AN IMPROVED METHOD TO DETECT INTRUSION USING MACHINE LEARNING ALGORITHMS

Urvashi Modi[1] and Anurag Jain[2]
[1, 2] CSE departments, Radharaman inst. of Tech & Science, Bhopal, India

ABSTRACT

An intrusion detection system detects various malicious behaviors and abnormal activities that might harm security and trust of computer system. IDS operate either on host or network level via utilizing anomaly detection or misuse detection. Main problem is to correctly detect intruder attack against computer network. The key point of successful detection of intrusion is choice of proper features. To resolve the problems of IDS scheme this research work propose "an improved method to detect intrusion using machine learning algorithms". In our paper we use KDDCUP 99 dataset to analyze efficiency of intrusion detection with different machine learning algorithms like Bayes, NaiveBayes, J48, J48Graft and Random forest. To identify network based IDS with KDDCUP 99 dataset, experimental results shows that the three algorithms J48, J48Graft and Random forest gives much better results than other machine learning algorithms. We use WEKA to check the accuracy of classified dataset via our proposed method. We have considered all the parameter for computation of result i.e. precision, recall, F – measure and ROC.

KEY WORDS:

IDS, KDDCUP 99, Machine learning, WEKA, Network Security, Precision, Recall

1. INTRODUCTION

Intrusion-Detection System (IDS) has been observed as the"silver bullet" that guarantees safety of an organization system against possible attacks. Although after the extension of this method, it's not successfully utilized due to the huge quantity of fake alarms that it creates. For example, the well identified open source Intrusion Detection System Snort technique [1] is performs on a network with few hundred machines and it generates thousands of alerts daily, which holds a bulk of fake alarms. IDS operation frequently generates huge quantity of results that usually flow into the organization's Safety Operation Center (SOC), therefore causing an idealistically great quantity of effort and lengthy working for protection analysts.

To overcome with this difficulty, researchers usually generate precise IDS rules (signatures) that specifically detain very precise attacks and decrease the overall False Positive (FP) rate. Though, this results in disappointment to separate other attacks or other forms of the targeted attack due to the polymorphic character of the attacks, which is an effect of the human intelligence that locates behind them. Additionally, to stop False Negatives (FN), i.e., detection misses the IDS' system researchers resort to merge the above methods with a more generic system, so that an action with even a remote opportunity of representing an attack will activate an alert.

1.1 Intrusion Detection Systems

Cyber attacks on PCs, Organizations, and governments have become every day events which break, privacy, reliability, and accessibility of the concerned computer systems. Thus, a system must be in put that could identify and prevent these attacks on a computer host or network. Therefore, various schemes and systems have appeared to automate this process. Different terms

[2] of IDS are as follows:

- **Intrusion:** effort to compromise Confidentiality, Integrity, and/or Availability (CIA) in a computer system or network
- **Intrusion Detection:** procedure of observing events happening in a computer organization or network and investigating them for sign terms of intrusions
- **Intrusion Detection System (IDS):** part of a software or hardware scheme that automates the intrusion detection procedure.
- **Intrusion Prevention System (IPS):** scheme holding all IDS capabilities but could also actively prevents possible events.

IDS can be classified into network based and host based and classified with method into signature, anomaly, and specification based. These classifications are detailed in the following:

- **Host Based IDS (HIDS):** scheme that exist in as an agent or host on the local computer and observes machine behavior, for example by examining the logs
- **Network Based IDS (NIDS):** scheme that observes network traffic usually consisting of sensors distributed over the network and a processing unit. The sensors sniff network packets, for example TCP/IP packets, and the scheme efforts to recognize malicious packets or abnormal action on the network [3].

1.2. Detection Approaches

Intrusion detection methods generally classified in following categories:

- **Misuse based or Signature based IDS:** a signature is a prototype that corresponds to a recognized attack or threat, misuse detection is the method to compare prototypes against captured occurrences to identify probable intrusions. It efforts in the similar way as most of the antivirus program do [4]. It checks the network for behavior that has been predetermined to be malicious. They are perfect and quick since they are only doing a comparison among what they are observing and a predetermined rule. The latest threats will not be identified by Signature based IDS. When a new attack is traced, the data files require to be updated before the network becomes insecure.

