A LAYERED APPROACH TO ENHANCE DETECTION OF NOVEL ATTACKS IN IDS

Neelam Sharma[1], Saurabh Mukherjee[2]

[1]Research Scholar & [2]Associate Professor,
Department of Computer Science, Banasthali University, Jaipur (Rajasthan), India

ABSTRACT

As the network attacks have increased in huge numbers over the past few years, intrusion detection system (IDS) is increasingly becoming a critical component to secure the network. Current IDSsare unable to detect all kind of novel attacks because they are designed on limited environment to restricted applications. Many researchers proposed approaches which are able to achieve good detection accuracy for old attacks but poor detection performance for new attacks In this paper we introduce a three-layer approach to enhance the perception of intrusion detection on reduced feature set to detect both known and novel attacks.

KEYWORDS: *Anomaly detection, Novel attacks, Layered approach, Sensitivity, Data mining, Naïve Bayes Classifier, Feature selection*

I. INTRODUCTION

An intrusion detection system (IDS) is a system for detecting intrusions that attempting to misuse the data or computing resources of a computer system. In the last decade lots of computer security techniques have been intensively studied to defend against various cyber-attacks and computer viruses. Among them, network intrusion detection system has been considered to be one of the most promising methods for defending vibrant intrusion behaviors. Different detection techniques can be employed to search for attack patterns in the data monitored. The two major approaches are misuse detection and anomaly detection. Misuse detection consists of using patterns of well-known intrusions to match and identify known labels for unlabeled datasets. It is unable to detect new attacks whose patterns are not presented. Anomaly detection approach is important in order to detect novel attacks. It typically relies on knowledge of normal behavior and detects any deviation of a new behavior from the learned normal profiles. However, false positives are also weaknesses of anomaly detection but the low false negatives are its strength. Current anomaly IDS is unable to detect all kind of novel attacks because they are designed to restricted applications on limited environment. The essential step in successfully detecting intrusions is to develop a model that describes most known as well as novel unseen attacks.

In this paper we use original NSL-KDD [1] training and the test dataset, which have totally different distributions due to novel intrusions, introduces in the test data. The training dataset is made up of 22 different attacks out of 39 present in the test data. The attacks that have any occurrences in the training sets should be considered as known attacks and others those are absent in the training set and present in the test set, considered as novel attacks. However, many researchers achieved good detection accuracy for old attacks that is included in the training data but poor detection accuracy for new attacks that are only in the test data. In our study we have introduce new merged dataset of training and test set to improve the precision and accuracy for all types of attacks.

Table 1 shows the different attack types for both training (known) and additional attack types included in testing (novel) for the following four categories:

Table 1: Depicts Known and Novel attacks

Attacks Category	Known Attacks	Novel Attacks
DoS	back, land, neptune, pod, smurf,teardrop	apache2,udpstorm, processtable, mailbomb
Probe	Ipsweep, satan, nmap, portsweep	saint, mscan
R2L	ftp_write, guess_passwd, warenzmaster, warezclient, imap, phf, spy, multihop	named, xclock, sendmail, xsnoop, worm,snmpgetattack, snmpguess
U2R	rootkit, loadmodule, buffer_overflow, perl	xterm, ps, sqlattack, httptunnel

- Denial-of-Service (DoS): Attackers tries to prevent legitimate users from using a service, these are smurf, neptune, back, teardrop, pod and land.
- Probe: Attackers tries to gain information about the target host. Port Scans or sweeping of a given IP-address range typically fall in this category (e.g. saint, ipsweep, portsweep and nmap).
- User-to-Root(U2R):Attackers has local access to the victim machine and tries to gain super user privileges, these are buffer_overflow, rootkit, landmodule and perl.
- Remote-to-Local(R2L): Attackers does not have an account on the victim machine, hence tries to gain access, these are guess_passwd, ftp_write, multihop, phf, spy, imap, warezclient and warezmaster.

The rest of this paper is organized as follows. In section 2, we discuss the related work. In section 3, we describe NBC with its complexity and discretization technique. Section 4 presents the Feature Selection. The proposed method described in section 5. The experiments and results are presented in section 6. Finally, we summarize the paper and outline future research in section 7.

II. RELATED WORK

IDSs are still experiencing difficulties in detecting intrusive activity on their networks since novel attacks are consistently being encountered. Many IDSs are rule based systems which have limited extensibility for novel attacks. The process of encoding rules is expensive and slow. To overcome these limitations, a number of IDSs employ data mining techniques which are more flexible and deployable.

