A Layered Approach for Pattern Recognition in Large Dataset Using Meta modeling with Classification Techniques

Ankita Gaur, Prof. Vineet Richariya

Abstract

Security of computers and networks that connect them is increasingly becoming of great significant. Intrusion detection is the act of detecting actions that attempts to compromise the clandestinely, credibility or availability of a network resource. It is an important attribute of defensive measure protecting computer system and network traffic from abuses. Here, we are focusing on two important aspects of intrusion detection; one is accuracy and other is performance. In the paper it is demonstrated that high attack detection accuracy can be achieved by using meta-modeling techniques in combination with classification techniques and high performance is attained by the layered approach. To test the results we have used NSL-KDD datasets; and also applied PCA for feature reduction that results in a significant improvement on learning algorithms. In this paper, we have designed and evaluated the combinational models for intrusion detection mechanism, and later we compared those models with each other and tried to find which is more accurate and appropriate to detect intrusion. We have applied meta-modeling because it gives better classification performance than any individual classifier. Our research has shown that the combination of meta-modeling algorithms with SVM gives better overall accuracy than any other combinational model.

Index Terms—Meta-modeling techniques, Classification techniques, Layered approach, PCA.

1.INTRODUCTION

“Absence makes the heart grow fonder” one of the popular proverb, explain us the importance of absence for innovation. Every invention took the footprints of the same proverb. We have encountered several milestones in this journey of inventions. There is no dilemma in considering computer as the most important invention. However, with the increase in demand of this technology, the need to connect it with every nook n corner was also increased. As need is the mother of all inventions; the need gave birth to networking (or internet). Lately, with the increase in the use of internet the concerns of making the internet more secure were also emerged among the technocrats and users. Because of this concern, many intrusion detection techniques came into existence. Though, IDS is considered to be immature and it does not provide a complete defense, but we believe that it can play a significant role in overall security architecture. As in battle field a warning can play a major role, similarly a warning can provide alert to the user about any skeptical attack on the system, hence, this warning indication that the system is under attack, even if the system is not assailable to specific attack, can help users to revamp their installation’s defensive posture and can increase resistance to attack.

Intrusion detection systems mainly base their decisions either on Signal (signature-based detection) or Noise (anomaly-based detection). IDS can also be classified on the phenomenology that they sense. Network-based system can simultaneously monitor numerous hosts; they can suffer from performance problems, especially with increasing network speeds. Another is host-based system that can monitor specific applications in ways that would be difficult or impossible in a network-based system [13]. While there is an existence of Hybrid System, this system is the combination of both signature-based and the anomaly-based systems. Hybrid Systems system can be very efficient when subjected to classification methods and can also be used to label unseen (new instances) as they assign one of the known classes to every test instance. This is because during training the system learns features from all the classes [11]. The only thing required by hybrid system is labeled data. In this paper, we are trying to make a result oriented comparison among different combinational models, which are created by us with layered approach and results are used to find best combinational method for all types of attacks. Organization of the paper - Section 2 gives an overview of the related work. Section 3 defines IDS systems. Section 4 describes meta-modeling techniques. Section 5 describes the classification techniques. Section 6 gives the detail of layered approach. Section 7 explains the reduction methodology. Section 8 shows the experimental setup and results; and lastly in Section 9 we have drawn the conclusion based on the generated results.

2. RELATED WORK

This detection approach was employed to detect attack categories in the NSL-KDD dataset. The technique has achieved the detection rate of 97.48% for DOS, 95.23% for Probe, 99.49% for U2R and 96.48% of R2L respectively. This statics shows that our approach is very much accurate for every type of attack.

In 1997[5], Richard Maclin and David Optiz, presented “An empirical evaluation of boosting and bagging” in which they have shown that when these meta-algorithms are used they produce a larger gain in accuracy. This encouraged many researchers and then in 2008[8] Weiming Hu and Wei Hu presented “An intrusion detection system using adaboost meta-algorithm” which also shows a significant or competitive performance with IDS systems.

