Design of two stage filter using Enhanced Adaboost for improving attack detection rates in network intrusion detection

P.Natesan[1], P.Balasubramanie[2]
Department of Computer Science and Engineering,
Kongu Engineering College, Perundurai, Erode 638 052, Tamilnadu, India

Abstract: Based on the analysis and distribution of network attacks in KDDCup99 dataset and real time traffic, this paper proposes a design of two stage filter which is an efficient and effective approach in dealing with frequent attacks and infrequent attacks in networks. The first stage of the filter is designed using Enhanced Adaboost with Decision tree algorithm to detect the frequent attacks happened in the network and the second stage of the filter is designed using Adaboost with Naïve Byes algorithm to detect the infrequent attacks in the network. The performance of this design is tested with the KDDCup99 dataset and is shown to have high detection and low false alarm rates.

Keywords: Enhanced Adaboost, frequent attack filter, infrequent attack filter, decision tree, Naive Bayes classification, detection rate, false alarm rate

I. INTRODUCTION

With the development of the Internet, web applications are becoming increasingly popular and it plays an important role in human life. Consequently, the various Internet resources are becoming the major targets of many attacks. Due to the growing number of users, networking components and the applications in the Internet, it is mandatory that development of new techniques that can secure and protect the Internet resources against the various attacks. These issues have given rise to a research on Network Intrusion Detection systems.

The goals of network intrusion detection are to identify, classify and possibly respond to malicious or suspicious activities [1, 2]. There are basically two types of intrusion detection systems namely anomaly detection and misuse detection. Anomaly detection system first learns normal system activities and then alerts all system events that deviate from the learned model and the misuse detection uses the signature of attacks to detect intrusions by modeling attacks.

A. Related work

The field of network intrusion detection and network security has been around since late 1990s.Since then, a number of frameworks and methodologies have been proposed and many tools have been built to detect network intrusion. Various methodologies such as rule based algorithm, classification, clustering, genetic algorithms, support vector machines, hybrid classification and others have been used to detect network intrusions. In this section, we briefly discuss few of these methodologies and frameworks.

Weiming Hu et. al, [3] have proposed an Adaboost based algorithm for network intrusion detection which used decision stump as a weak learner. The decision rules are provided for both categorical and continuous features and some provision was made for handling the overfitting.

N.B.Amor et. al, [4] discussed the use of Decision tree and Naïve Bayes classifiers for network intrusion detection. The decision trees select the best features for each decision node during the construction of the tree based on some well defined criteria. One such criterion is to use the information gain ratio, which is used in C4.5. Decision trees generally have very high speed of operation and high attack detection accuracy. The Naïve Bayes classifiers make strict independence assumption between the features in an observation resulting in lower attack detection accuracy when the features are correlated, which is often the case for intrusion detection.

Natesan et. al, [5] proposed the use of multiple base learners with Adaboost algorithm. The Decision tree algorithm, Naïve Bayes and Bayes Net are used as base learners and it is also combined in three different ways with Adaboost and its performance is better in terms of Attack detection rate and false alarm rate. Xiang et al [6] proposed multiple-level Hybrid Classifier (MLHC), which involved both the supervised classification stages and unsupervised Bayesian clustering to detect intrusions. There are four classification stages in hybrid classifier which uses Bayesian clustering and decision tree technologies. Gupta et al [7] proposed a Layered approach using Conditional Random Fields (CRFs), where it is considered that the attack categories as layers and different features were selected for each layer.

The most closely related work, to our work is of Kok – Chin khor et al [8]. They divided the training set in to non-rare attack categories and rare attack categories and trained the classifiers using these two training datasets. The methodologies used in the cascaded classifier approach were Bayes net and C4.5 decision tree. In our work, we define two stage filters, first stage is to filter the frequent attack categories and the second stage is to filter infrequent attack categories. The key difference between our work and in [8] is that they used same set of features for both the classifiers, while we use different set of features for frequent attacks detection and the infrequent attacks detection in our filter. The second difference is that the model in [8] fails to improve the detection rate of Probe attack category which can be addressed in our system. Finally we test the performance of our system on the novel attacks.