Over the past several years, a growing number of research projects have applied data mining to intrusion detection with different algorithms [2,3,4].On the basis of type of processing related to the behavioral model of the target system, authors in [5] classified anomaly detection techniques into three main categories: statistical-based, knowledge-based and machine learning based. In [6] authors investigated the applicability of the Junction Tree Algorithm (JTA) in anomaly based intrusion detection. The approach aims at building privileged process profiles and to detect anomalies by measuring deviation from the created profile. The disadvantage of this proposed method is its considerable computational price i.e. $O(TM^2)$, where M^2 is the cardinality of the clique state space. Anomaly network intrusion detection based on data mining techniques such as decision tree (DT), naïve Bayesian classifier (NB), neural network (NN), support vector machine (SVM), k-nearest neighbors (KNN), fuzzy logic model, and genetic algorithm have been widely used by researchers to improve the process of intrusion detection [7]-[13]. However, there exist various problems that induce the complexity of detection systems such as low detection accuracy, unbalanced detection rates for different attack types, and high false positives.

There have been many techniques used for machine learning applications to tackle the problem of feature selection for intrusion detection. In [14], author used PCA to project features space to principal feature space and select features corresponding to the highest eigen values using Genetic Algorithm. In [15] author used feature ranking algorithm to reduce the feature space by using 3 ranking algorithm based on Support Vector Machine (SVM) , Multivariate Adaptive Regression Splines (MARS) and linear Genetic programs (LPGs).In [16] author proposes “Enhanced Support

Vector Decision Function "for feature selection, which is based on two important factors. First, the feature's rank, and second the correlation between the features. In [17], author proposes an automatic feature selection procedure based on Correlation –based Feature Selection (CFS).
In [18] author investigate the performance of two feature selection algorithm involving Bayesian network(BN) and Classification & Regression Tee (CART) and ensemble of BN and CART and finally propose an hybrid architecture for combining different feature selection algorithms for intrusion detection. In [19], author proposes two phases approach in intrusion detection design. In the first phase, develop a correlation-based feature selection algorithm to remove the worthless information from the original high dimensional database. Next phase designs an intrusion detection method to solve the problems of uncertainty caused by limited and ambiguous information. In [20] author presents an Intelligent Intrusion Detection and Prevention System (IIDPS), which monitors a single host system from three different layers; files analyzer, system resource and connection layers. The approach introduced, a multi – layered approach, in which each layer harnesses both aspects of existing approach, signature and anomaly approaches, to achieve a better detection and prevention capabilities. In [21] authors presented "A frame work using a layered approach for intrusion detection". They have addressed two main issues of ID i.e. accuracy and efficiency by using conditional random fields and layered approach. In [22], Author presents a novel approach for learning from imbalanced data sets, based on a combination of the SMOTE algorithm and the boosting procedure. Unlike standard boosting where all misclassified examples are given equal weights, SMOTE Boost creates synthetic examples from the rare or minority class, thus indirectly changing the updating weights and compensating for skewed distributions. To address such issues, recently many IDSs are designed but only some of them are handling novel attacks

III. NAÏVE BAYES CLASSIFIER

The Naïve Bayes classifier technique is based on the Bayesian theorem and is particular suited when the dimensionality of the input is high. Despite its simplicity Naïve Bayes can often outperforms more sophisticated classification method. It works on strong independence relation assumption [9], that is, features are independent in the context of a session class and the probability of one attribute does not affect the probability of the other. It is defined as follows:

$$P(c|X) = \frac{P(X|c)P(c)}{P(X)} \qquad \text{i.e. } (c|X) = P(x1|c) \times P(x2|c) \times \ldots \times P(xn|c) \times P(c) \tag{1}$$

Where,

- P(c|x) is the posterior probability of class (target) given predictor (attribute).
- PI is the prior probability of class.
- P(x|c) is the likelihood which is the probability of predictor given class.
- P(x) is the prior probability of predictor.

Assume that, the effect of the value of a predictor(X) on a given class Iis independent of the values of other predictor.

Time & Space Complexity

The theoretical time complexity for learning a Naïve bayes classifier is O(Np), where N is the number of training examples and p is the number of features. The theoretical space complexity for Naive Bayes algorithm is O(pqr), where p is the number of features, q is values for each feature, and r is alternative values for the class [23].

Discretization for Naïve Bayes Classifier

Research study shows that Naïve Bayes classification works best for discretized attributes and discretization effectively approximates a continuous variable [24]. We used the entropy-based supervised discretization (EBD) method proposed by Fayyad and Irani [25]. It discretizes numeric attributes first using Minimum Description Length (MDL) method.

