinivas Mukkamala and Andrew H. Sung, 2005 "Feature Selection for Intrusion Detection using Neural Networks and Support Vector Machines", Proc. Second Int. Symposium on Neural Networks, ISNN 2005.

THE AUTHORS

Mr. A.Siles Balasingh M.Tech (I.T) working as Asst. Professor in School of Computer Science and Engineering in St.Joseph University in Tanzania, East Africa. He has published six network research papers in referred International journals; He has guided 30 projects for B.E final year students. His areas of interests are Artificial Intelligence, Computer Architecture, Computer Networks and security, and Ethical hacking.

Mr.N.Surenth M.Tech (I.T) He is currently working as Asst. Professor in School of Computer Science and Engineering in St. Joseph College of Engineering and Technology, Dar Es Salaam, Tanzania, East Africa. He has guided more than 15 projects to final year B.E/B.Tech students and his area of interest in Computer Networks and Artificial Intelligence