In 2010[2], Kapil Kumar Gupta, Baikunth Nath and Ramamohanaroa Kotagiri 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. They have shown that layered CRFs have very high attack detection rate 98.6% for probe and 97.40% for DOS. However, they were outperformed by a significant percent for the R2L and U2R attacks. Where, our approach performs fantastically.

We are also influenced by the work of [9], [12] [5], [22], [21] and many more authors. Literature survey has shown us that in all particular purposes most of the researchers have applied a single algorithm to address all the four attack categories. This has motivated us and helps us to draw an assumption, that the combination of different algorithms would perform different predictions on different attack categories, and may yield a good performance and high prediction comparatively.

3.INTRUSION DETECTION

Intrusion detection as defined by the System Administrators, Audit, Networking and Security (SANs) Institute is the art of detecting inappropriate, inaccurate or anomalous activity. We use intrusion detection systems to protect our network from attacks and abuses, we also use it to detect the violation in security and attacks on network, to document them and to get detailed information about intrusions that occurred [13]. There are following approaches for IDS:

a) Signature-based approach: Design to detect the known attacks. It is very effective for detecting the attacks without generating an overwhelming number of false alarms; it can quickly and reliably diagnose the use of a specific attack tool. But it has a loophole, that it can only detect the attacks which are described in its database.

b) Classification-based approach: This approach uses normal and abnormal datasets of user behavior and uses data mining techniques to train the IDS system. This creates more accurate classification models for IDS as compared to signature-based approaches and thus they are more powerful in detecting known attacks. But still they are not capable of detecting unknown attacks.

c) Anomaly-based approach: The basic assumption of anomaly detection approach is that, attacks are different from normal activities and thus they can be detected by IDS systems that identify these differences. This detection approach can detect unknown attacks also, but still it has a loophole, this approach generates a large number of false alarms due to unpredictable behaviors of users and networks [7]. Data mining approaches are relatively new technique for intrusion detection. There are a wide variety of data mining algorithms drawn from the fields of statics, pattern recognition, machine learning and database. Previous research of data mining approaches for intrusion detection model identified several types of algorithms as useful techniques [11]. Classification and meta-modeling data mining algorithms are investigated as a useful technique for intrusion detection models. In this paper, we investigate following combinational model and compare them with each other. We have six combinational models; they are Adaboost with SVM, decision tree and Naïve Bayesian. And Bagging with SVM, decision tree and Naïve Bayesian. We find the best combination in terms of accuracy and performance.

4.META-MODELING

Meta-modeling is the analysis, construction and development of the frames, rules, constraints, models and theories applicable and useful for modeling a predefined class of problems. With the same concept meta-modeling is also used in data mining world. Here, these meta-modeling techniques are used to improve the accuracy of classifiers and predictors. In our experiment we are using two meta-algorithms one is adaboost and other is bagging.

a) Adaboost:

Adaboost is an abbreviation for Adaptive Boosting, is a machine learning algorithm. It is a meta-algorithm and can be used in conjunction with many other learning algorithms to improve their performance. In 1997, Freud and Schapiro introduced adaboost [8] which was since then enjoying a remarkable attention. Adaboost is adaptive in the sense that subsequent classifiers built are weaker in favor of those instances misclassified by previous classifiers. Simply we can say adaboost generates a set of hypotheses, and combines them through weighted majority voting of the classes predicted by the individual hypotheses. The hypotheses are generated by training a weak classifier, using instances drawn from an iteratively updated distribution of the training data. This distribution update ensures that instances misclassified by the previous classifier are more likely to be included in the training data of the next classifier [22]. We will use adaboost because of its properties. Our approach uses adaboost in this way:

If we consider the steps of our approach then adaboost lies at the fifth position (our complete approach is described in experiment setup section). After applying PCA we apply those reduced features to adaboost in combination with classification algorithm and try to generate our desired results. Adaboost work in following ways:

```
Input: A set of d class label training tuples, D;
       Number of rounds, k;
       A classification learning scheme;

Method:
  • Initialize the weight of each tuple in D to 1/d;
  • For i = 1 to k do // for each round:
  •       Sample D with replacement according to the
  •       tuple weights to obtain    ;
  •     Use training set    to derive a model,   ;
  •     Compute error(   ), error rate of
  •     If error(   ) > 0.5 then
  •         Reinitialize the weights to 1/d
•        Go back to step3 and try again;
•    End if
•      For each tuple in      that was correctly
      Classified do
•       Multiply the weight of the tuple by
        error ( $_i$)/(1—error( $_i$));
•    normalize the weight of each tuple;
•    end for
```

Output: Composite model

To use composite model to classify tuple, X;

```
• Initialize weight of each class to 0;
• For i = 1 to k do // for each classifier:
•      = log(1− error( $_i$)/ error( $_i$));
•   C =    $_i$(X); // get class prediction for X
•   Add      to weight for class c
•   end for
•   Return the class with the largest weight;
```

Where error ($_i$) = $\sum_j^d$ × err ( ). [23]

Above steps explain the working of adaboost in our procedure. Our final goal is to combine this algorithm with following classification algorithms and find the best combination for all attacks. Adaboost is good for IDS because datasets are heterogeneous mixture of categorical and continuous types.

b) Bagging:

Bagging is a method for improving results of machine learning classification algorithms. This method was formulated by Leo Breiman. Its name was deduced from the phrase "bootstrap aggregating". Diversity in bagging is obtained by using bootstrapped replicas of the training data: different training data subsets are randomly drawn with replacement from entire training data. Each training set is a bootstrap sample because sampling with replacement is used. Some of the original tuples of D may not be included in , where as others may occur more than once. A classifier model $_i$ is learned for each training set . To classify an unknown tuple, X, each classifier $_i$ returns its class prediction, which counts as one vote. The bagged classifier Icounts the votes and assign the class with the most votes to X. The increased accuracy occurs because the composite model reduces the variance of the individual classifiers. Bagging works in following way:

Input: A set of d training tuples, D;
Number of models, k;
A learning scheme;

Method:

```
• For i = 1 to k do // create k models:
•     Create bootstrap sample,     , by sampling D
      with replacement;
•     Use     to derive a model,   $_i$;
•   end for
```

To use the composite model on a tuple, X:

```
• if classification then
•   Let each of the k models classify X and return
    the majority vote;
• if prediction then
•   let each of the k models predict a value for X
    and return the average predicted value;
```

Output: A composite model

Above steps define the working of bagging. This is what we are using in our approach. Like adaboost we apply bagging at the fifth position in our approach by replacing adaboost with it, for generating a healthy comparison between different combinations [5]. And for doing this we are using RapidMiner5.0 tool which integrate meta-modeling analysis while other simulating tools are not successful to do so.

5.CLASSIFICATION

Classification defines the task of data analysis, where a model or a classifier is constructed to predict categorical labels. Classification can be described as a supervised learning algorithm in the machine learning process. In classification a given set of records is divided into training and test datasets. The training dataset is used in building a classification model, while test data is used in validating the data models. There are number of classification algorithms. In our experiment we are using three of them i.e. SVM, decision tree and Bayesian network.

a) Support Vector Machine:

One of the novel techniques of intrusion detection is support vector machine. A SVM maps input feature vectors into a higher dimensional feature space through some nonlinear mapping. SVMs are powerful tools for providing solutions to classification, regression and density estimation problems. Computing the hyper plane to separate the data points i.e. training a SVM leads to quadratic optimization problem. In other words, training sets become nonlinear to establish a relation. Under nonlinear situation we should use nonlinear map to map the input space to some certain feature space [6]. For this we use a kernel function. So k can be defined to all of the training samples as
K (x_i,x_j) = < ϕ (x_i).ϕ (x_j) >, where (x_i,x_j) are inner products and ϕ is a mapping from x to a feature space F.
Problem under nonlinear situation is:

$$\min_{w \cdot b \cdot \xi} \frac{1}{2}\sum_{i=1}^{L} \sum_{j=1}^{L} y_i y_j a_i a_j \text{ K } (x_i,x_j) - \sum_{j=1}^{L} a_j$$

s.t. $\sum_{i=1}^{L} y_i a_i = 0$
$0 \leq a_i \geq c$, i = 1,2….l

Final decision function is:

F(x) = sgn(w^*.K (x_i,x_j) + b^*).