Given a set of samples *I,* the basic method for EBD of an attribute A is as follows:

1. Each value *v* of *A* can be considered as a potential interval boundary Band thereby can create a binary discretization (e.g. $A < v$ and $A \geq v$).
2. Given I, the boundary value selected is the one that maximizes the information gain resulting from subsequent partitioning. The information gain is:

$$\text{InfoGain (I,B)} = \text{E (I)} - \text{CIE(I,B)} \quad (2)$$

Where CIE (I, *B*) is the *class information entropy* determined by the formula:

$$CIE(I,B) = \frac{|I_1|}{|I|}\text{E}(I_1) + \frac{|I_2|}{|I|}\text{E}(I_2) \quad (3)$$

Where$|I_1|$ and $|I_2|$ correspond to the examples of *I* satisfying the conditions $A < B$ and $A \geq B$ respectively. The entropy function E for a given set I_i is calculated based on the class distribution of the samples in the set, i.e.:

$$E(I_i) = -\sum_{j=1}^{m} \frac{c_j}{c}\log_2\left(\frac{c_j}{c}\right) \quad (4)$$

Where $\frac{c_j}{c}$ is the probability of class c_j in I_i, determined by the proportion of samples of class c_j in the set I_i and m is the number of classes in I_i.

3. The process of determining a new interval boundary is recursively applied to each interval produced in previous steps, until the following stopping criterion Δ based on MDL principle is satisfied:

$$InfoGain(I,B) < \Delta$$

$$\Delta = \frac{log_2(n-1) + log_2(3^m - 2) - [mE(I) - m_1E(I_1) - m_2E(I_I)]}{n} \quad (5)$$

Where m_i is the number of classes represented in the set I_i and n is the number of samples in I.

Since the described above procedure is applied independently for each interval, it is possible to achieve the final set of discretization intervals with different size that is, some areas in the continuous spaces will be partitioned very finely whereas others (with relatively low entropy) will be partitioned roughly.

IV. FEATURE SELECTION

The 41 features for network connection records fall into three categories [26].

- **Intrinsic features**. Intrinsic features describe the basic information of connections, such as the duration, service, source and destination host, port, and flag.
- **Traffic features**. These features are based on statistics, such as number of connections to the same host as the current connection within a time window.
- **Content features**. These features are constructed from the payload of traffic packets instead of packet headers, such as number of failed logins, whether logged in as root, and number of accesses to control files.

Feature selection is an effective and an essential step in successful high dimensionality data mining applications [27]. It is often an essential data processing step prior to applying a learning algorithm. Reduction of the irrelevant features leads to a better understandable model, enhances the accuracy of detection while speeding up the computation and simplifies the usage of different visualization technique. Thus reducing attribute space improves the overall performance of IDS.

Currently features are designed by domain knowledge experts. We also employ domain knowledge and the Backward Sequential Elimination (BSE) [28] to identify the important set of features: starting from the full set of features, BSE successively eliminates the attributes whose elimination most improves accuracy, until there is no further accuracy improvement.

V. PROPOSED METHOD

During the analysis of intrusion detection we observe two main challenging issues in this system. First, the number of intrusions on the network is typically a very small fraction of the total traffic. Therefore the essential step in successfully detecting intrusions is to develop a model that describes most known as well as novel unseen attacks. Second, the attack groups are different in their impact and hence, it becomes necessary to treat them differently.

To improve the novel attack detection rate, while maintaining a reasonable overall detection rate. We proposed a layered model with naïve bayes classifier on discretized values. Since results using discretized features are usually more compact, shorter and accurate using continuous values. In layered model we define three layers for detecting DoS, probe, R2L and U2R attacks. The first layer is to detect major attacks like DoS and Probe. The second layer is for R2L and the third layer for U2R attack detection. Each layer is separately trained with a small set of relevant features and then deployed sequentially. However some researchers have proposed four layer models to detect each type of attack separately. But during experiments we have found that, the feature set for DoS and Probe is almost nearer to the similarity. It motivates us to define a single superset for the investigation of these two major attacks which reduces time and complexity of the model at the major extent.

We select features for each layer, based upon the type of attacks that the layer is trained to detect and feasibility of each feature before selecting it for a particular layer. The selected feature set of proposed model for all the three layers are:

Table 2: Depicts Proposed Model Feature Set for Three Layers

Layer Number	Attack Type	Feature Number	Feature Set for Layer
Layer 1	DoS& PROBE Attack	1, 3, 5, 6, 30, 33, 36	duration, service, src_bytes, dst_bytes, diff_srv_rate, dst_host_srv_count, dst_host_same_src_port_rate
Layer 2	R2L attacks	3, 5, 6, 10, 32	Service, src_bytes, dst_bytes, hot, dst_host_count
Layer 2	U2R attacks	2, 3, 5, 6, 10, 14, 24, 32, 33, 34, 35, 36, 37, 40	protocol_type, service, src_bytes, dst_bytes, hot, root_shell, srv_count, dst_host_count,dst_host_srv_count, dst_host_same_srv_rate, dst_host_diff_srv_rate,dst_host_same_src_port_rate, dst_host_srv_diff_host_rate, dst_host_reeror_rate

*Appendix A shows all Feature set of NSL KDD Dataset

To make the layers independent, some features may be present in more than one layer i.e. the feature set for the layer is not disjoint.

