International Journal of Current Trends in Engineering & Technology
ISSN: 2395-3152
Volume: 02, Issue: 02 (MAR-APR, 2016)

Enhanced Method for Intrusion Detection over KDD Cup 99 Dataset

Vivek Nandan Tiwari, Prof. Satyendra Rathore, Prof. Kailash Patidar
Computer Science & engineering Department, SSSIST, Sehore, INDIA

Abstract: **-** Intrusion detection is especially vital features of protecting the internet infrastructure from assaults or hackers. Intrusion prevention method for instance firewall, filtering router policies fails to prevent such type of assaults. An intrusion detection scheme is utilized to identify several types of malevolent activities that could negotiate the safety and faith of a computer organization. This comprises network assaults against susceptible services, information driven assaults on applications, host based assaults such as advantage escalation, unauthorized logins and entrance to sensitive files and malware. The KDD cup 99 dataset is a well-recognized standard in the research of Intrusion Detection Techniques. Various efforts is going on for the enhancement of intrusion detection strategies while the research on the data utilized for training and testing the detection model is uniformly of prime concern since improved data superiority could advance offline intrusion detection. In this work the investigation is carried out with respect to two important evaluation metrics, True Positive (TP)/Recall and Precision/Accuracy for an Intrusion Detection System (IDS) in KDD cup 99 dataset. As a outcome of this experiential investigation on the KDD cup 99 dataset, the contribution of every of four assault classes of attributes on Recall and Precision is illustrate which can assist to improve the correctness of KDD cup 99 dataset which attain highest accuracy with lowest false positive (FP).

Keywords: **- Intrusion Detection, WEKA, Classifiers, Precision, Recall, Machine Learning.**

I. INTRODUCTION

In an ideal environment, systems could be developed using *provable security*. The term was coined about cryptology originally, but it applies to other aspects of secure system design [1]. Provable security uses formally defined security requirements based on assumptions about the adversary. With the complexity of modern information systems, it is difficult to define the security requirement adequately. Any proof based on the requirements relies on the accuracy and completeness of the requirements. When the security requirements underestimate the adversary's capabilities, the proof is not completely valid [2]. Even if a system is too complex for provable security, processes should be developed and followed to minimize the risk of vulnerability in software applications [3]. The processes for secure software development include similar concepts as provable security. Developers identify the potential adversary and the risk is analyzed based on the value of the data and the estimated capabilities of the adversary. Use cases are developed to help developers create and verify security. Even with security requirements, use cases, code walkthroughs, and vulnerability testing, unknown vulnerabilities still make it into systems. Controls such as intrusion detection systems, firewalls, and local access controls are used to improve the security posture of a system [4]. Firewalls are common for perimeter security, but they have limitations. Properly implemented access controls are essential for internal security; however, they are unable to protect a system that contains vulnerable software fully. Vulnerabilities in software are not the only factors making a system insecure. Frequently, the security of a system depends on its configuration. In many cases, when a system's configuration is incorrect, malicious users can easily pass through controls such as a firewall to gain access to a system. Intrusion detection adds another layer of security on top of access controls. Intrusion detection can be extremely powerful in detecting novel exploits, but it can require an excessive amount of resources [5]. For known exploits, intrusion detection systems can quickly classify and shun attacks. Systems that only have the resources to use intrusion detection systems that rely on pre-existing knowledge of individual exploits are vulnerable to novel exploits until security experts can manually create classifiers for those exploits. Automated signature generation (ASG) is used to fill the gap until security experts can analyze novel exploits [6]. With ASG, classifiers are dynamically created by hosts that can handle the required heavy resource burden and then the system distributes them to hosts that only have enough resources to compare data to existing classifiers. Data mining based IDS can efficiently identify these data of user interest and also predicts the results that can be utilized in the future. Data mining or knowledge discovery in databases has gained a great deal of attention in IT industry as well as in the society. Data mining has been involved to analyze the useful information from large volumes of data that are noisy, fuzzy and dynamic. Fig. 1 illustrates the overall architecture of IDS. It has been placed centrally to capture all the incoming packets that are transmitted over the network. Data are collected and send for pre-processing to remove the noise; irrelevant and missing attributes are replaced. Then the preprocessed data are analyzed and classified according to their severity measures. If the record is normal, then it does not require any more change or else it send for report generation to raise alarms. Based on the state of the data, alarms are raised to make the administrator to handle the situation in advance. The attack is modeled so as to enable the classification of network data. All the above process continues as soon as the transmission starts. A problem with existing security controls is the limited capabilities in detecting novel structured access exploits. For complex systems, it is infeasible to implement a design with provable security. Controls are enacted to mitigate the security risk to systems. One of the controls is intrusion detection. Signature-based intrusion detection looks for known malicious patterns. It can process events much faster and with fewer resources than anomaly detection. A limitation

of signature-based intrusion detection is that it requires existing knowledge of an exploit. Automated signature generation refers to the creation of signatures based on some learning technique. Existing automated signature generation systems all have limitations.

