Performance Evaluation of Attack Detection Algorithms using Improved Hybrid IDS with Online Captured Data

Vinita R. Shewale
PG Student
SSVPS's BS Deore College of Engineering,
Dhule, 424005,
India

Hitendra D. Patil, PhD
Professor and Head,
SSVPS's BS Deore College of Engineering,
Dhule, 424005,
India

ABSTRACT
The role of Intrusion Detection System (IDS) is having a very essential role in network Security. As the need of internet is increasing day by day, the importance of security is also increasing. A traditional intrusion detection technology indicates the limitations like low detection rate, high false alarm rate and so on. Performance of the classifier is a necessary concern in terms of its effectiveness; also number of feature to be examined by the IDS should be improved. In this, hybrid IDS is applied using Snort with J48 Graft Decision tree algorithm, J48 Graft Decision tree with Pruning using feature selection and Naïve Bayes algorithm. In J48 Graft Decision tree with pruning, only discrete value attributes for classification are considered and for Naive Bayes redundant records are removed with feature selection. KDDCup'99 dataset is used to train and test the classifier. The performance of the classifiers is also tested on dataset created by capturing online packets which classifies packet as either normal or anomaly. Results and analyses show that, J48 Graft decision tree with pruning and Naive Bayes approach is giving better results with enhanced accuracy than existing classification techniques.

General Terms
Intrusion Detection system, Network Security, Misuse based system, Training, Decision tree.

Keywords
Classification Algorithms, Pruning, Anomaly Detection, Accuracy, KDD, Hybrid, Snort.

1. INTRODUCTION
The computer networks are expanding day by day and number of internet users is also increase. The vast amount of attacks over the Internet makes computer users and many organizations under potential violation of security. As a result it is strongly required to protect network systems, organizations from intrusions. The concept of intrusion can be defined as an attempt to invade a system and violate aspects such as integrity, availability, confidentiality or the quality of the services in the system. The preventive measures employed by organizations to protect network systems with password, firewalls or other mechanisms do not satisfy the wholly protection because they are unable to detect some variants of attack [1].

Data Mining is the use of algorithms to extract the information and patterns derived from the knowledge discovery process in databases. Classification is the process of assigning task to predefined instances or classes. It is mainly referred to as supervised learning because the classes are determined before examining the data. An Intrusion Detection System (IDS) monitors the activities happening inside a system for suspicious behavior or patterns that indicate system attack or misuse.

Intrusion detection system can be classified as: (i) misuse based system, (ii) anomaly based system, and (iii) hybrid system. Misuse based IDS simply matches the given pattern, and a database of known attack patterns and produce very low false positive (FP) for known attacks. It requires the signature of the rules to be already stored in signature database. For this system, snort is used as misuse based system, open source network IDS [4]. But, even though a well-known attack is changed slightly then, detector is unable to detect that one. Anomaly based system collects the data related to the behavior of system or user and then applies statistical tests to the observed behavior, which determines whether that behavior is legitimate or not. For this, J48 Graft Decision tree with Pruning using feature selection and Naïve Bayes algorithms are used. But, hybrid IDS is the combination of misuse and anomaly detection system. So, it is able to detect both known and unknown attacks with few false positives.

Experimental results are generated on KDDCup'99 data set. These results have demonstrated that this classifier model is much more efficient in detecting network intrusions, compared to the other classification techniques. Naive bayes classifiers have worked quite well in very much complex real-world situations. Decision trees are created using algorithms for building the tree iteratively in a very short period of time. Creating decision trees requires a pre-classified dataset for the algorithms to learn patterns in the data. Decision tree are used mainly for classification i.e in supervised learning. They are comparatively efficient and capable of working with huge volume of data.

2. RELATED WORK
A lot of research has been carried out in the field of intrusion detection system. J. Marin et al. [2] described the hybrid approach that they employed which starts with the application of expert rules for reducing the dimensionality of the data, followed by the initial clustering of the data and then it does subsequent refinement of the cluster locations using the competitive network called as Learning Vector Quantization. M. Shyu et al.[3] proposed Principal Component Anomaly based detection scheme. It uses principal component analysis as an outlier detection scheme to detect intrusions. But, Covariance matrix needed is difficult to be evaluated in accurate manner.