Working of the Layered Model

We implement the layered approach by selecting small set of features for every layer rather than using all the 41 features. This is because all the 41 features are not required for detecting attacks belonging to a particular attack group. We integrate layered approach and the naïve bayes classifier to build a single system. For our experiment we use original NSL-KDD [1] training and the test dataset, which have totally different distributions due to novel intrusions, introduces in the test data. The training dataset is made up of 22 different attacks out of 39 present in the test data. These 39 attacks can be grouped into four classes; Probe, DoS, R2L and U2R.

We train and test each layer to detect only a particular type of attacks. For example, first layer of our proposed model is trained to detect major attacks DoS and Probe only. When such a system is deployed online, other attacks such as R2L can either be seen as normal or major attack. If R2L attacks are detected as normal, we expect them to be detected as attack at other layers in the system. However, if the R2L attacks are detected as major attacks, it must be considered as an advantage since the attack is detected at an early stage. Similarly, if some major attacks are not detected at the major attack layer, they may be detected at subsequent layer. Hence, for four attack classes, we have three

independent layer models, which are trained separately with specific features to detect attacks belonging to that particular group. We represent the layered model in Fig. 1

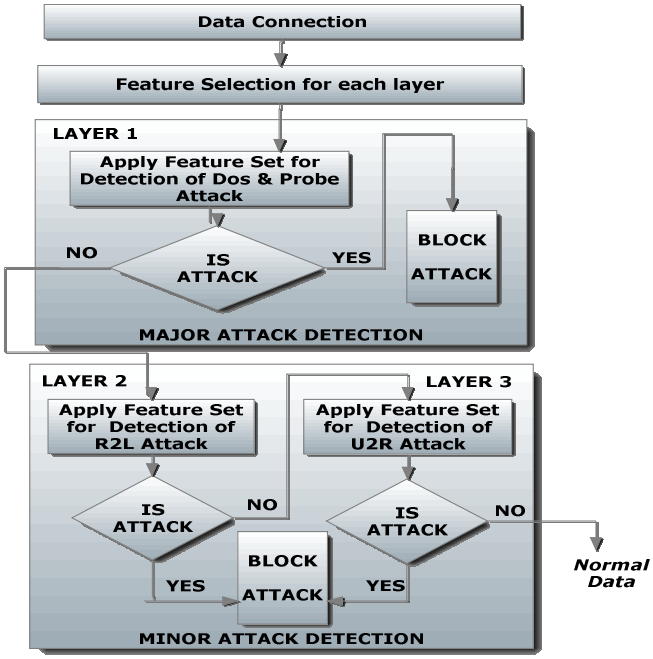

Fig. 1 Shows the working of proposed layered model

VI. EXPERIMENTS AND RESULTS

Dataset Description

For our experiments, we use the NSL-KDD intrusion training and test data set.. It is important to note that the test data is not from the same probability distribution as the training data and it includes specific attack types not present in the training data. Thus, for the purpose of this paper, we modified the datasets. We merged original training and test datasets (i.e. train + test set). As we described in section 1, that the test set contains some novel attacks which are not present in the training set. If we use training-test model i.e. build the learning model using training set and validate it using test set, will never be succeeded because it will fail to recognize/learn those attacks which are not present in the learned model. Finally, we used the new merged dataset with 1, 48, 517 instances in our experiment. Table 2 gives the number of instances for each group of attack in the merged data set.