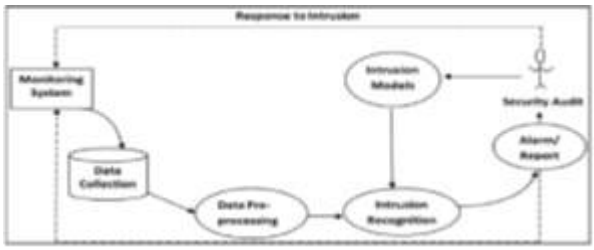

Fig. 1: Overall structure of Intrusion Detection System

The most common limitations are that many existing systems require labeled training data and conduct all training on a static dataset. It is also common for systems to use expert knowledge for feature extraction. There are some approaches that can train over time with unlabeled data. These systems work well with worms, but are not able to detect more covert exploits.

II. OVERVIEW OF IDS

In day to day life the need for speed access of information through internet has increased. Hence the room for maintaining security in any organization either public or private system has become fundamental. Because of increase in network connections and systems, unauthorized access and interruption of the data is triggered. As a result, it is indispensable to create a virtual access path. In general intruders have capacity to find out defect in systems or networks and can spawn vulnerabilities. Even though the access control points exist in network, they fail in providing scrupulous security to the systems. To identify intruders, developing Intrusion Detection Systems (IDSs) is the best solution to protect systems and networks. Therefore the task of IDS is not only to detect intruders but also to monitor the raid of intruders. An accurate system of protecting data and resources from illicit access, damaging and denial of use is to be built. For every system, the security perspective is to be planned based on the expected performance. Mainly security is concerned with the following aspects in a computer system.

- Confidentiality – information is to be accessed only by permissible persons.
- Integrity – information must remain unaffected by destructive or malicious attempts.
- Availability – computer is responsible to function without downgrading of access and provide resources to legal users when they require it.

Specifically an intrusion is defined as a set of events which are unknown and unforeseen to the user, which compromises the protection of a computer system. It can be done from external side or internal side of the system. Earlier in 1980"s James P Anderson has defined intrusion as the scope of illegal force to access information, defraud information, or making the computer system unsafe. Intrusion Detection System (IDS) was commercially promoted in the year 1990. From then a variety of layouts were introduced to adapt intrusion detection systems [7] [8]. It acts like a burglar alarm and detects any kind of violation and generates alarms like audible, visual and also messages like e-mail. On the whole, IDS is primarily exploited for stopping defective activities that may attack or misuse the system by identifying attacks through providing desirable support for defense management and also give constructive information regarding intrusion. But structure of IDS should possess low fake alarms while undertaking the discovery of attacks. IDSs have become shielding mechanisms everywhere in current networks. There is no thorough and proficient methodology offered in checking the strength of these systems. Because Intrusion Detection Systems performance is increased with usage of the Soft Computing methods to IDS, the computer security researchers are trying to apply. Soft computing is the collection of approaches that were set up to model and obtain guaranty solutions to real world problems, which are not modeled or very difficult to model mathematically. Soft computing is a general term for developing the enduring for imprecision, partial truth, uncertainty, and estimate of achieving flexibility and minor solution cost. A masterful and accurate tool for real time intrusion discovery is the target of main experimenters in IDSs. There is a variety of Artificial Intelligence (AI) concepts were exploited for transforming intrusion discovery procedure, therefore human involvement is decreased. And also in common, the procedures which deal with IDS are utilizing machine learning. Basically Soft Computing techniques that were used in IDS implementation are Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), Bayesian Networks, Fuzzy logic, Particle Swarm Optimization and Genetic Algorithms (GA).