In 2009, M. Aydin et al.[4] discussed a hybrid IDS by combining 2 approaches in one system which is obtained by combination of packet header anomaly detection (PHAD) and network traffic anomaly detection (NETAD) which are anomaly-based IDSs with the signature-based IDS Snort that is an open-source project. But, it has to suffer from high false alarm rate.

By using a hybrid approach supported a pattern matching engine and a neural network function parallel, C. Amza et al.[5] planned a unique Intrusion Detection System that boost the detection accuracy. This approach depends on the Netpy traffic observance and analysis tool that they developed. Netpy is the application involves the help of the administrator by observing the state of a network which detects known as well as unknown attacks. But, it is not able to detect large number of attacks.

3. DATASET

The KDD Cup 1999 dataset was derived from the DARPA [9] Intrusion detection evaluation program of 1998 that is prepared and managed by MIT Lincoln Laboratory. DARPA dataset contains network traffic including the entire payload of each packet which was recorded in tcpdump format and provided for evaluation purpose [10]. This tcpdump data is given as an input to Snort to detect known attacks.

KDD Cup 99 dataset was a collection of simulated raw TCP dump data, used for evaluation of Anomaly based system. For training any machine learning algorithm, it requires previously stored data. There is necessity of appropriate size of data to train and test the models. KDDCup'99 dataset [8] is an intrusion related data with almost fifty lacks of records which is prepared by seven weeks of network traffic and two weeks of testing data consisting around 2 lacks of records. The training data is containing 22 diverse categories of attacks and testing data have total of 39 different attacks by adding 17 new types of attacks other than training dataset. This used 10% of this KDDCup'99 dataset containing about 4,94,020 records having 41 attributes of which some are discrete value attributes and others attributes are continuous valued. The 22 types of attacks are categorized into 4 broad categories like DoS, U2R, probe and R2L.

4. METHODOLOGY

4.1 Snort

Misuse-based IDS used in hybrid IDS is the open-source project Snort. Snort [7] uses a signature based detection technique which is rule based. It captures network data packets, preprocesses them and checks their contents with the predefined attack patterns (using Snort rules) for any correlation [6]. Example of Snort rule is given as-

Example - alert tcp any any -> any 20 (msg:"TEARDROP"; sid:10007)

There are four fields in writing rule format. First field is action field which decides action to be taken i.e. alert which detects attack. Second field is the name of protocol used i.e. tcp, udp, icmp etc. Next field is source or destination address and port which act as direction indicator. Forth and last field is rule option which gives message to be displayed that what type of attack is detected and other options may be dsize, flags, content or session. A lot of options are available to configure snort as a Network IDS. The rules for detection of attacks are updated in rules.rules file based on port number of Snort by configuring it.

4.2 Decision Tree Concept

A decision tree is a classification method which uses precise approach to compare the competing alternatives with top down approach. Instances are organized from root node to leaf node. Each node present in the tree specifies a test of particular attribute of the instance, and every branch sliding from the node relates to one of the possible values for that attribute. Creating an optimal tree is maximum times computationally not feasible, as the number of possible trees increases exponentially with the set of attributes. Decision trees provide easy set of rules that can categorize new data. When the decision tree is built from these features, the rules for information characterization can be used to identify and classify new data with help of classification.

The classification process starts from the root node in which it is testing the attribute related to that node, and then sliding down the tree related to the value of attribute. This process is repetitive for the sub-tree rooted at the new node. It can easily transform a decision tree to a group of rules by mapping them one by one from the root node to the leaf nodes. The main advantage of decision trees over other diverse classification techniques is that they produce a set of rules that are transparent, easy to understand and automatically constructs decision tree from a given dataset. Decision tree work with both nominal and numeric attributes.