Table 3. Exemplify distribution of classes and the percentage of attacks occurred in the merged dataset

Category of Class (class)	Number of instances	Percentage of Class Occurrences (Approximate)
Normal	77,054	51.88
DoS	53,385	35.94
Probe	14,077	9.47
R2L	3,749	2.52
U2R	252	0.1
Total	1,48,517	100%

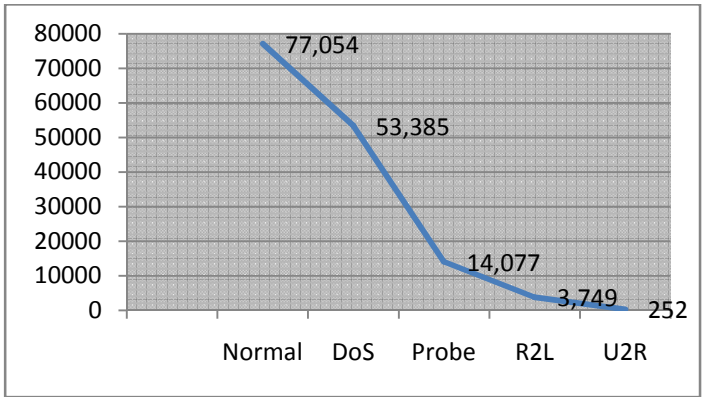

Fig 2 The distribution of network connections in the new dataset

We used WEKA 3.6 a machine learning tool [29], to compute the feature selection subsets for different layers and to measure the classification performance for each of these feature sets. We choose the NBC with full training set and 10-fold cross validation for the testing purposes. In 10-fold cross-validation, the available data is randomly divided into 10 disjoint subsets of approximately equal size. One of the subsets is then used as the test set and the remaining 9 sets are used for building the classifier. The test set is then used to estimate the accuracy. This is done repeatedly 10 times so that each subset is used as a test subset once. The accuracy estimates is then the mean of the estimates for each of the classifiers. Cross-validation has been tested extensively and has been found to generally work well when sufficient data is avai

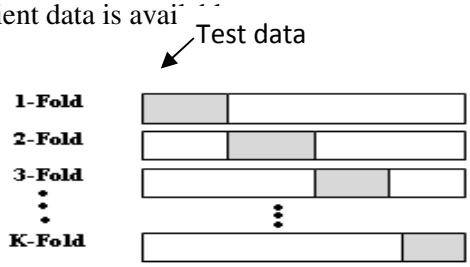

Fig 3 K-Fold Cross Validation

The advantage of K-Fold Cross validation is that all the examples in the dataset are eventually used for both training and testing. The true error estimate is obtained as the average of the separate estimates E_i of test examples/instances.

$$E = \frac{1}{K}\sum_{i=1}^{K} E_i \tag{6}$$

Results

To evaluate the results of our experiments, we have used standard metrics such as confusion & cost matrix, true positive rate, false positive rate and classifier's accuracy.

Confusion Matrix- This may be used to summarize the predictive performance of a classifier on test data. It is commonly encountered in a two-class format, but can be generated for any number of classes. A single prediction by a classifier can have four outcomes which are displayed in the following confusion matrix.

Confusion Matrix		Predicted Class	
		Class=Yes	Class=No
Actual Class	Class=Yes	TP	FN
	Class=No	FP	TN

True Positive (TP), the actual class of the test instance is positive and the classifier correctly predicts the class as positive. False Negative (FN), the actual class of the test instance is positive but the

classifier incorrectly predicts the class as negative. False Positive (FP), the actual class of the test instance is negative but the classifier incorrectly predicts the class as positive. True Negative (TN), the actual class of the test instance is negative and the classifier correctly predicts the class as negative.

True Positive Rate (TPR) or Sensitivity or Recall (R) is defined as:

$$TPR=TP/(TP+FN) \quad (7)$$

False Positive Rate (FPR) is:

$$FPR=FP/(TN+FP) \quad (8)$$

We can obtain the accuracy of a classifier by

$$Accuracy =(TP+TN)/(TP+FN+FP+TN) * 100\ \% \quad (9)$$

Cost Matrix-Different misclassifications have different levels of consequences. For example, misclassifying R2L as Normal is more dangerous than misclassifying DoS as Normal. We use the cost matrix [30] to measure the damage of misclassification.

Let M_{ij} denotes the number of samples in Class I misclassified as Class *j*. C_{ij} indicates the corresponding cost in the cost matrix. *N* be the total number of the samples.

The cost that indicates the average damage of misclassification for each connection is computed as:

$$Cost = \sum \frac{M_{ij} \times C_{ij}}{N} \quad (10)$$

Class Types	Normal	DoS	Probe	R2L	U2R
Normal	0	1	2	2	2
DoS	1	0	2	2	2
Probe	2	1	0	2	2
R2L	3	2	2	0	2
U2R	4	2	2	2	0

We perform two sets of experiments. From the first experiment, we wish to examine the accuracy of NBC and DT for intrusion detection. For this experiment we do not consider feature selection, and the systems are trained using all the 41 features. From the results of this experiment we observe that decision tree achieves higher attack recall rate for DoS, Probe and R2L, with low FPR for all the attack types, while NBC performs much better for U2R attacks. Table 4shows the results achieved from the first experiment.