B. Dataset Analysis

Under the sponsorship of Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory (AFRL), MIT Lincoln laboratory has collected and distributed the datasets for the evaluation of researches in computer network intrusion detection systems. The KDDCup99 dataset is subsets of the DARPA benchmark dataset [9].

KDDCup99 training dataset is about four giga bytes of compressed binary TCP dump data from seven weeks of network traffic, processed into about five million connections record each with about 100 bytes. The two weeks of test data have around two million connection records. Each KDDCup'99 training connection record contains 41 features and is labeled as either normal or an attack, with exactly one specific attack type [9].

There are about 494,020 records in KDDCup'99 training set and 311,029 records in the KDDCup'99 test set. The various attack types in the datasets are grouped into attack categories in order to combine similar attack types into a single category which could improve the detection rate. The training set consists of 24 attack types and the test set contains 38 attack types in which 14 attack types are novel attacks. All the attacks in the dataset fall into four major categories, namely, *Denial of Service (Dos), Probing (Probe), Remote to Local (R2L) and User to Root (U2R).* Table.1 shows the attacks in the KDDCup'99 training set and the additional attacks present in the KDDCup'99 test set. Table.2 and Table.3 shows the number of records for each attack category in the training and testing datasets respectively.

Table 2: Number of samples in the KDDCup'99 Training set and distribution of attacks

Attack Category	Number of samples	Distribution of Attacks in %
Normal	97,277	19.6909
Dos	391,458	79.2393
Probe	4,107	0.8313
R2L	1,126	0.2279
U2R	52	0.0105
Total	494,020	100

Table 3: Number of samples in the KDDCup'99 Test set and distribution of attacks

Attack Category	Number of samples	Distribution of Attacks in %
Normal	60,593	19.4814
Dos	229,853	73.9008
Probe	4,166	1.3394
R2L	16,189	5.2049
U2R	228	0.0733
Total	311,029	100

Table 1: Attacks present in the KDDCup'99 Datasets

Attack Category	Attacks in KDDCup'99 Training set	Additional attacks in KDDCup'99 Test set
Dos	back, neptune, smurf, teardrop, land, pod.	apache2, mailbomb, processtable.
Probe	satan, portsweep, ipsweep, nmap.	mscan, saint.
R2L	warezmaster, warezclient, ftpwrite, guesspassword, imap, multihop, phf, spy	sendmail, named, snmpgetattack, snmpguess, xlock, xsnoop, worm.
U2R	rootkit, bufferoverflow, loadmodule, perl.	httptunnel, ps, sqlattack, xterm

The percentage of distribution of attacks is not uniform in the training set and the test set. Also, the probability of distribution of attacks in the test set is different from training set. For example, there are about only 0.2% of R2L attacks in the training set but it is about 5.2% in the test set. This is one of the challenging tasks in the classification of attacks.

II. METHODOLOGY IN BUILDING IDS

This section will give a detailed description about the Rough set theory which is used to extract the relevant features for detecting the specific category of attacks from the generic 41 features present in the KDDCup99 dataset. With the selected features, the enhanced Adaboost algorithm can achieve the low false alarm rate with high attack detection rate. This section also discuss about the enhanced Adaboost algorithm.

A. Rough set theory

Z.Pawlak introduced the Rough set theory in the early 1980s, is an extension of set theory for study of the intelligent systems characterized by insufficient and incomplete information [10]. Recently, rough set theory has attracted a lot of attention and has been applied in the areas of patterns extraction, text classification, machine learning, information retrieval and etc. [11-13].

We present an overview of rough set theory and the notations used in this paper. In rough set theory, an information system, which is also called a decision table, is defined as S = {U, A, V, f}, where U = {U_1, U_2, … , U_m} is a non-empty, finite set of objects called universe. A = {a_1, a_2, … a_n} is a non-empty, finite set of attributes. Here, m is the number of objects and n is the number of attributions. It includes two non-intersecting subsets: one is condition attributes subset C and another is decision attributes subset D, namely $A = C \cup D$, $C \cap D = \Phi$. $V = \cup V_a$

($a \epsilon A$) is a set of values of attributes in A, V_a is

called the domain of a.