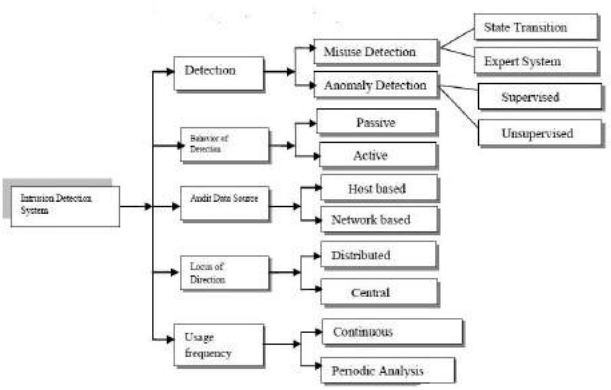

Fig. 2: Classification of IDS based on its characteristics

Discovering intrusions merely with human eyes will be tremendously intricate. Towards diminishing the crisis, system security scientists use prevailing data mining and artificial intelligence methodologies in exploring probable intrusions. Conversely, if the total set of features employed in network data is increased then classifying intrusions become complicated, since complex relationships exists between features [9]. There are complex relationships existing among features as well as intrusion classes. It will produce more processing costs and also delays in detecting intrusions. In view of the restrictions on humans and computers together, feature selection is accordingly essential such that burden in handling data and time required in noticing intrusions will be lessened [10]. In detecting intrusions, IDS defends a computer network from illicit persons, possibly insiders. The attack recognition

task is considered as the model of classification expert in distinguishing "harmful" connections referred as intrusions or attacks, and "sympathetic" connections referred as normal. There are various categories of IDSs are prevailing that are based on structure and detection method. In addition to these, there are other characteristics one can used to classify IDS as shown in the fig.2.

III. CLASSIFICATION OF IDS

IDSs are mainly classified based on the source of data used for intrusion detection into two types as Host-based IDS (HIDS) and Network based IDS (NIDS). A HIDS [11] monitors system logs for evidence of malicious or suspicious application activity in real time and also monitors key system files for evidence of tampering. A NIDS [12] monitors live network packets and looks for signs of computer crime, network attacks, network misuse and anomalies. Both host and network based IDSs generate alarms whenever they detect any suspicious activity in the network. These alarms are used by the network administrator or some automated response tool to trigger a response in order to safeguard the network from such attackers [13]. IDSs are categorized based on the detection techniques into Signature-based IDS and Anomaly-based IDS [14]. The Signature-based or Misuse IDS [15] monitors packets on the network and compares them against a database of signatures or attributes from known malicious threats. This is similar to the way most antivirus software detects malware. The issue is that there will be a lag between a new threat being discovered in the wild and the signature for detecting that threat being applied to IDS. During that lag time, IDS would be unable to detect the new threat. An Anomaly based IDS [16] monitor network traffic and compare it against an established baseline. This baseline is used to distinguish the normal and abnormal activities happening in the network by analyzing the bandwidth and protocols used. It also analyzes the ports and devices used to connect with each other and then alerts the administrator when traffic is detected which is significantly different than the baseline. Both HIDS and NIDS employ these techniques to detect intrusions. Since, each technique has advantages and disadvantages, a hybrid system which incorporates two or more of these techniques can provide a good level of security. The security technique has to be chosen depending upon the environment in which it is used, rather than the performance of individual techniques. Another classification of IDS is based on the behavior as Passive IDS and Reactive IDS. The passive IDS simply detect and alerts. When suspicious or malicious traffic is detected an alert is generated and sent to the administrator or user and it is up to them to take action to block the activity or respond in some way. The reactive IDS will not only detect suspicious or malicious traffic and alert the administrator, but will take pre-defined proactive actions to respond to the threat. Typically this means blocking any further network traffic from the source Internet Protocol (IP) address or user.

IV. DATA MINING APPROACHES FOR IDS

Most IDSs are based on hand-crafted signatures that are developed by manual encoding of expert knowledge. These systems match activity on the system being monitored to known signatures of attacks. The major problem with this approach is that these IDSs fail to generalize to detect new attacks or attacks without known signatures. Recently, there has been an increased interest in Data Mining based approaches have been proposed to building detection models for IDSs [17]. These models generalize from both known attacks and normal behavior in order to detect unknown attacks. They can also be generated in a quicker and more automated [18] than manually encoded models that require difficult analysis of audit data by domain experts. Besides, there are many other works that use intelligent data mining techniques for intrusion detection such as Artificial Neural Network (ANN) [19], Genetic Algorithm (GA) [20], Support Vector Machine (SVM) [21], Fuzzy Logic (FL), Decision Tree (DT), and Genetic Programming (GP) for the discovery of useful knowledge in order to detect intrusive patterns. Data Mining is the process of extracting valid, authentic, and actionable information from large databases. Data Mining is an analytic process designed to explore data in search of consistent patterns and/or systematic relationships between variables, and then to validate the findings by applying the detected patterns to new subsets of data. The ultimate goal of data mining is prediction - and predictive data mining is the most common type of data mining and one that has the most direct business applications. Data mining strategies fall into two broad categories [22] namely Supervised Learning and Unsupervised Learning. Supervised Learning methods are deployed when there exists a field or variable (target) with known values and about which predictions will be made by using the values of other fields or variables (inputs). Unsupervised Learning methods tend to be deployed on data for which there do not exist a field or variable with known values, while fields or variables do exist for other fields or variables. Several effective data mining techniques have been developed for detecting intrusions [23, 24, 25] which perform close to or better than systems engineered by domain experts. However, successful data mining techniques are themselves not enough to create deployable IDSs. Despite the promise of better detection performance and generalization ability of data mining-based IDSs, there are some inherent difficulties in the implementation and deployment of these systems. These difficulties can be grouped into three general categories: accuracy (i.e., detection performance), efficiency, and usability. Typically, data mining-based IDSs (especially anomaly detection systems) have higher false positive rates than traditional hand-crafted signature based (misuse detection systems) methods, making them unusable in real environments. Also, these systems tend to be inefficient (i.e., computationally expensive) during both training and evaluation. This prevents them from being able to process audit data and detect intrusions in real time. Finally, these systems require large amounts of training data and are significantly more complex than traditional systems. In order to be able to deploy real time data mining based IDSs, these issues must be addressed. Large amount of data exists in the system which could be gathered by network personnel to detect security policy violations. With this scenario, the analysis is a tedious one and network administrators do not have the resources to analyze the data for security policy violations especially in the presence of a high number of false positives that cause them to waste their limited resources. One of the