- **Anomaly based IDS:** Anomaly Based recognition is based on defining the network activities. Activities of the network are the predefined, when it is accepted or else it activates the event in the anomaly detection [5]. The recognized activities of the network is arranged or learned by the specifications of the network managers. The significant stage in defining the network activities is the IDS engine that is competent to cut through the different protocols at all levels. The engine is competent to execute the protocols and realize the target. With this protocol analysis is computationally expensive, the profit it produces like increasing the rule set helps in fewer false positive alarms. The main disadvantage of Anomaly based detection is defining its rule set. The effectiveness of the scheme depends on implementation and testing on all protocols that are available.

- **Specification-based IDS:** In this method, manually extended dimensions are used to explain authorized program performances. This technique is depends on vendor developed common statements to specific protocols that permit it to trace protocol conditions. Commonly, the network protocol models in specification base IDS are based on protocol standards from international standard organizations. The benefit of this method is that it doesn't produce fake alarms when legal unusual user activities are encountered. It could also identify earlier unknown attacks because of its capability to identify attacks that deviate from the specified legal activities. However, requirement development for such a method requires significant effort, which affects approach usability. In addition, the efficiency in reducing false positive is still doubtful.

Hybrid IDS: The Hybrid IDS are described by both the technique utilized to identify attacks and the assignment of the IDS on the network. IDS might execute either misuse detection or anomaly detection and might be deployed as either a network based system or a host based method. This fallout in four common groups namely misuses host, misuse network, and anomaly host and anomaly network [6]. Some IDSs merge qualities from all these categories (generally implementing both misuse and anomaly detection) and are recognized as hybrid systems. It is vital to create the key differences among anomaly detection and misuse detection approaches [7].

The major difficulties encountered by IDS are huge number of false positive alerts that are incorrectly classified as normal traffic due to security violations. A perfect IDS doesn't produce fake or irrelevant alarms. In practice, signature based IDS establish to create extra fake alarms than expected. This is due to the excessively general signatures and lack of integral authentication tool to authenticate the success of the attack. The huge amount of false positives in the alert log creates the procedure of taking remedial action for the true positives, i.e. successful attacks, delayed and effort intensive.

Rest of the paper is organized as follow: in section 2 we explained previous research work done in the field of IDS, in section 3 we give detailed overview of different machine learning algorithms used in simulation of proposed technique, section 4 present proposed work followed by results analysis and discussions in section 5. At last we conclude our paper in section 6.

2. LITERATURE SURVEY

In recent years plenty of Intrusion detection systems have been developed commercially and noncommercial that identify intrusions in the system. Latest methods are being utilized to get better success rate of such kind of schemes. Data mining techniques could handle huge dataset and permits automation of IDS. Local anomaly detection models have been extended that could identify an intrusion with an immense degree of accurateness. According to the reviewed research, two types of profiling are made.

Some IDS systems sustain a database of probable intrusion action patterns and activate alarm when such action is identified. These systems result in less fake alarms because of a difference in node usage prototypes, however, intrusion behavior with new prototypes are likely to be underreported. Another class of IDS schemes maintains a usual operational profile formed by a learning process. Anything that falls outside such a profile of behavior is categorized as a possible intrusion. These schemes have a superior fake alarm rate, but are more probable to determine unknown intrusions.

Xiao et al [8] presented a method for detection of intrusion that applies GA to identify intrusion in networks during valuable feature selection methods. Their technique utilizes information theory to mine related features and decrease the difficulty. After that, they created a linear structure rule from the selected features in order to categorize network activities into normal and anomalous activities. However, their technique considers only discrete features.