F: U x A -> V_a is an information function, for

any , $a \epsilon A$, $x \epsilon U$, $f(x, a) \epsilon V_a$

The reduction of attributes means to find the minimum condition attribution subset whose classification quality is identical to the original condition attribution set.

Suppose U is the universe R is a group of

equivalence relation, r ⍰⍰ R, if U | IND(R) = U |

IND(R-r), then we say r can be deducted from R. If every attribution in P = R-{r} cannot be deducted, then we say P is a reduction of R.

B. Enhanced Adaboost Algorithm

AdaBoost is a fast machine learning algorithm, can be used in conjunction with many other learning algorithms to improve their performance. It calls a weak classifier repeatedly in a series of rounds. In each round of operation the classification error is calculated for various categories of attacks. The reweight is calculated and assigned to the instances. The Enhanced Adaboost algorithm uses C4.5 Decision tree and Naïve Bayes classification algorithms as its base learner. The pseudo code of our Enhanced Adaboost algorithm is given.

Input: *D*, a set of *d* class-labeled, n training network connections

```
    k, the number of rounds , apply a classification learning scheme.
Output: An IDS model.

Steps:

(1)   Split the training dataset into two datasets $D_1$ and $D_2$
(2)   Extract the relevant features by applying Rough set
(3)   Initialize the weight of each network connection in $D_x$ to $1/d$; x is the number of
      splits
(4)   For x=1 to 2 do
(5)     For $i$ = 1 to $k$ do
(6)           Sample $D_i$ with replacement according to the network connection weights to obtain $D_s$;
(7)           Use training set $D_s$ to derive a model, $M_i$;
(8)           Calculate error ($M_i$), the error rate of the model $M_i$
(9)           if $error(M_i) > 0.5$ then
(10)                Reinitialize the weights to $1/d$
(11)                Go back to step 5 and try again;
(12)          End if
(13)  For each network connection in $D_s$ that was correctly classified
        Do
(14)          Update the weight of the network connection by
              Error ($M_i$) = (1-Error ($M_i$));
(15)          Normalize the weight of each network connection
(16)      End for
(17)  End for
```

III. EXPERIMENTS

The various stages involved in our experiments are: splitting the training dataset, filter design and feature selection, training the filter and testing the model using KDDCup99 test set.

A. Splitting the training dataset

The KDDCup99 training dataset is split in to two training datasets where one consisted of records for Normal, DoS and Probe attack categories and the other consisted of records for Normal, R2L and U2R attack categories

B. Filter design and Feature selection

The large number of features in the KDDCup99 dataset increases the computational and space cost, besides the redundant characteristics of the attributes make the attack detection accuracy dropped. Hence, to reduce the cost involved in computation and storage, rough set based feature selection was applied to select features that were relevant to detect the frequent attacks and infrequent attacks in the KDDCup99 Training set. We have proposed a two stage filter which consists of two stages as shown in Figure.1, the first stage is used to filter the frequent attacks and the second stage is for filtering the infrequent attacks in the network. The proposed method is called a two stage filter with Adaboost.

C. Frequent attacks detection stage

This stage of the filter is aimed to detect Dos and Probe attacks which happened frequently. In Dos attacks the attacker makes some computing or memory resources too busy or too full to handle legitimate requests or denies legitimate users access to a machine. Probe attacks scan a network to gather information or to find known vulnerabilities. An intruder with a map of machines and services that are available on a network can use the information to look for exploits. The selected features to detect the frequent attacks are shown in Table.4.

D. Infrequent attacks detection stage

Feature Number	Feature Name	Description
1	Duration	Length of the connection in seconds
2	protocol_type	Type of the protocol used. For eg. TCP
3	Service	Network service on the destination like HTTP, Telnet.
4	Flag	Normal or error status of the network connection.
5	src_bytes	Number of data bytes from source to destination.
6	dst_bytes	Number of data bytes from destination to source.
7	Land	1 if connection is from/to the same host/port; 0 otherwise
19	num_access_files	Number of operations on access control files
20	number_outbound_cmds	Number of outbound commands in an ftp session
23	Count	Number of connections to the same host as the current connection in the past two seconds.
25	serror_rate	Percentage of connections that have "SYN" errors.
27	rerror_rate	Percentage of connections that have "REJ" errors.
30	diff_srv_rate	Percentage of connections that have same services.
32	dst_host_count	Count for destination host.
33	dst_host_srv_count	srv_count for destination host
34	dst_host_same_srv_rate	same_srv_rate for destination host
35	dst_host_diff_srv_rate	diff_srv_rate for destination host.
39	dst_host_srv_serror_rate	srv_serror_rate for destination host.