There are many kernel functions; some of them are polynomial, ANOVAs, Gaussian, sigmoid etc. The user may provide one of these functions at the time of training classifier, which selects support vectors along the surface of this function. SVM classify data by using these support vectors which are members of the set of training inputs that outline a hyper plane in feature space. Selecting a kernel plays an important role in SVM because it can increase or decrease the performance of the classifier [21]. In our experiment we are applying SVM in combination with adaboost and bagging, and compare the results with other combinations. We are taking Radial as a kernel for our evaluation (explanation of the kernel is giving in experimental section).

b) Decision trees:

Another classification algorithm in data mining is decision tree induction. Classification algorithm is inductively learned to construct a model from a pre-classified dataset. Each data item is defined by the values of the attributes. Classification may be viewed as mapping from a set of attributes to a particular class. The decision tree classified the given data items using the values of its attributes. The decision tree initially constructed from a set of pre-defined data. The main approach is to select the attribute, which best divides the data items into their classes. According to the values of these attributes the data items are partitioned. This process is recursively applied to each subset of data items. The process terminates when all the data items in current subset belongs to the same class. A node of a decision tree specifies an attribute by which the data is to be partitioned. Each node has a number of edges, which are labeled according to a possible value of the attribute in the parent node. An edge connects either two nodes or a node and leaf.

Induction of the decision tree uses the training data, which is described in the terms of attributes. The main problem here is deciding the attribute, which will best partition the data into various classes [6]. To classify an unknown object, one starts at the root of the decision tree and follows the branch indicated by the outcome of each test until a leaf node is reached. The name of the class at leaf node is the resulting classification. Decision tree works well with large datasets. Generalization accuracy of the decision trees another useful property for intrusion detection model. In our experiment we are applying decision tree in combination with adaboost and bagging, and compare the results with other combinations.

c) Bayesian networks:

A Bayesian network is a model that encodes probabilistic relationships among variables of interest. This technique is generally used for intrusion detection in combination with statistical schemes, a procedure that yields several advantages, including the capability of encoding interdependencies between variables and of predicting events, as well as the ability to incorporate both for prior knowledge and data. Bayesian classifier has exhibited high accuracy and speed when applied to large databases.

Naïve Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called class conditional independence. It is made to simplify the computations involved and, in this sense, is consider "Naive". Naïve Bayesian classifiers allow the representation of dependencies among subsets of attributes [23]. Though the use of Bayesian networks has proved to be effective in certain situations, the results obtained are highly dependent on the assumptions about the behavior of the target system, and so a deviation in these hypotheses leads to detection errors, attributable to the model considered [24]. In our experiment we are using Naïve Bayesian classification in combination with adaboost and bagging, and compare the results.

6.LAYERED APPROACH

Layered-based intrusion detection system gets its motivation from Airport security model, where a number of security checks are performed one after the other in sequence. Similar to this model, the layered intrusion detection system represents a sequential layered approach and is based on ensuring clandestinely, credibility and availability of data or is significant services over a network.

The goal of using a layered model is to reduce computation and the overall time required to detect anomalous events. The time required to detect an intrusive event is significant and can be reduced by eliminating the communication overhead among different layers. This can be achieved by making the layers autonomous and self-sufficient to block an attack without the need of a central decision maker. Every layer in layered intrusion detection system framework is trained separately and then deployed sequentially. We define four layers that correspond to the four attack groups mentioned in the dataset. They are probe layer, DOS layer, U2R layer and R2L layer. Each layer is then separately trained with a small set of relevant features. Feature selection or reduction is important for layered approach and discussed in next section. In order to make the layers independent, some features may be present in more than one layer. The layers essentially act as filters that block any anomalous connection, thereby eliminating the need of further processing at subsequent layers enabling quick response to intrusion. The effect of such a sequence of layers is that the anomalous events are identified and blocked as soon as they are detected [2].