challenges of intrusion detection systems is to analyze data so that a legitimate or intrusive activity could be detected [26]. The solution is to employ data mining techniques [27] in an offline environment. This kind of approach would add additional depth to the network administrator's defenses, and allows them to more accurately determine what the threats against their network are through the use of multiple methods on data. Data mining techniques are used in classification and identification [28] of new patterns from large volume of training data that are collected from KDD (Knowledge Discovery in Data Mining) CUP 1999 benchmark dataset in order to perform hybrid intrusion detection in host as well as in network. Moreover, intrusion detection has been carried out using classification and clustering algorithms integrated with feature selection [29].

V. KDD CUP 99 DATASET

The KDD (Knowledge Discovery in Data mining) CUP 1999 Dataset is used to validate the effectiveness of the proposed Hybrid IDS. The KDD CUP 1999 intrusion detection dataset is based on the 1998 DARPA initiative, which provides designers of intrusion detection systems with a bench mark on which to evaluate different methodologies. To do so, a simulation is made of a factitious military network consisting of three 'target' machines running various operating systems and services. Additional three machines are then used to spoof different IP addresses to generate traffic. Finally, there is a sniffer that records all network traffic using the TCP dump format. The total simulated period is seven weeks. Normal connections are created to profile that expected in a military network and attacks fall into one of four categories as follows:

- **Denial of Service (DoS)**: Attacker tries to prevent legitimate users from using a service.
- **Remote to Local (R2L)**: Attacker does not have an account on the victim machine, hence tries to gain access.
- **User to Root (U2R)**: Attacker has local access to the victim machine and tries to gain super user privileges.
- **Probe**: Attacker tries to gain information about the target host.

In 1999, the original TCP dump files were preprocessed for utilization in the Intrusion Detection System benchmark of the International Knowledge Discovery and Data Mining Tools Competition. To do so, packet information in the TCP dump file is summarized into connections. Specifically, a connection is a sequence of TCP packets starting and ending at some well-defined times, between which data flows from a source IP address to a target IP address under some well-defined protocol. This process is completed using the Bro IDS, resulting in 41 features for each connection. Features are grouped into four categories as given below:

- **Basic Features**: Basic features can be derived from packet headers without inspecting the payload.
- **Content Features**: Domain knowledge is used to assess the payload of the original TCP packets. This includes features such as the number of failed login attempts.
- **Time-based Traffic Features**: These features are designed to capture properties that mature over a 2 second temporal window. One example of such a feature would be the number of connections to the same host over the 2 second interval.
- **Host-based Traffic Features**: Utilize a historical window estimated over the 100 number of connections instead of time. Host-based features are therefore designed to assess attacks, which span intervals longer than 2 seconds.

The KDD CUP 1999 intrusion detection benchmark dataset consists of three components, which are detailed in Table 1.1. In the International Knowledge Discovery and Data Mining Tools Competition, only "10% KDD" dataset is employed for the purpose of training. This dataset contains 22 attack types and is a more concise version of the "Whole KDD" dataset. It contains more examples of attacks than normal connections and the attack types are not represented equally. Because of their nature, denial of service attacks account for the majority of the dataset. On the other hand the "Corrected KDD" dataset provides a dataset with different statistical distributions than either "10% KDD" or "Whole KDD" and contains 14 additional attacks. Since "10% KDD" is employed as the training set in the original competition, the analysis of proposed hybrid IDS was performed on the "10% KDD" dataset. To carry the experiments effectively, KDD CUP 1999 dataset containing connection records with varying distribution of attack types and normal class has been used in the proposed hybrid IDS. Also, the proportion of data in the testing dataset is not same as that of training dataset and also the test data include some specific type of attacks which are not in the training set. This makes the real- time intrusion detection more practical.