4.3 J48 Graft Decision Tree Algorithm

J48 graft decision tree algorithm is the Java implementation of C4.5 algorithm, introduced by Ross Quinlan. It is used to produce decision tree for purpose of classification. This uses the concept of information gain ratio as splitting criteria that can also deal with continuous value attributes along with discrete value attributes. It can make use of various techniques of pruning to stay away from over-fitting of decision tree. Normalized information gain ratio can be used as the splitting criteria in J48 Graft.

The algorithm picks out the best attributes which partitions the dataset into its subsets which contain either one class or the other. The attribute with highest information gain can be used to partition the data. The algorithm is repetitively applied to its sub-tree. The leaf node in the decision tree created by it represents any one class label. The splitting process brings to a halt when the number of instances to be split is below a certain threshold. In this study, all the 41 attributes for the creation of decision tree are considered.

Gain measures however well a given attribute separates training record into output class label. The calculation of information gain ratio can be done as follows: To describe information gain correctly, one measure is commonly used in information theory, well-known as entropy, which makes different the impurity of a random group of samples. Suppose that S is a set included by sample of data s, and contains n C_i (i = 1, . . . ,n) with different labels, s_i is the sample of C_i type in S set, P_i is the probability of any sample possibly that belonging to C_i. Entropy assesses the level of impurity in a cluster of samples. To measure the effectiveness of an attribute in classifying the training data, it uses information gain that is desired reduction in entropy caused by partitioning related to this attribute. Entropy can be specified as,

$$Entropy = -\sum_{i=1}^{n} P_i \ log_2 P_i \qquad (1)$$

Gain(S,A) of an attribute A, relative to a collection of samples [12] S, can be given as,

$$Gain(S, A) = Entropy(S) - \sum_{j \in Values\ (A)} \frac{|S_j|}{|S|} Entropy(S_j) \qquad (2)$$

Where values(A) is the set of all potential values for the attribute A, and Sj is the subset of S for which attribute A has value j. The value of Gain(S,A) is the number of bits saved when encoding the target value of random member of S, by

knowing the value of attribute A. For sample set S, on the assumption that attribute A has j different discrete values, then the partition information divided by attribute A can be given by the following formula,

$$SplitInfo(S,A) = -\sum_{k=1}^{c} \frac{|S_k|}{|S|} log_2 \frac{|S_k|}{|S|} \quad (3)$$

The Gain Ratio for an attribute A is calculated as,

$$Gain\ Ratio(S,A) = \frac{Gain(S,A)}{SplitInfo(S,A)} \quad (4)$$

4.4 J48 Graft with Pruning

Due to usage of compact stopping criteria, it creates small and not overfitted decision trees. At the other hand, it also make use of insecurely stopping criteria that results into the creation of large and complex decision tree which are over-fitted to the training set. To resolve this issue, pruning method is build up. Pruning is a way that condenses the size of decision tree by the elimination of portion of the tree that is not contributing to accuracy of classifier. The use of pruning method can improve the performance of a decision tree, in case of noisy domain. Idea behind pruning was to find out the attributes which doesn't add any value in classifying the class label. In the given KDD data set, there are different attributes which not participate in classifying the class labels, as their range values lies into its both buckets like normal and anomaly.

4.5 Naive Bayes Classifier

A Naive Bayes classifier is a probabilistic classifier based on application of Bayes theorem having strong (naïve) independence assumptions. A Naïve Bayes classifier pretends that the presence (or absence) of a specific attribute of a class is not associated with the presence (or absence) of other feature. In that classification, it has a hypothesis that the given information belongs to a specific class. Then calculate the probability for the hypothesis for being true.

Depending on the accuracy the probability model, training of Naïve Bayes classifiers can be done in effective way. Bayes Theorem can be expressed as:

$$P(H|X) = P(X|H)P(H)/P(X) \quad (5)$$

Let *X* be the data record, H be some hypothesis showing data record *X*, which belongs to a particular class *C*. For classification, there is need to determine $P(H|X)$, which is the probability that the hypothesis *H* holds, given an already observed data record i.e. *X*. $P(H|X)$ is the posterior probability of *H* conditioned on *X*. On the contrary, $P(H)$ is the prior probability. The posterior probability $P(H|X)$, is based on more information such as background or previous knowledge than the prior probability$P(H)$, which is independent of *X*. Similarly, $P(X|H)$ is posterior probability of *X* conditioned on hypothesis *H*. Bayes theorem is useful because it provides way for calculation of posterior probability $P(H|X)$ from $P(H)$, $P(X)$, and $P(X|H)$ [11].