7.FEATURE REDUCTION METHODOLOGY

We are using Principle Component Analysis for feature reduction. It is a way of identifying patterns in data and expressing data in such a way as to highlight their similarities and differences. PCA is a powerful tool for analyzing data. Another advantage of PCA is, once you have found these patterns in the data, you can compress data i.e. by reducing the number of dimensions without much loss of information. A generalized view of PCA must contain the following steps-For apply PCA our basic need is a dataset. Here in our experiment this dataset is NSL-KDD dataset, which can be applied to PCA through a layered approach. Secondly for PCA to work properly, you have to subtract the mean from each of the data dimensions. The mean subtracted is the average across each dimension. This produces a dataset whose mean is zero. After this covariance matrix was calculated. Then eigenvectors and Eigen values were calculated of covariance matrix. So, by this process of taking the eigenvectors of covariance matrix, we have been able to extract lines that characterize the data. And rest of the steps involves transforming the data, so that it can be expressed in terms of those lines. Now notion of data compression and reduced dimensionality come into existence. Here we find the appropriate Principle component. The eigenvector with the highest Eigen value is the principle component of the dataset. What needs to be done now is; need to form a feature vector which is just a fancy name for a matrix of vectors. This is constructed by taking the eigenvectors that you want to keep from the list of eigenvectors and forming a matrix with these eigenvectors in the columns.

Feature vector = $(eig_1, eig_2 \ldots\ldots eig_n)$

Now the final step of PCA is deriving the new dataset. Once we have chosen the components that we wish to keep in our data and formed a feature vector, we simply take the transpose of the vector and multiply it on the left of the dataset, transposed.

Final data = RowFeatureVector × RowDataAdjust

Where RowFeatureVector is the matrix with the eigenvectors in the columns transposed so that the eigenvectors are now in rows, with the most significant eigenvector at the top and the RowDataAdjust is the mean adjusted data transposed i.e. the data items are in each column, with each row holding a separate dimension [9]. We use it in our experiment to increase the performance of our model.

8.EXPERIMENTAL SETUP AND RESULTS

a) Intrusion Data:

In Our experiment we use NSL-KDD datasets. Due to some inherent problems of KDD'99 dataset, NSL-KDD comes into existence. The number of records in

The NSL-KDD train and test sets are reasonable. This advantage makes it affordable to run the experiments on the complete set without the need of randomly selecting a small portion. The training datasets of NSL-KDD are similar to KDD'99 and consist of approximately 4,900,000 single connection vectors each of them contains 41 features and is labeled as normal or attack type with exactly one specific attack type.

NSL-KDD datasets have following advantages over the original KDD datasets:

- Train sets are free from redundant records, which results into unbiased classifier.
- Duplicate records are not present in the test set, which results into unbiased performance of learner.
- The classification rates of distinct machine learning methods vary in a wider range, which makes NSL-KDD datasets more efficient to have an accurate evaluation of different learning techniques.

During attack, an attacker sets up a connection between a source IP address to a target IP address and sends data to attack the target. The simulated attacks fall in one of the following categories:

- DOS (Denial of service)
- Probing attack
- U2R (User to root attack)
- R2L (Remote to local attack)

Following tables shows all the features found in the connection. For easier referencing each feature assign a label (A to AO). [9]

Network Data Feature Label-

Label	Network data label
A	Duration
B	protocol_type
C	service
D	flag
E	src_bytes
F	dst_bytes
G	land
H	wrong fragment
I	urgent

Table (a) Basic features of individual TCP connection.

Label	Network Data Feature
J	Hot
K	num_failed_logins
L	logged_in
M	num_compromised
N	root_shell
O	su_attempted
P	num_root
Q	num_file_creations

R	num_shells
S	num_access_files
T	num_outbound_cmds
U	is_hot_login
V	is_guest_login

Table (b) content features within a connection suggested by domain knowledge.