Table 4 Performance of Non Layered Naïve Bayes Classifier (NBC) & Decision Tree (DT) with 41 features set

Attack Classes	**Non-layered NBC**		**Non-layered DT**	
	Recall (%)	FPR(%)	Recall(%)	FPR(%)
DoS	91.0	0.2	99.9	0.1
Probe	96.5	2.1	99.3	0.4
R2L	91.2	1.3	95.7	0.1
U2R	83.7	1.2	69.8	0

The U2R attacks are very difficult to detect since they involve the semantic details that are very difficult to capture at an early stage. In fact, within decision trees, when a class is represented by a low number of training instances, it leads to a weak learning regarding this class and consequently to a misclassification of testing connections really belonging to it. However, from computational point of view, the construction of naïve bayes is largely faster than decision tree, table 5 shows time consumed to build model decision tree and naïve bayes classifier.

Table 5: Depict time consumed to build model by Naïve Bayes and Decision Tree (with all 41 features)

Classifiers	Time taken to build model
Non-Layered NBC	41.95 Sec.
Non-layered DT	398.64 Sec

Result obtained from first experiment motivate us to move forward for our second experiment using naïve bayes classifier with feature selection for every attack group separately, instead of using all 41 features. This integrated model call by us as layered model with naïve bayes. Investigate on the performance of proposed model theTable6 clearly indicates empirical result of high detection rate for R2L and U2R attack as well as majority attacks.

Table 6: Attack Detection Performance of proposed approach

Attack Types	Recall (%)	FPR(%)
DoS	99.05	0.6
Probe	98.68	0.1
R2L	95.4	0.7
U2R	90.8	0.1

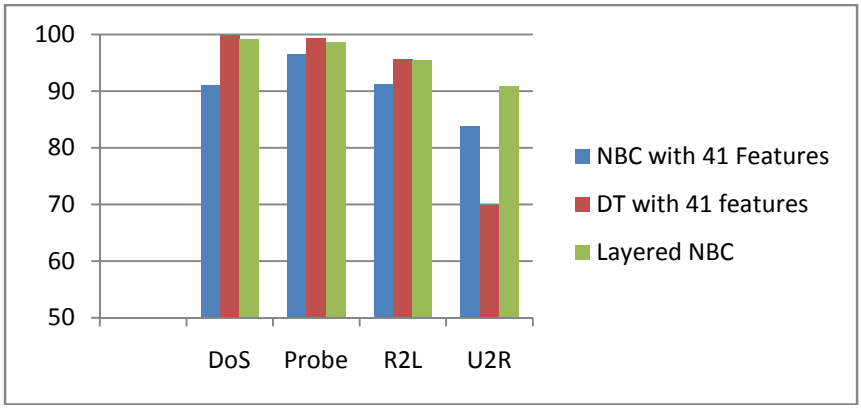

Fig 4 shows comparative results on Recall rate

Fig 4 shows the result of proposed model in comparison of non-layered approach using NBC and DT.

Table 7 Layer-wise time taken to build proposed model

Classifier	Layer 1	Layer 2	Layer 3	Total
NBC	9.16Sec.	3.64 sec.	6.49 Sec	19.29 Sec.

Table 7 shows the total time consumed to build proposed model which is very less in comparison of non-layered NBC and DT as Fig 5 shows.