The Infrequent attacks detection stage is aimed to detect R2L and U2R attacks. Remote to User (R2L) is an attack that a remote user gains access of a local user/account by sending packets to a machine over a network communication.

User to Root (U2R) is an attack that an intruder begins with the access of a normal user account and then becomes a root-user by exploiting various vulnerabilities of the system. The selected features to detect the infrequent attacks are shown in Table. 5.

Figure .1: Two stage filter

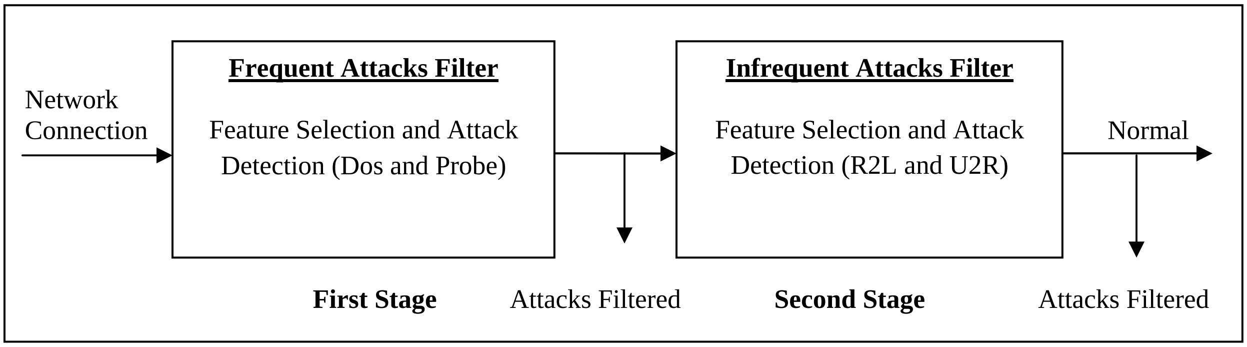
Network Connection
Frequent Attacks Filter
Feature Selection and Attack Detection (Dos and Probe)
Infrequent Attacks Filter
Feature Selection and Attack Detection (R2L and U2R)
Normal
First Stage
Attacks Filtered
Second Stage
Attacks Filtered

Table. 4: Features selected for Frequent attacks detection stage

Table. 4 : Features selected for frequent attacks detection stage

Table. 5: Features selected for infrequent attacks detection stage

Feature Number	Feature Name	Description
1	Duration	Length of the connection in seconds
2	protocol_type	Type of the protocol used. For eg. TCP
3	Service	Network service on the destination like HTTP, Telnet.
4	Flag	Normal or error status of the network connection.
5	src_bytes	Number of data bytes from source to destination
6	dst_bytes	Number of data bytes from destination to source.
10	Hot	Number of “hot” indicators.
11	num_failed_logins	Number of failed login attempts.
12	logged-in	1 if successfully logged in; 0 otherwise.
13	num_compromised	Number of compromised conditions.
14	root_shell	1 if root shell is obtained; 0 otherwise.
16	num_root	Number of “root” access.
17	num_file_creations	Number of file creation operations.
18	num_shells	Number of shell prompts.
19	num_access_files	Number of operations on access control files.
21	is_host_login	1 if the login belongs to “hot” list; 0 otherwise.
22	is_guest_login	1 if the login is a guest login; 0 otherwise.
23	Count	Number of connections to the same host as the current connection in the past two seconds
25	serror_rate	Percentage of connections that have “SYN” errors.
27	rerror_rate	Percentage of connections that have “REJ” errors.
30	diff_srv_rate	Percentage of connections that have same services.
32	dst_host_count	Count for destination host.
35	dst_host_diff_srv_rate	diff_srv_rate for destination host.
39	dst_host_srv_serror_rate	srv_serror_rate for destination host.