Algorithm:

The algorithm consists of two phases – learning and classifying phase.

Input: *D*: Data set having *n* Connection Records

C: Set of classes e.g. {Normal; Anomaly}

X: Data record that is to be classified

H: Hypothesis (that *X* is classified into Class *C*)

Output: The predicted class C_{NB} where *X* should be classified into either normal or anomaly.

Algorithm:

Learning Step

For $k \leftarrow 1$ to no. of classes

Step 1: Calculate the prior probabilities of Class C

$C_k\_count \leftarrow no.of\ D_j\ where\ D_j.class_{label} = k;$

$P(C_k) \leftarrow C_k\_count/n;$

Step 2: Calculate prior probabilities of Record X

For each *attribute value* X_m in *X*

$X_m\_count \leftarrow no.of\ X_m\ in\ C_k$

$P(X_m\ |C_k) \leftarrow X_m\_count\ /\ C_k\_count$

EndFor

Step 3: Calculate posterior probability of Record X

$P(X) \leftarrow average(P(X_m|C_k))$

Endfor

Classification Step

Step 4: Determine the required Naive Bayes probability

For $k \leftarrow 1$ to *no_of_classes*

$P(C_k|X) \leftarrow P(C_k|H) * P(C_k)/P(X)$// ***Using Eq. 5*** //

Endfor

Step 5: Get the class with maximum probability

$$C_{NB} = \max(P(C_k|X))$$

For Naïve Bayes, some of the attributes are irrelevant and redundant which results lengthy detection process and reduces the performance of an intrusion detection system (IDS). The motive of this study is to find important reduced input features in building IDS.

4.6 Anomaly based System Design

The Block diagram of Anomaly based intrusion detection is shown in Figure 1.

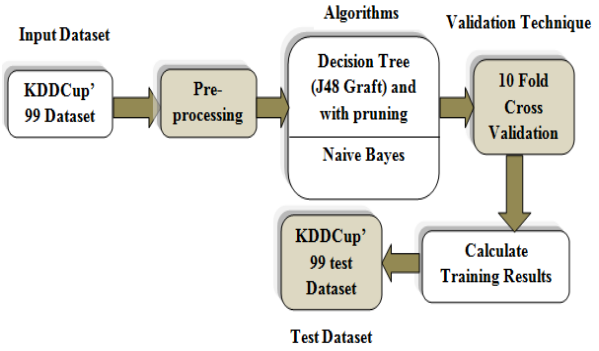

Figure 1: System Flow of Anomaly based System

The first step is collecting input dataset and pre-processes it. The pre-processing steps consist of formatting, cleaning and sampling of data. For training purpose, 10% of KDD dataset is used. In the next step any one approach among the

algorithms is selected along with the 10 fold cross validation technique. The J48 Graft decision tree and Naive bayes algorithm are developed with the help of WEKA APIs. But, for J48 Graft with Pruning and improved Naive Bayes technique, algorithms are used and developed them in JAVA. Along with the 10 fold cross validation technique, the dataset can also be splitted using randomized methods like 90% training and 10 % testing, 80% training and 20% testing and so on. Last attribute that is 42nd attribute in dataset is class label (Normal, Anomaly) to be identified based on classifier used. Based on that, training results are generated and then, tested on KDD Cup'99 test dataset.

5. EXPERIMENTAL ANALYSIS

5.1 Parameters

The confusion matrix is represented by 4 values which are TP, FN, FP and TN

1. **True positive (TP)** - Indicates the instances which are predicted as normal appropriately.
2. **False negative (FN)** - Denotes wrong prediction i.e. it detects instances which are attacks in reality, as normal one.
3. **False positive (FP)** - Gives a hint of the number of detected attacks which are normal in the reality.
4. **True negative (TN)** - Indicates instances which are correctly detected as the attack.