Label	Network Data Features
W	Count
X	sev_count
Y	serror_rate
Z	sev_serror_rate
AA	rerror_rate
BB	srv_rerror_rate
AC	same_srv_rate
AD	diff_srv_rate
AE	srv_diff_host_rate
AF	Dst_host_count
AG	Dst_host_srv_count
AH	Dst_host_same_srv_rate
AI	Dst_host_diff_srv_rate
AJ	Dst_host_same_src_host_rate
AK	Dst_host_srv_diff_host_rate
AL	Dst_host_server_rate
AM	Dst_host_srv_serror_rate
AN	Dst_host_rerror_rate
AO	Dst_host_srv_rerror_rate

b) Hardware requirement and analysis tool:

In our experiment, we have written a Java program for preprocessing the data. We have also applied a layered approach for intrusion detection as a result we could generate generates four different layers of attack namely normal with DOS, normal with Probe, normal with U2R and normal with R2L attack datasets. The main purpose of intrusion detection model is to classify the datasets into one of the four attack types. The dataset of our experiment contains 32,357 records. Which will accordingly distributed to each attack class type, such as we have 9461 records for DOS, 7933 records for probe, 8090 records for R2L and 6873 records for U2R.

To perform our experiment we have used RapidMiner5.0 tool [25]. We have developed a Java program for data formatting and implementing a layered approach. We used RapidMiner5.0 because it is a well accepted simulation tool in the world of research. It is available as a stand-alone application for data analysis. RapidMiner5.0 is also capable in overcoming some loopholes of WEKA. We used it because of its some unquestionable properties [25]- Data integration, analytical ETL, data analysis and reporting in one single suite.

- Powerful but intuitive graphical user interface for the design of analysis process.
- Repositories for process, data and metadata handling.
- Only solution with metadata transformation.
- Only solution which supports on-the-fly error recognition and quick fixes.
- Complete and flexible.

We have performed our experiments on a desktop running with Intel(R) core(TM) i3, CPU @ 2.10 GHz and 3.00 GB RAM under exactly the same conditions. For our results, we have given the precision, recall and F-measure and not the accuracy alone as the given dataset. While precision, recall and F-measure are not dependent on the size of the training and the test samples. They are defined as-

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

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

$$\text{F-measure} = \frac{(1 + \beta) * \text{recall} * \text{precision}}{\beta^2 * (\text{recall} + \text{precision})}$$

Where TP is true positive, FP stand for false positive, FN is false negative and β corresponds to the relative importance of precision versus recall and is usually set to 1.

The accuracy of the algorithms is measured by the percentage of false positive and false negative that was generated during the classifying process. A higher false negative implies that the recall of the classifier is lower. “Based on the evaluation results, best combinational algorithm for each category is chosen”.

c) Experimental results:

It was already been stated that we have used RapidMiner5.0 as our simulation tool. And from the above mentioned details it becomes clear what we really want to propose? Our proposed work was accomplished in a virtue of following steps-

- Select the attack repository.
- Set the role of the attribute.
- Normalize the dataset.

- Apply PCA for feature reduction, and as a result we get 12 reduced features from 41 features.
- Split the r repository with the ratio of 6:4; 60% of our dataset is used for training purpose and 40% is used for testing, then
- Apply our selected meta-modeling algorithms i.e. adaboost and bagging.
- Integrate our selected classification algorithms with meta-modeling algorithms.
- At last we obtained the results for each combination and compared them to find the most appropriate combination for all attacks.

Let's take a closer look by taking each combination one by one.