VI. PROPOSED WORK

In this paper an "Enhanced Intrusion Detection Method Using Machine Learning for KDD Cup 99 Dataset" is proposed to enhance the efficiency of intrusion detection using KDD cup Intrusion dataset. We utilized Naïve Bayes, J48 and Random forest classifiers for the classification. Classifiers are evaluated based on Precision, recall, f-measures and ROC Curve Area performance criteria's. A WEKA 3.7.1 workbench is used for experimental study. It is observed that random forest is the best classifier among all used methods. Following algorithm is used to implement proposed method on Linux OS (Ubuntu 14.04)

Input: KDD Cup99 Dataset

Output: classified dataset in ARFF format

Step 1: Create Temp file for processing

Step 2: Pre-process input dataset

Step 3: Remove outliers // trim the dataset

Step 4: Replace all attacks by their parent category

Step 5: create WEKA compatible file of classified attacks // .ARFF file

Step 6: check the accuracy of classification of proposed method on WEKA by applying different classifiers (for

instance we used Naive Bayes, J48 and Random forest classifiers)

Table 1: Characteristics of the KDD CUP 99 Intrusion Detection Dataset

Dataset	DoS	Probe	U2R	R2L	Normal
10% KDD	391458	4107	52	1126	97277
Corrected KDD	229853	4166	70	16347	60593
Whole KDD	3883370	41102	52	1126	972780

Used WEKA Classifiers: - Processed dataset is applied to the Naive Bayes, J48, and Random forest classifiers. Brief description of each classifier is given here.

1. **Naive Bayes classifier: -** A naive Bayes classifier is a simple probabilistic classifier based on applying Baye's theorem with strong (naive) independence assumptions. In simple terms, a naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature, given the class variable. Naïve Bayes classifier assumes that the effect of the value of a predictor (X) on a given class (C) is independent of the values of other predictors. This assumption is called class conditional independence.
2. **J48 Classifier: -** A decision tree is a predictive machine-learning model that decides the target value (dependent variable) of a new sample based on various attribute values of the available data. The internal nodes of a decision tree denote the different attributes; the branches between the nodes tell us the possible values that these attributes can have in the observed samples, while the terminal nodes tell us the final value (classification) of the dependent variable. The attribute that is to be predicted is known as the dependent variable, since its value depends upon, or is decided by, the values of all the other attributes. The other attributes, which help in predicting the value of the dependent variable, are known as the independent variables in the dataset. The J48 Decision tree classifier follows the following simple algorithm. In order to classify a new item, it first needs to create a decision tree based on the attribute values of the available training data. So, whenever it encounters a set of items (training set) it identifies the attribute that discriminates the various instances most clearly. This feature that is able to tell us most about the data instances so that we can classify them the best is said to have the highest information gain. Now, among the possible values of this feature, if there is any value for which there is no ambiguity, that is, for which the data instances falling within its category have the same value for the target variable, then we terminate that branch and assign to it the target value that we have obtained.
3. **Random forests Classifier**:-Random forests are an ensemble learning technique for classification, regression and extra tasks, that operate by constructing a large number of decision trees at training time and outputting the category that's the mode of the categories (classification) or mean prediction (regression) of the individual trees. Random forests correct for decision trees' habit of over fitting to their training set.

VI. RESULT ANALYSIS

Experiment is carried out on the system having Intel Core i3 Processors, 4 GB RAM, 500 GB HDD, UBUNTU 14.10 Operating System and WEKA Machine V 3.6.11 Learning Workbench developed by university of Waikato is utilized for the classification task. WEKA [29], for Waikato Environment for Knowledge Analysis, is a collection of various Machine Learning algorithms, implemented in Java that can be used for data mining problems. Besides applying ML algorithms on datasets and discuss about the results generated, WEKA also gives options for pre-processing classification, regression, clustering, association rules and visualization of the dataset. It can be extended by the user to execute new algorithms. Classification Models are evaluation based on following criteria's.

1. **True Positive (TP) / Recall:** Recall in this context is defined as the number of true positives divided by the total number of elements that actually belong to the positive class (i.e. the sum of true positives and false negatives, which are items which were not labeled as belonging to the positive class but should have been).