E. Training and Testing the Model

The two stage filter is constructed using the selected features in Table. 4 & 5. There are 18 features selected for the first stage of the filter which are used for detecting the frequent attacks in the network and 24 features selected for the construction of the second stage of the filter which are used to filter out the infrequent attacks in the network. The learning ability of any classification algorithm is dependent on the characteristics of the connections in the training data set.

The Enhanced Adaboost with Decision tree algorithm is used in constructing the first stage of the filter and it is trained with the training set which consists of Normal, DoS and Probe attack categories. The second stage of the filter is constructed by using the Enhanced Adaboost with Naïve Bayes classification algorithm as its base learner. The training dataset which consists of Normal, R2L and U2R attack categories is used in training the second stage of the filter. The trained model is tested using the KDDCup99 test set.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

For our experiments we use the benchmark KDDCup99 dataset, in which each record represents a separate connection and hence each connection between the two IP addresses is considered to be independent of any other connection. We use the Weka tool to implement the modified Adaboost algorithm and to perform classification with Decision Trees and naïve Bayesian classification.

A. Performance Measures

In machine learning and data mining algorithms, many different metrics are used to evaluate the classification models. We employed two performance measures: Attack detection rate and false alarm rate. These two measures are calculated by using a confusion matrix.

Confusion matrix is a two dimensional matrix representation of the classification results. The upper left cell in the matrix denotes the number of connections classified as Normal while they actually were Normal (i.e. TP), and the lower right cell in the matrix denotes the number of connections classified as Attack while they actually were Attack (i.e. TN).The other two cells (lower left cell and the upper right cell) denote the number of connections misclassified. Specifically, the lower left cell in the matrix denoting the number of connections classified as Normal while they actually were Attack(i.e.FN), and the upper right cell denoting the number of connections classified as Attack while they actually were Normal (i.e. FP).

	Classified as Normal	Classified as Attack
Normal	TP	FP
Attack	FN	TN

Attack Detection Rate (ADR): It is the ratio between total numbers of attacks detected by the system to the total number of attacks present in the dataset.

Attack Detection Rate = $\frac{\text{Total detected attacks}}{\text{Total attacks}}$ * 100

…(1)

False Alarm Rate (FAR): It is the ratio between total numbers of misclassified instances to the total number of normal instances.

False Alarm Rate = $\frac{\text{Total misclassified instances}}{\text{Total normal instances}}$ * 100

….(2)

B. Detecting the frequent attacks

We conducted two sets of experiments. In the first experiment, we considered all the 41 features of the dataset and examined the detection rate of frequent attacks categories such as DoS and Probe attacks. Initially the Decision Tree is constructed and then to improve its classification accuracy the enhanced Adaboost algorithm is used. We perform the same experiment with the 18 features selected as in Table.4 by using Enhanced Adaboost with Decision Tree as its base learner. The experimental results are shown in Table 6.

Table 6: The attack detection rate of first stage of the filter

No. of features	Attacks category	% of detection rate	Training Time(sec)	Test Time(sec)
41	Dos	97.8	12.8	0.59
	Probe	91.7		
18	Dos	98.7	8.7	0.38
	Probe	92.4		

The system takes only 8.7 sec for its training when 18 features considered. Also it takes only 0.38 sec for the testing of the incoming network connection.

C. Detecting the infrequent attacks

To detect the infrequent attacks, initially we conducted an experiment by considering all the 41 features. The Naïve Bayes algorithm is used as a base learner with the enhanced Adaboost algorithm. We perform the same experiment with 24 features which are relevant to detect infrequent attacks and the results are shown in Table 7. The training time and the testing time decreases marginally with the selected 24 features and there is a slight increase in the detection rate of attacks.