(a) Detecting attacks using layered approach by applying adaboost with support vector machine:

Here, first of all we apply our layered approach i.e. applying our attacks repositories one by one to this combination. While applying SVM to our experiment, we choose Radial as our kernel function (term kernel is already explained in our section V (a)). Radial basic function is a nonlinear kernel function; it's a combination of Gaussian exponential function. Radial basic function follows WINNER TAKE TYPE approach. This can be expressed as-

$$k(x, x_i) = \exp\left\{-\frac{|x - x_j|^2}{\sigma^2}\right\} = \exp(-gama \cdot |x - x_i|^2)$$

There is no proper theory for how to choose a kernel function. Another term which influence the performance of SVM is the c-value which is the complexity constant of SVM. We set c = 0.0, we set it to 0.0 because lower c-value generates more robust and strongly generalized model. The table1 below shows the results of our combination with different attack datasets.

Attack Database	Precision	Recall	F-measure	Accuracy
DOS	98.91%	92.63%	95.66%	97.67%
Probe	94.34%	82.71%	88.14%	96.89%
R2L	99.06%	99.20%	99.13%	98.52%
U2R	1.11%	0.00%	0.00%	99.49%

Table 1- Adaboost with Support Vector machine

We have also applied adaboost with naïve Bayesian and Decision Tree to compare our results and find the best combination. And our combinations generated following results.

Attack Database	Precision	Recall	F-measure	Accuracy
DOS	76.14%	84.11%	79.93%	82.27%
Probe	72.93%	77.31%	75.05%	93.00%
R2L	92.67%	90.38%	91.51%	85.94%
U2R	7.14%	6.25%	6.66%	98.98%

Table2- Adaboost with Naïve Bayesian

Attack Database	Precision	Recall	F-measure	Accuracy
DOS	96.30%	74.22%	83.88%	92.05%
Probe	92.64%	49.54%	64.55%	92.59%
R2L	97.92%	98.64%	98.28%	97.10%
U2R	14.29%	6.25%	8.69%	99.24%

Table 3- Adaboost with Decision Tree

(b) Detecting attacks using layered approach by applying bagging with support vector machine:

As we applied SVM with adaboost, now we apply bagging with SVM using the same parameter configuration. SVM and bagging are explained in section V.Table4 shows our obtained results-

Attack Database	Precision	Recall	F-measure	Accuracy
DOS	98.28%	94.41%	96.30%	97.99%
Probe	93.96%	84.21%	88.81%	97.23%
R2L	99.06%	99.02%	99.03%	98.52%
U2R	1.11%	0.00%	0.00%	99.49%

Table 4- Bagging with support vector machine

We have applied bagging with Decision tree and Naïve Bayesian to compare our results and find the best combination. And our combinations generates, following results-

Attack Database	Precision	Recall	F-measure	Accuracy
DOS	95.33%	85.51%	90.15%	94.82%
Probe	83.33%	84.96%	84.14%	95.53%
R2L	98.90%	98.59%	98.74%	97.88%
U2R	11.11%	4.76%	6.66%	99.32%

Table 5- Bagging with Decision Tree

Attack Database	Precision	Recall	F-measure	Accuracy
DOS	82.07%	77.95%	79.95%	89.17%
Probe	69.41%	73.68%	71.98%	91.79%
R2L	96.23%	73.02%	83.03%	74.66%
U2R	16.42%	52.38%	25.00%	98.40%

Table 6- Bagging with Naïve Bayesian

From the above generated results it can be easily concluded that our approach is good with almost very combination but it is clearly visible that SVM with both adaboost and bagging provides best accuracy detection rate. Best results are attained especially when used with bagging. Our approach has not only increased the performance but also provided the highest accuracy rate i.e. 97.99% for Dos, 97.23% for Probe, 98.52% for R2l and 99.49 % for U2R.

9.CONCLUSION

In this paper, we have addressed aspects of intrusion detection system and made it more robust and efficient; one aspect is accuracy and other is performance. Our experimental results have shown that, "A layered approach for intrusion detection using meta-modeling with classification techniques" is very effective in improving the attack detection rate. The area for future research include, finding the robustness of our system with noisy dataset.

REFERENCES

[1] Shun-ichi Amari and Si Wu, "Improving support vector machine classifiers by modifying kernel function", RIKEN Brain Science Institute Japan.