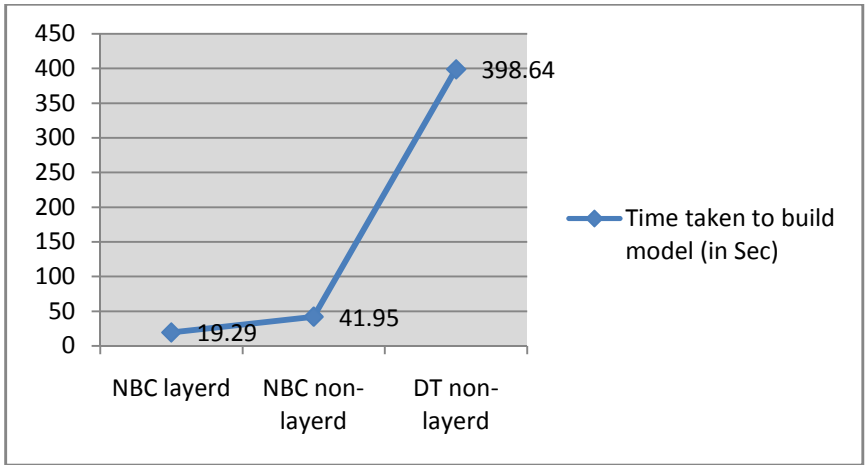

Fig 5 Shows comparative results on time taken to build models

VII. CONCLUSION AND FUTURE WORK

In this paper we have addressed two challenging issues of IDS. First, the number of intrusions on the network is typically a very small fraction of the total traffic and second the attack groups are different in their impact. For these two issues it becomes necessary to develop a model which describes most known as well as novel attacks accurately and efficiently. In our work we performed two experiments. The first experiment is Non *layered approach* using NBC and DT. The result indicates that the performance of DT is much better than NBC, but consumes very large computational time. Hence, we preferred NBC for the second experiment which comprises of three layers for attack detection. The first layer for the major attack detection, second for the R2L and the last one for the U2R attack detection. Since the attack groups are different in their impact, hence we treat them separately by selecting different feature subsets for each layer. The experimental results indicate that the performance of non-layered approach using DT with 41 features is nearly comparable to proposed model but computational point of view, the construction of proposed model is largely faster than decision tree and also improve the detection of minor attacks.

We have not make emphasis on precision vales of attacks. Since the recall and precision goals are often conflicting and attacking them simultaneously may not work well, especially when one class is rare. The area of future research includes improving the recall rate of attacks, without sacrificing the precision value.

REFERENCES

[1] NSL-KDD dataset for network –based intrusion detection systems" available on http://iscx.info/NSL-KDD/

[2] Wenke Lee, Sal Stolfo, and KuiMok, "Adaptive Intrusion Detection: A Data Mining Approach", Artificial Intelligence Review, Kluwer Academic Publishers, 14(6):533-567, December 2000.

[3] Daniel Barbarra, Julia Couto, SushilJajodia, Leonard Popyack, and Ningning Wu, "ADAM: Detecting Intrusions by Data Mining", Proceedings of the 2001 IEEE, Workshop on Information Assurance and Security T1A3 1100 United States Military Academy, West Point,NY, June 2001.

[4] Tamas Abraham, "IDDM: Intrusion Detection Using Data Mining Techniques", DSTO Electronics and Surveillance Research Laboratory, Salisbury, Australia,May 2001.

[5] Lazarevic A, Kumar V, Srivastava J. Intrusion detection: a survey, Managing cyber threats: issues, approaches, and challenges.SpringerVerlag; 2005. p. 330.

[6] E. Nikolova, V Jecheva" Some Evaluations of the Effectiveness of Anomaly Based Intrusion Detection Systems Based on the Junction Tree",2008

[7] Barbara, Daniel, Couto, Julia, Jajodia, Sushil, Popyack, Leonard, Wu,andNingning, "ADAM: Detecting intrusion by data mining," IEEEWorkshop on Information Assurance and Security, West Point, NewYork, June 5-6, 2001.

[8] Lee W., Stolfo S., and Mok K., "Adaptive Intrusion Detection: A datamining approach," Artificial Intelligence Review, 14(6), December2000, pp. 533-567.

[9] N.B. Amor, S. Benferhat, and Z. Elouedi, "Naïve Bayes vs. decisiontrees in intrusion detection systems," In Proc. of 2004 ACM Symposiumon Applied Computing, 2004, pp. 420-424.