D. System performance on Novel attacks

The KDDCup99 test dataset contains some specific type of new attacks that did not present in the KDDCup99 training dataset and this makes the classification task as more challenging. There are about 24 types of attacks in the training set and 14 types of additional novel attacks present in the test dataset as shown in the Table. 1. The network domain experts and intrusion detection system experts suggested that most of these attacks are slight variants of known attacks which are present in training set and the “patterns” of known attacks can be sufficient to detect the novel attacks. We conducted an experiment to test the performance of our system on novel attacks with the selected features and the results are shown in Table.8.

Table 7: The attack detection rate of second stage of the filter

No. of feature	Attacks category	% of detection rate	Training Time(sec)	Test Time(sec)
41	Normal	98.7	14.8	0.83

	R2L	44.5		
	U2R	82.6		
24	Normal	99.3	11.2	0.64

	R2L	49.5		
	U2R	87.5		

Table 8: Detection rate on Novel attacks

Attacks Category	Attacks Name	Number of attack connections in KDDCup99 test set	Number of attacks detected
Dos	apache2	794	567
	mailbomb	5,000	4,567
	process table	759	613
Probe	mscan	1,053	974
	saint	736	728
U2R	httptunnel	158	121
	ps	16	11
	sqlattack	2	1
	xterm	13	10
R2L	sendmail	17	12
	named	17	5
	snmpgetattack	7,741	3,865
	snmpguess	2,406	1,986
	xlock	9	2
	xsnoop	4	1
	worm	2	0
Total		18,729	13,463(71.8%)

The performance of our two stage filter with Adaboost on novel attacks is remarkably high (an increase of 32%) as compared with the work in [18].

E. Comparisons of detection rate with different algorithms

The detection rate of our algorithm is compared with existing work as shown in Table. 9, which is tested on the benchmark KDDCup'99 dataset. The performance of our proposed two stage filter was comparatively better than existing work in detecting the DoS, R2L and U2R attacks (DoS - 98.7%, R2L - 49.5% and U2R - 87.5%). Hence, it should be considered for the building of IDS.

V. CONCLUSION AND FUTURE WORK

We have proposed the Rough set theory for extracting relevant features and the Enhanced Adaboost algorithm for detecting the network intrusion. The experiment is conducted with all 41 features and with the selected features. The attacks detection rate is increased considerably with the selected features and the computational cost falls drastically. The various issues of intrusion detection system such as attack detection rate, false alarm rate and computational time for building robust, scalable and efficient system are addressed. It is important to have very quick attack detection with higher detection rate. The experiment result shows that the Rough set theory with enhanced Adaboost algorithm has quick attack detection with a high detection rate.

The areas for future research include the considering other feature selection techniques and increasing the stages of filter to seek the possibility of improving the Probe attacks detection rate and also the overall detection rate.

Table 9: Comparison with other algorithms

Name of the method	% of Detection rate				
	Normal	DoS	Probe	R2L	U2R
Cascaded classifier using J48-BN [8]	97.4	97.8	73.3	48.2	87.3
Multi layered hybrid classifier [6]	96.8	98.6	93.4	46.9	71.4
KDD'99 Winner[19]	**99.5**	97.1	83.3	8.4	13.2
Layered approach using CRFs[7]	N/A	97.4	**98.62**	29.6	86.3
Proposed two stage filter using Adaboost	99.2	**98.7**	92.4	**49.5**	**87.5**