[2] Kapil Kumar Gupta, Baikunth Nath and Ramamohanaroo kotagiri, "A layered approach using conditional random fields for intrusion detection", IEEE Tranc. on Dependence and secure computing, Vol.7, 2010

[3] G.MeeraGandhi, Kumaravel Appavoo and S.K Srivasta, "Effective network intrusion detection using classifiers decision trees and decision rules",Int. J. Advanced network and application, Vol2, 2010

[4] Bernhard scholkopf, Kah kay Sung, Chris Burges, Federico and other, IEEE Transactions on signal processing, Vol. 45 , 1997

[5] Richard Machlin and David Opitz, "An empirical Evaluation of bagging and boosting", National conference on A.I, providence Rhode Island 1997.

[6] Sandy Peddabachigari, Ajit Abraham and Johmson Thomas, "Intrusion detection system using decision trees and SVM", Oklahoma state university USA.

[7] Huy Anh Nguyen and Deokjai choi, "Application of data mining to network intrusion detection", Korea.

[8] Weiming Hu, Wei Hu and Steve Maybank, "Adaboost based algorithm for network intrusion detection", Tranc. On system man and cybernetics, 2008.

[9] Shilpa Lakhina, Sini Joseph and Bhupendra Verma, "Feature reduction using PCA for effective Anomaly-based intrusion detection on NSL-KDD", Int. J. of engineering science and technology, 2010

[10] Snehal A.Mulay, P.R Devale and G.V Garje, "Intrusion detection using SVM and decision tree", Int. J. of computer application, 2010

[11] J.Vishumathi and K.L Shunmuganathan, "A computational intelligence for evaluation of intrusion detection system ", Indian J. of science and technology, Jan 2011

[12] Ritu Ranjani Singh, Neetesh Gupta and Shiv Kumar, "To reduce the false alarm in intrusion detection system", Int. J. of soft computing and engineering, May 2011

[13] Defending yourself: IEEE software September/October 2000 tutorial

[14] Xunyi Ren, Ruchuan Wang and Hejunzhou, "intrusion detection system method using protocol classification and Rough set based SVM", www.ccsenet.org/journal.html,2009

[15] Peyman Kabiri and Ali A. Ghorbani, "Research on ID and Response:A survey ", Int. J. of network security, 2005

[16] "Bagging and boosting", Srihari@cedar.buffalo.edu

[17] Tich Phuoc Tran, Longbing cao, Dat Tran and Cu ong Duc Nguyen, "Novel ID using probabilistic Neural network and adaptive boosting", Int. J. of CS and information security, 2009

[18] NSL-KDD.html

[19] Lindsay I Smith A tutorial on Principal Components Analysis February 26,2002.

[20] K. Fukunaga. "Introduction to Statistical Pattern Recognition" Academic Press Professional, Inc., San Diego, CA, USA,1990.

[21] A. B. M. S. Ali, A. Abraham. An Empirical Comparison of Kernel Selection for Support Vector Machines. 2nd International Conference on Hybrid Intelligent Systems: Design, Management and Applications, The Netherlands, 2002

[22] Robi Polikar, "Ensemble based systems in decision making", IEEE circuit and system magazine, 2006

[23] Jaiwei Han and Micheline Kamber, Elseiver book

[24] P.Garcia- Teodoro, J.Diaz- Verdejo, "Anomalyy network intrusion detection: Techniques, systems and challenges", www.elsevier.com , 2009

[25] Rapid-I.com

Ankita Gaur had received her B.E degree in Information Technology from Maharana Pratap Callege of Technology (MPCT), University RGPV Bhopal, India, in 2008. She is currently a PG research Scholar of Software Engineering from Lakshmi Narain College of Technology (LNCT), University RGPV Bhopal, India.

Prof. Vineet Richariya HOD of Computer Science and Technology at LNCT Bhopal, India. He did his M.Tech. (Computer Science and Engineering) from BITS Pilani in year 2001. He did his B.E (Computer Science and Engineering) from Jiwaji University, Gwalior, India in year 1990.