It is the proportion of positive cases that were correctly classified as positive, as calculated using the equation:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

2. **False Positive (FP):** It is the proportion of negative cases that were incorrectly classified as positive, as calculated using the equation:

$$\text{FP} = \frac{\text{FP}}{\text{TN} + \text{FP}}$$

3. **True Negative (TN):** It defined as the proportion of negatives cases that were classified correctly, as calculated using the equation:

$$\text{TN} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

4. **False Negative (FN):** It is the proportion of positive cases that were incorrectly classified as negative, as calculated using the equation:

$$\text{FN} = \frac{\text{FN}}{\text{FN} + \text{TP}}$$

5. **Precision: -** Precision for a class is the number of true positives (i.e. the number of items correctly labeled as belonging to the positive class) divided by the total number of elements labeled as belonging to the positive class (i.e. the sum of true positives and false positives, which are items incorrectly labeled as belonging to the class). Accuracy (i.e., Precision) is the proportion of the total number of attacks that are correctly detected. It is determined using the equation:

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

Here, TP is True Positive; FP is False Positive, TN is True Negative, FN is False Negative.

6. **F- Measure: -** A measure that combines precision and recall is the harmonic mean of precision and recall, the traditional F-measure or balanced F-score.

F- Measure that mixes precision and recall is the harmonic mean of precision and recall is known as F-measure.

$$\text{F} - \text{measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

This is also known as the F1 measure, because recall and

precision are evenly weighted. [34]

7. **ROC: -** Receiver operating characteristics (ROC) graphs are useful for organizing classifiers and visualizing their performance. Receiver Operating Characteristic (ROC), or ROC curve, is a graphical plot that illustrates the performance of a binary classifier system as its discrimination threshold is varied. The curve is created by plotting the true positive rate against the false positive rate at various threshold settings. Receiver Operator Characteristics (ROC) illustrates the trade-off between sensitivity and specificity. ROC curves plot the true positive rate vs. the false positive rate, at varying threshold cut-offs. The ROC is also known as a relative operating characteristic curve, because it is a comparison of two operating characteristics (TPR and FPR) as the criterion changes.

The Results of naïve bayes, J48 and random forest Classifier is shown in Table 2, 3 & 4. As can be seen the performance of Naive Bayes Classifier is below par. For U2R and R2L attack is it's below 41% mark. The reason for this is due to the assumption of Naive Bayes approach that all parameters are independent. However this is not always the case. Many security parameters are inter-dependent to one another. As a result Naive Bayes Classifier, though takes less memory and is faster in computation is avoided because of poor results To improve upon Naive Bayes Classifier we have used J48 and Random Forrest classifier in WEKA. These two classifiers have shown significant improvements in detection rate and accuracy. As can be observed in Figure 1 that average TP rate for J48 and Random Forrest classifier is above 98% which is quite higher as compared to naive Bayes whose weighted average is 78.1%. Almost all the attacks have precision of exceeding 81% in J48 and Random Forrest classifier except for R2L attack.

Table 2: Results of Naive Bayes Classifier

	TP-Rate	FP-Rate	Precision	Recall	F-Measure	MCC	ROC-Area	PRC-Area	Class
	0.793	0.010	0.995	0.793	0.883	0.699	0.987	0.994	dos
	0.732	0.003	0.130	0.732	0.220	0.307	0.997	0.141	u2r
	0.984	0.139	0.087	0.984	0.161	0.272	0.994	0.794	probe
	0.966	0.076	0.411	0.966	0.576	0.603	0.976	0.718	r2l
	0.674	0.005	0.970	0.674	0.796	0.775	0.977	0.925	normal
Weighted-Avg.	0.781	0.014	0.947	0.781	0.840	0.703	0.985	0.963	

Table 3: Results of J48 Classifier

	TP-Rate	FP-Rate	Precision	Recall	F-Measure	MCC	ROC-Area	PRC-Area	Class
	1	0.001	1	1	1	0.999	0.999	1	dos
	0.761	0	0.857	0.761	0.806	0.807	0.934	0.773	u2r
	0.978	0	0.985	0.978	0.981	0.981	0.994	0.976	probe
	0.83	0.011	0.81	0.83	0.82	0.81	0.994	0.913	r2l
	0.947	0.011	0.953	0.947	0.95	0.938	0.997	0.991	normal
Weighted-Avg.	0.98	0.003	0.98	0.98	0.98	0.977	0.998	0.993	

We have compared our contribution with the work with [30] and it's given in table 5. In [30] the authors have used C4.5 and SVM for classification. We have used Naive Bayes, J48 and Random Forrest for classification. The table IX shows the precision under various classifiers used. Though the effectiveness of detection of probing attack have been reduced, the improvement in DoS, U2R, R2L have been significant. The use of naive bayes results in poor results since naive bayes assumes all parameters to be independent