[10] Mukkamala S., Janoski G., and Sung A.H., "Intrusion detection usingneural networks and support vector machines," In Proc. of the IEEEInternational Joint Conference on Neural Networks, 2002, pp.1702-1707.
[11] J. Luo, and S.M. Bridges, "Mining fuzzy association rules and fuzzyfrequency episodes for intrusion detection," International Journal ofIntelligent Systems, John Wiley & Sons, vol. 15, no. 8, 2000, pp. 687-703.
[12] YU Yan, and Huang Hao, "An ensemble approach to intrusion detectionbased on improved multi-objective genetic algorithm," Journal ofSoftware, vol. 18, no. 6, June 2007, pp. 1369-1378.
[13] Shon T., Seo J., and Moon J., "SVM approach with a genetic algorithmfor network intrusion detection," In Proc. of 20th InternationalSymposium on Computer and Information Sciences (ISCIS 2005),Berlin: Springer-Verlag, 2005, pp. 224-233.
[14] I Ahmad, A B Abdulah, A S Alghamdi, K Alnfajan,MHussain, Feature Subset Selection for Network Intrusion Detection Mechanism Using Genetic Eigen Vectors, Proc .of CSIT vol.5 (2011)
[15] A. H. Sung, S. Mukkamala. (2004) The Feature Selection and Intrusion Detection Problems. In Proceedings of the 9th Asian Computing Science Conference, Lecture Notes in Computer Science 3029 Springer 2004, pp.
[16] S Zaman, F KarrayFeatures selection for intrusion detection systems based onsupport vector machinesCCNC'09 Proceedings of the 6th IEEE Conference on Consumer Communications and Networking Conference 2009.
[17] H Nguyen, K Franke, S PetrovicImproving Effectiveness of Intrusion Detection by Correlation Feature Selection, 2010 International Conference on Availability, Reliability and Security,IEEE Pages-17-24
[18] S Chebrolu, A Abraham, J P. ThomasFeature deduction and ensemble design of intrusion detection systems, Computers &Security,Volume 24, Issue 4, June 2005, Pages 295-307
[19] T. S. Chou, K. K. Yen, and J. Luo "Network Intrusion Detection Design Using Feature Selection of Soft Computing Paradigms. International Journal of Computational Intelligence 4;3 2008
[20] OludeleAwodele, Sunday Idowu, OmotolaAnjorin, and Vincent J. Joshua "A Multi-Layered Approach to the Design of Intelligent Intrusion Detection and Prevention System (IIDPS)" Issues in Informing Science and Information Technology Volume 6, 2009.
[21] Kapil Kumar Gupta, BaikunthNath and Ramamohanarookotagiri, "A layered approach using conditional random fields for intrusion detection", IEEE Tranc.on Dependence and secure computing, Vol.7, 2010
[22] Nitesh V. Chawla1, Aleksandar Lazarevic2, Lawrence O. Hall3, Kevin Bowyer4, "SMOTEBoost: Improving Prediction of the Minority Class in Boosting" , 7th European conference on principles and practice of knowledge discovery in databases (PKDD) pp. 107 to 109, dubrovnik, Croatia, 2003.
[23] Chris Fleizach, Satoru Fukushima, A naive Bayes classifier on 1998 KDD Cup
[24] Chun-Nan Hsu, Hung-Ju Huang, Tsu-TsungWong,"Why Discretization works for Naïve Bayesian Classifiers", 17th ICML, pp 309-406, 2000.
[25] U.M. Fayyad, K.B Irani, "Multi-interval discretization of continuous-valued attributes for classification learning", In Proceedings of the 13th International Joint Conference on Artificial Intelligence, pp. 1022–1027, 1993.
[26] Wenke Lee and Salvatore J. Stolfo, "A Framework forConstructing Features and Models for IntrusionDetection Systems", ACM Transactions on Informationand System Security (TISSEC), Volume 3, Issue 4,November 2000.
[27] Liu H ,Setiono R, Motoda H, Zhao Z Feature Selection: An Ever Evolving Frontier in Data Mining, JMLR: Workshop and Conference Proceedings 10: 4-13 The Fourth Workshop on Feature Selection in Data Mining
[28] Kittler, J.: Feature selection and extraction. In Young, T.Y., Fu, K.S., eds.:Handbook of Pattern Recognition and Image Processing. Academic Press, New York (1986)
[29] Weka: http://www.cs.waikato.ac.nz/~ml/weka/
[30] Charles Elkan, "Results of the KDD'99 Classifier Learning", SIGKDD Explorations 1(2): 63-64, 2000.

APPENDIX A

List of 41 Features of NSL-KDD Dataset

S.No.	Feature Set	S.No.	Feature Set	S.No.	Feature Set
1.	duration	15.	su attempted	29.	samesrv rate
2.	protocol type	16.	# root	30.	diffsrv rate
3.	service	17.	# file creations	31.	srv diff host rate
4.	flag	18.	# shells	32.	dst host count
5.	source bytes	19.	# access files	33.	dst host srv count

6.	destination	20.	# outbound cmds	34.	dst host same srv rate
7.	land	21.	is host login	35.	dst host diff srv rate
8.	wrong fragment	22.	is guest login	36.	dst host same src port rate
9.	urgent	23.	Count	37.	dst host srv diff host rate
10.	hot	24.	srv count	38.	dst host serror rate
11.	failed logins.	25.	serror rate	39.	dst host srvserror rate
12.	logged in	26.	srvserror rate	40.	dst host rerror rate
13.	compromised	27.	rerror rate	41.	dst host srvrerror rate
14.	root shell	28.	srvrerror rate		

AUTHORS

Neelam Sharma received the M.Tech degree in computer science from Banasthali University, India in 2008. Presently she is pursuing Ph.D in computer science from the same university, focusing on Intrusion Detection System field.

Saurabh Mukherjee is an Associate Professor in Computer Science in Banasthali University. He is Ph.D research supervisor of 05 research scholar. His research areas are in Medical image processing, Cognitive science and Data mining.