REFERENCES

1. Xuan Dau Hoang, Jiankun Hu, and Peter Bertok, “A program-based anomaly intrusion detection scheme using multiple detection engines and fuzzy inference,” Journal of Network and Computer Applications, Vol. 32, pp. 1219-1228, 2009.
2. P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fernandez, and E.Vazquez, “Anomaly-based network intrusion detection: Techniques,systems and challenges,” Computers & Security, Vol. 28, pp. 18-28, 2009.
3. Weiming Hu, Wei Hu and Steve Maybank, “AdaBoost-Based Algorithm for Network Intrusion Detection,” IEEE Transactions on Systems, Man and Cybernetics, Vol. 38, pp. 577-583, April-2008.
4. N. B. Amor, S. Benferhat, and Z. Elouedi, “Naïve Bayes vs. decision trees in intruison detection systems,” In Proc. of the 2004 ACM Symposium on Applied Computing, New York, pp. 420-424, 2004.
5. Natesan P, Balasubramanie P and Gowrison G, "Adaboost with Single and Compound weak classifier in Network Intrusion Detection”, In Proceedings of International conference on Advanced computing, Networking & Security” , Vol. 1, pp 282-290, Dec-2011.
6. Xiang C, Yong PC, Meng, “Design of multiple-level hybrid classifier for intrusion detection system using Bayesian clustering and decision trees”. Pattern Recognition Letters Vol. 29, No. 7, pp. 918–924, 2008.
7. Gupta KK, Nath B Layered approach using conditional random fields for intrusion detection. IEEE Trans Dependable Secure Computing Vol. 7, No. 1, pp. 35–49, 2010.
8. Kok-Chin Khor, Choo-Yee Ting and Somnuk Phon-AmnuaisukA, "cascaded classifier approach for improving detection rates on rare attack categories in network intrusion detection", Journal of Applied Intelligence, DOI 10.1007/s10489-010-0263-y
9. KDDCup99,Dataset,http://kdd.ics.uci.edu/databases/kddcup99/ kddcup99.html. 1999.
10. Z.Pawlak, “Rough sets”, International Journal of Computer and Information Sciences, Vol.11, No. 5, pp. 341-356, 1982.
11. Lei Shi, Li Zhang, Xinming Ma and Xiaohong Hu, “Rough set Based Personalized Recommendation in Mobile Commerce. 2009 International conference on Active Media Technology”, Lecture Notes in Computer Science, pp. 370-375, 2009.
12. Sabhnani, M. R., & Serpen, G. Application of machine learning algorithms to KDD intrusion detection dataset with in misuse detection context. In Proceedings of the international conference on machine learning: Models, technologies, and applications. pp. 209–215, 2003..
13. Xuren, W., Famei, H., & Rongsheng, X. Modeling intrusion detection system by discovering association rule in rough set theory framework. In Proceedings of the international conference on computational intelligence for modeling control and automation, and international conference on intelligent agents. Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), 2006.
14. J.W. Han and M.Kamber, Data Mining: Concepts and Techniques, 2^{nd} edition. Morgan Kaufmann, pp. 310-318, 2006.
15. N.Friedman, D.Geiger and M.Goldsmidt, “Bayesian Network Classifiers,” Machine Learning, Vol.29, pp 131-163, Nov 1997.

16. Yoav Freund, Robert E.Schapire. "A Decision theoretic generalization of on-line learning and an application to boosting," Journal of Computer and System Sciences. Vol. 55, No.2, pp, 119-139, 1997.
17. P.N. Tan, Introduction to Data Mining. Reading, MA:Addison-Wesley, 2006.
18. Shi-Jinn Horng, Ming-Yang Su, Yuan-Hsin Chen, Tzong-Wann kao, Rong-Jian Chen, Jui-Lin Lai, Citra Dwi Perkasa, "A novel intrusion detection system based on hierarchical clustering and support vector machines", Expert Systems with Applications, Vol.38, pp.306-313, 2010, DOI:10.1016/j.eswa.2010.06.066
19. Pfahringer, B. Winning the KDD99 classification cup: Bagged boosting.SIGKDD Explorations, 1(2), 65–66, 2000.

AUTHORS PROFILE

P.Natesan has completed his M.E., (Computer Science &Engineering) in Anna University, Tamilnadu , India in 2005 . Now he is doing research in the field of Network Security. Currently, he is working as Associate Professor in the Department of Computer Science & Engineering, Kongu Engineering College, Tamil Nadu, India. He has completed 15 years of teaching service. He has published 5 articles in International / National journals.

Dr.P.Balasubramanie has completed his Ph.D., degree in Theoretical Computer Science in 1996 in Anna University. He was awarded Junior Research Fellow in the year 1990 by CSIR. Currently he is a Professor in the department of Computer Science & Engineering in Kongu Engineering College, Tamilnadu, India. He has completed 17 years of teaching service. He has published more than 100 articles in International /National Journals. He has authored 8 books with reputed publishers. He has guided 14 Ph.D., scholars and guiding 10 research scholars. He is serving as a Editorial/Advisory board member for many Journals.