Table 4: Results of Random Forrest Classifier

	TP-Rate	FP-Rate	Precision	Recall	F-Measure	MCC	ROC-Area	PRC-Area	Class
	1.000	0.001	1.000	1.000	1.000	0.999	1.000	1.000	dos
	0.915	0.000	0.956	0.915	0.935	0.935	0.986	0.951	u2r
	0.989	0.000	0.996	0.989	0.993	0.992	0.999	0.997	probe
	0.820	0.010	0.816	0.820	0.818	0.808	0.995	0.921	r2l
	0.950	0.012	0.952	0.950	0.951	0.939	0.999	0.993	normal
Weighted-Avg.	0.981	0.003	0.981	0.981	0.981	0.978	0.999	0.995	

Table 5: Comparison of Proposed Work with Previous Methods

	C4.5	SVM	Naive Bayes	J48	Random Forrest
DoS	93.87	93.84	99.5	100	100
Probe	95.38	89.09	13	85.7	95.6
U2R	33.33	66.67	8	98.5	99.6
R2L	16.44	15.9	41	81	81.6

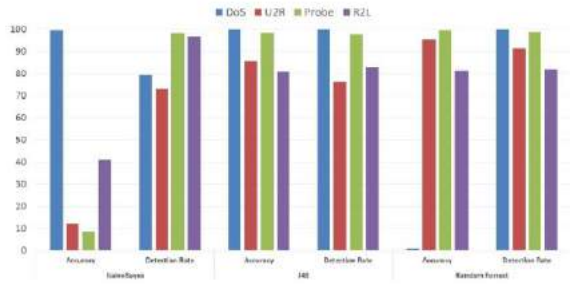

Fig 3: Accuracy and Detection Rate of Classifiers Utilized

VII. CONCLUSION

Intrusion Detection Systems provide the fundamental detection techniques to secure the systems present in the networks that are directly or indirectly connected to the Internet and effectively analysis the problems available in the existing intrusion detection techniques. In this paper we are providing solution on the existing intrusion detection techniques through speedup and accurate anomaly network intrusion detection system. In this work, the proposed method of machine learning for intrusion detection system is presented the proposed method is evaluated on KDD Cup 99 dataset and training of 66% is done. The performances of WEKA classifies are measured in terms of True Positive (TP)/Recall and Precision/Accuracy and false positives. The performance of the all method is compared with other standard machine learning techniques. The experimental results show that the proposed machine learning technique provides highest classification accuracy of 99.67 %

REFFERENCES

[1] Degabriele, Jean Paul, Kenneth G. Paterson, and Gaven J. Watson. "Provable security in the real world." IEEE Security & Privacy 3 (2010): 33-41.

[2] Koblitz, Neal, and Alfred J. Menezes. "Another look at" provable security". "Journal of Cryptology 20, no. 1 (2007): 3-37.

[3] Mead, Nancy R., Julia H. Allen, Sean Barnum, Robert J. Ellison, and Gary McGraw. Software Security Engineering: A Guide for Project Managers. Addison-Wesley Professional, 2004.

[4] Stewart, J. Michael. Network Security, Firewalls and VPNs. Jones & Bartlett Publishers, 2013.