Performance measures can be given as-

Accuracy – It is proportion of correctly classified classes TP and TN over total number of classifications [1] and can be calculated by the formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} * 100\%$$

a) **Sensitivity** - It is nothing but True Positive Rate. It indicates percentage of intrusions correctly detected.

$$Sensitivity = \frac{TP}{TP + FN} * 100\%$$

b) **Precision** - It estimates the probability of a positive prediction that are being correct.

$$Precision = \frac{TP}{TP + FP}$$

c) **F1 score** - It is given as the harmonic mean of precision parameter and sensitivity parameter.

$$F1\ Score = \frac{2 * TP}{2TP + FP + FN}$$

d) **Training time** - It is the time that classifier consumes to build the model on the applied dataset.

5.2 Result Analysis

All experiments are performed in a computer with the configurations of Intel(R) Core i5 CPU 2.30GHz; with 4 GB RAM, and the operating system platform is Microsoft Windows 8. In this study, implementation of two different systems is done so as to make hybrid IDS. First is signature based system and other is anomaly based system.

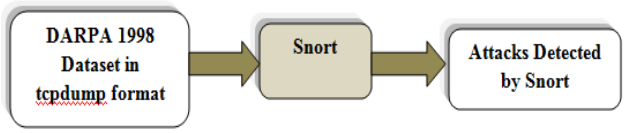

Figure 2:Signature based Detection based on Snort

When signature or misuse based system- Snort as shown in figure 2 was evaluated with the DARPA 1998 dataset and tcpdump data is given as input, results are shown in Table 1. It can be noted in Table 1 that some of the attacks for particular type may get detected whereas some from the same attack type may not get detected.

Table 1. Attacks detected by snort from DARPA'98 dataset

Misuse based Detection	Attacks Detected
Snort	DoS, Ipsweep, Smurf,Rootkit, Spy, Pod, Eject,Sylog, Portsweep, Land, Neptune, Teardrop

The dataset to be used in this experiments is KDDCup'99 labeled supplied as input to the classifier. Out of those, for training purpose, 15000 records are considered and for testing purpose, 10000 records from KDD test dataset are taken for performing experiment. The results of J48 Graft and Naïve Bayes developed using WEKA APIs are compared with the respective modified algorithms for both J48 graft with pruning and improved Naïve Bayes for performance evaluation. In J48 Graft with pruning and Naïve Bayes, this has considered only discrete value attributes like protocol_type, Service, land, flag, logged_in, is_guest_login, is_host_login and class for classification. A random sampling of training and testing parts may generate diverse results in different runs. The results of training as shown in figure 1 are calculated and compared. Table 2 shows result for all 4 algorithms with 10 fold cross validation for training data. First results are calculated on each fold and then average is taken of each fold.

Table 2. Results of training with 10 fold cross validation

Parameters	J48 Graft Decision Tree	J48 Graft with Pruning	Naïve Bayes	Improved Naive Bayes
Accuracy (%)	99.435	99.834	92.715	97.811
Inaccuracy (%)	0.565	0.166	7.285	2.189
Time to Build (msec)	2769	434	2457	583
Sensitivity (%)	99.426	99.872	85.635	97.829
Precision	0.994	0.999	0.852	0.978
F1 Score	0.993	0.998	0.916	0.977

Figure 3 shows rate of accuracy, inaccuracy and sensitivity compared among all 4 classifiers. Figure 4 shows results for precision and F1 Score. J48 Graft with pruning gives better accuracy, precision and f1 score compared with other classifiers and even improved Naive Bayes is having better results than original one.

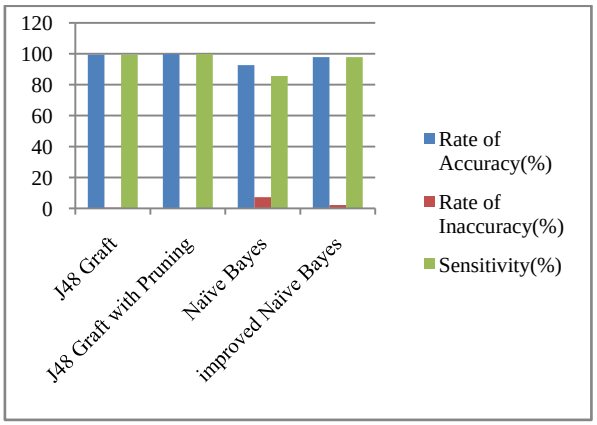

Figure 3: Rate of Accuracy, Inaccuracy and Sensitivity (%) among all four Classifiers

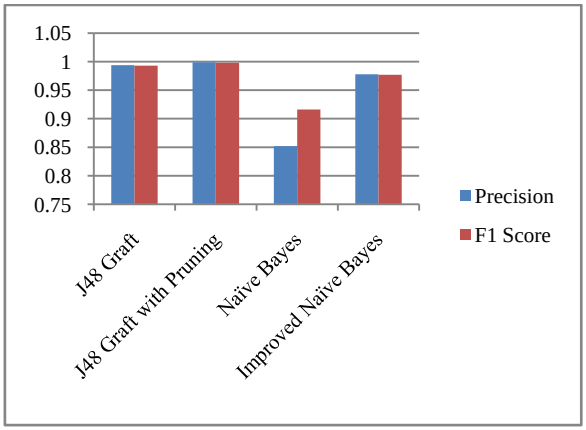

Figure 4: Comparison of Classifiers Based on Precision and F1 Score

Figure 5 shows training time that is time required to build model for all classifiers. Results show that both improved algorithms take less time as compared to other classifiers.

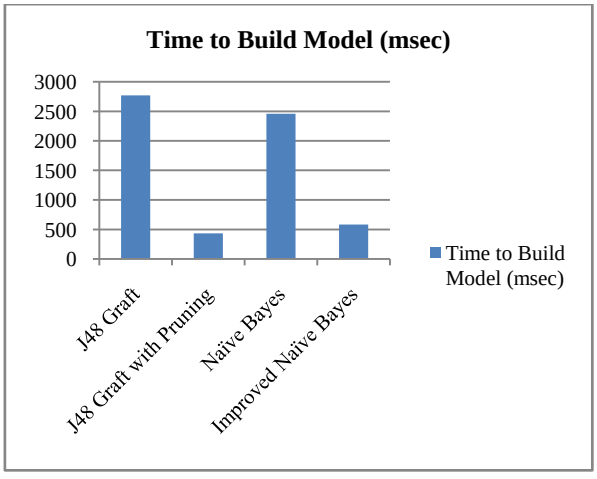

Figure 5: Time to Build Model (msec) between All Four Classifiers

In this work, it is also capturing packets online and creating test dataset based on captured packets. Figure 6 shows the raw packet capturing for which jnetpcap library is used. At first, selection of device interface and creation of packet handler is done to receive packets. Packet handler has to determine type of packet, protocol and feature. Then, scan packet buffer and decode header to be available in readable form and at last close pcap handle.

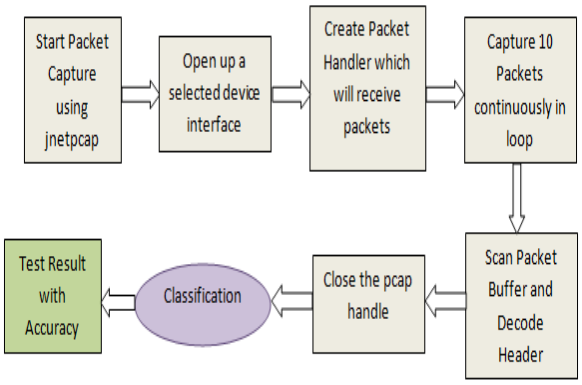

Figure 6: Online packet capture process

After capturing packets, those captured packets can be added into the dataset and the dataset is rearranged by labeling the packets as normal or intrusion. Figure 7 indicates the experimental results of detecting attacks and the normal behavior of captured packets. For each test, number of instances in test varies with value specified in bracket. Both improved algorithms are tested for online traffic test data and it shows more than 99% classifier accuracy for these captured packets.