[5] Caswell, Brian, Jay Beale, and Andrew Baker. Snort Intrusion Detection and Prevention Toolkit. Syngress, 2007.
[6] Wang, Lanjia, Zhichun Li, Yan Chen, Zhi Fu, and Xing Li. "Thwarting zero-day polymorphic worms with network-level length-based signature generation." IEEE/ACM Transactions on Networking (TON) 18, no. 1 (2010): 53-66.
[7] Teresa F. Lunt., "A survey of intrusion detection techniques", Computers and Security, Elsevier Advanced Technology Publications, 12(4):405–418, 1993.
[8] Emilie Lundin, Erland Jonsson" Survey of Intrusion Detection Research" ,Technical Report 02-04, Department of Computer Engineering, Chalmers University of Technology, 2002
[9] Kok-Chin K., Choo-Yee T., Somnuk-Phon A," A Feature Selection Approach for Network Intrusion Detection", International Conference on Information Management and Engineering, IEEE computer Society,K.L., PP.133-137, . 2009.
[10] Shichao Z., Li L., Xiaofeng Z. and Chen Z.,"A Strategy for Attributes Selection in Cost-Sensitive Decision Trees Induction", IEEE 8th International Conference on Computer and Information Technology Workshops, PP. 1-13,2008
[11] Ye, Nong, Syed Masum Emran, Qiang Chen, and Sean Vilbert. "Multivariate statistical analysis of audit trails for host-based intrusion detection. "Computers, IEEE Transactions on 51, no. 7 (2002): 810-820.
[12] Mukherjee, B., Heberlein, L.T. and Levitt, K.N. "Network Intrusion Detection", IEEE Networks, Vol. 8, No. 3, pp. 26-41, 1994.
[13] Chirillo, J. "Network Security for Windows, UNIX and Linux Networks: Hack Attacks Denied", Wiley Publishing Inc., 2nd Edition, 2002.
[14] Lee, W., and Stolfo, S.J. "Data Mining Approaches for Intrusion Detection", in Communications of the ACM, Vol. 39, No.11,pp. 27-34, 1996.
[15] Axelsson, S. "Intrusion Detection Systems: A Taxonomy and Survey", Technical Report No 99-15, Department of Computer Engineering, Chalmers University of Technology, Sweden, 2000.
[16] Lazarevic, A., Ertoz, L., Kumar, V., Ozgur, A. and Srivastava, J. "A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection", in Proceedings of the 3rd SIAM International Conference on Data Mining, 2003.
[17] Peng, T. and Zuo, W. "Data Mining for Network Intrusion Detection System in Real Time", International Journal of Computer Science And Network Security, Vol. 6, No. 2b, 2006.
[18] Sifalakis, M., Fry, M. and Hutchison, D. "Event Detection and Correlation for Network Environments", IEEE Journal on Selected Areas in Communications, Vol. 28, No. 1, pp. 60-69, 2010.
[19] Peddabachigari, S., Abraham, A., Grosan, C. and Thomasa, J. "Modeling Intrusion Detection System using Hybrid Intelligent Systems", Journal of Network and Computer Applications, Special Issue: Network And Information Security: A Computational Intelligence Approach, Vol. 30, No. 1, pp. 114-132, 2007.
[20] Ling, S.H., Leung, F.H.F., Lam, H.K., Lee, Y.S. and Tam, P.K.S. "A novel GA-Based Neural Network for Short-Term Load Forecasting", IEEE Transactions on Industrial Electronics, Vol. 50,No. 4, pp. 793-799, 2003
[21] Chen, Y. and Wang, J.Z. "Support Vector Learning for Fuzzy Rule-Based Classification Systems", in IEEE Transactions on Fuzzy Systems, Vol. 11, No. 6, pp. 716-728, 2003.
[22] Dunham, M.H. "Data Mining – Introductory and Advanced Topics", Pearson Education, 2006.
[23] Eskin, E. "Anomaly Detection over Noisy Data using Learned Probability Distributions", in Proceedings of 17th International Conference on Machine Learning, pp. 255-262, 2000.
[24] Prema Rajeswari, L. and Kannan, A. "An Active Rule Approach for Network Intrusion Detection with Enhanced C4.5 Algorithm", International Journal of Communications on Network and System Sciences, pp. 285-385, 2008.
[25] SivathaSindhu, S.S., Geetha, S. and Kannan, A. " Decision Tree based Light Weight Intrusion Detection using a Wrapper Approach", in Journal of Expert Systems with Applications, Vol. 39, pp. 129-141, 2012.
[26] Mukkamala, S., Janoski, G. and Sung, A. "Intrusion detection using Neural Networks and Support Vector Machines", in Proceedings of IEEE International Joint Conference on Neural Networks, pp. 78-83, 2002.
[27] Fayyad, U.M. and Uthurusamy, R. "Evolving Data Mining into Solutions for Insights", in ACM Communication, Vol. 45, pp. 28-31, 2002.
[28] Lu, C.T., Boedihardjo, A.P. and Manalwar, P. "Exploiting Efficient Data Mining Techniques to Enhance Intrusion Detection Systems" in Proceedings of the 2005 IEEE International Conference on Information Reuse and Integration (IEEE IRI-2005 Knowledge Acquisition and Management), pp. 512-517, 2005.
[29] Luo, L., Ye, L., Luo, M., Huang, D., Peng, H. and Yang, F. "Methods of Forward Feature Selection Based on the Aggregation of Classifiers Generated by Single Attribute", Computers in Biology and Medicine, Vol. 41, No. 7, pp. 435-441, 2011.
[30] Ektefa, Mohammadreza, Sara Memar, Fatimah Sidi, and Lilly Suriani Affendey. "Intrusion detection using data mining techniques." In Information Retrieval & Knowledge Management, (CAMP), 2010 International Conference on, pp. 200-203. IEEE, 2010.