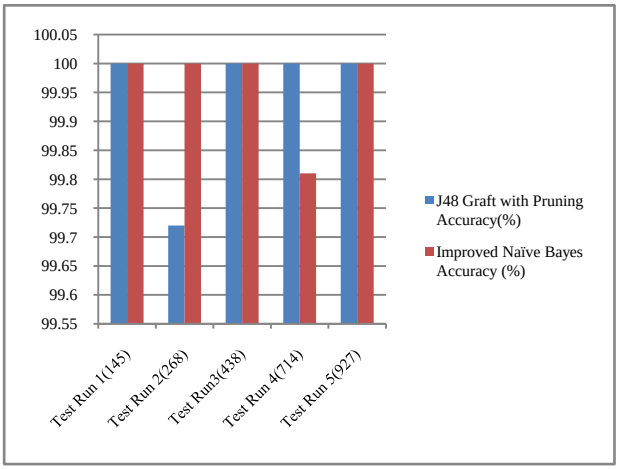

Figure 7: Classifier accuracy (%) for 5 test set

6. CONCLUSION

Intrusion detection is very important part in network security. For the evaluation of intrusive pattern, use of all the 41 features degrades the performance of the classifier and also its time consuming. In this framework of hybrid IDS, only discrete value attributes like protocol_type, Service, land, flag, logged_in, is_guest_login, is_host_login and class for classification are considered. Most of the researchers were used KDD99 dataset. This used KDD dataset as well as created dataset by online packet capturing. The experimental results show that, the intrusion detection algorithm based on snort with J48Graft decision tree with pruning and improved Naïve Bayes is feasible and effective. It increases the

classifier accuracy, precision and even takes less time to train as compared to existing algorithms. Also for online captured data, it achieves more than 99% accuracy. Thus, performance is improved. In future, this hybrid IDS approach can be applied to create large dataset based on capturing packets and try to detect different categories of attack like Probe, U2R, R2L in that dataset.

7. REFERENCES

[1] S. Hussein, F. Ali and Z. Kasiran, "Evaluation Effectiveness of hybrid IDS Using Snort with Naive Bayes to Detect Attacks," *Digital Information and Communication Technology and its Applications (DICTAP),* pp. 256-260, 2012.

[2] J. Marin, D. Ragsdale and J. Surdu, "A Hybrid Approach to the Profile Creation and Intrusion Detection," in *DARPA Information Survivability Conference and Exposition* , 2001.

[3] M. L. Shyu, S. C. Chen, K. Sarinnapakorn and L. Chang, "A novel anomaly detection scheme based on principal component classifier," in *Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop,* Melbourne, FL, USA, 2003.

[4] M. A. Aydin, A. H. Zaim and K. G. Ceylan, "A hybrid intrusion detection system design for computer network security," *Computers & Electrical Engineering,* vol. 35, pp. 517-526, May 2009.

[5] C. Amza, C. Leordeanu and V. Cristea, "Hybrid Network Intrusion Detection," in *IEEE International Conference on Intelligent Computer Communication and*, 2011.

[6] D. J. Brown, B. Suckow and T. Wang, *A Survey of Intrusion Detection Systems*, Department of Computer Science, University of California, San Diego, 2002.

[7] J. Beale, A. Baker, J. Esler and S. Northcutt, Snort: IDS and IPS toolkit: Syngress Media Inc, 2007.

[8] KDDCup99 Dataset, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html 1999

[9] DARPA intrusion detection evaluation, https://www.ll.mit.edu/ideval/data/1998data.html

[10] C. Thomas V. Sharma N. Balakrishnan, "Usefulness of DARPA Dataset for Intrusion Detection System Evaluation" Proceedings of SPIE, Vol. 6973, 2008.

[11] R. Chitrakar and H. Chuanhe, "Anomaly based Intrusion Detection using Hybrid Learning Approach of combining k-Medoids Clustering and Naïve Bayes Classification", IEEE,2012.

[12] J. Han and M. Kamber, Data mining concepts and techniques, 2nd ed., Morgan Kaufmann Publishers, 2006.