Ciaulkns et al [9] stated a dynamic data mining system for identifying anomalies utilizing decision tree in networks. Gudadhe et al [10] presented a model that utilizes boosted decision tree i.e. hoeffding tree categorization method to amplify the efficiency of the intrusion detection system. Boosting technique improves ensemble performance by utilizing adaptive window and range hoeffding tree like base learner. The primary idea of boosting is to merge simple rules to form an ensemble such that the efficiency of the single ensemble element is improved. The boosting algorithm begins by giving all data training tuples the similar weight w0. Later than a classifier is built the load of every tuple is modified according to the categorization given by that classifier. Then, a second classifier is constructing the reloaded training tuple. The concluding classification of intrusion detection is a loaded average of the individual classifications of overall classifiers.

Pan et al [11] stated a misuse detection scheme utilizing the grouping of neural network and C4.5 algorithm. Gaddam et al [12] stated supervised anomaly detection scheme by cascading K-Means clustering and ID3 Decision Tree learning algorithms. Yasami and Mozaffari [13] stated host based IDS using a grouping of K-Means clustering and ID3 Decision Tree learning algorithms used for unsupervised classification of abnormal and normal behavior in existing networks. In their proposed work, the K-Means clustering algorithm was primarily applied to the normal training data and it was division into K clusters utilizing the Euclidean distance measure. Decision Tree was created on every cluster using ID3 algorithm. The anomaly score value of the K-Means clustering algorithm and decision rules from ID3 were mined. Resultant anomaly score value was acquired using a special algorithm which merges the output of the two algorithms. The threshold rule was applied for making the decision on the test instance normality. The efficiency of the merged approach was evaluated with individual K-Means clustering, ID3 categorization algorithm and the other approaches based on Markov chains and stochastic learning automata. Improvement in correctness had been monitored in the merged approach when evaluated with other approaches.

In almost all study work, SVM has been utilized for categorization of network traffic patterns. The disadvantage with this method is that it obtains a long time for training the scheme. So, it is significant to optimize that difficulty utilizing clustering, fuzzy logic genetic algorithm and neural networks. Platt [14] stated an express training technique for SVM utilizing sequential minimal optimization. Lin and Wang [15] & Tang and Qu [16] stated Fuzzy Support Vector Machines in which a fuzzy membership to every input point was applied to reformulate the SVMs such that different input points can create different contributions to the learning of decision surface. Kim et al [17] stated a GA based approach to get better the capability of the SVM based intrusion detection models utilized in network intrusion detection systems. The rules produced in their research work were more capable in categorization of recognized and unidentified prototypes since the proposed neurotree detection pattern incorporates neural network to preprocess the data in to amplify the generalization capability.

Khan et al [18] presented a method for optimizing the training time of SVM, mainly when handling huge datasets, utilizing hierarchical clustering analysis. A dynamically rising self organizing tree algorithm for clustering was utilized by them because it has verified to conquer the problems of existing hierarchical clustering algorithms. Clustering analysis helps in discovering the edge points, which are mainly competent data prototype to train SVM, among two classes, abnormal and normal. Their algorithm added considerably in improving the training stage of SVM with superior generalization precision. A novel algorithm for multiclass SVM was proposed by Guo et al [19]. The tree build in their algorithm consists of a sequence of two class SVMs. Considering both separability and balance, in every iteration multiclass prototypes are separated into two sets according to the distances among pair wise classes and the number of prototypes in every class. This algorithm could well treat with the irregularly distributed difficulties. Lei and Zhao [20] projected a model utilized on IDS; this model is based on Rough Set theory and Fuzzy Support Vector Machines (RS-FSVM). Experimental results were shown that the RS-FSVM achieves the most excellent recognition capacity.

Mulay et al [21] projected the IDS based on SVM utilizing decision tree. Chen et al [22] applied Support Vector Machine to multiclass categorization difficulties and solved the multiclass error diagnosis tasks. They measured the restrictions of conventional techniques, and hence projected the Decision Tree based SVM (DTSVM) that utilizes genetic algorithm (GA) which preserves the higher generalization ability. In their work, decision tree was created by utilizing the GA with maximum distance to create the two subclasses as divisible as possible and hence offers relatively improved generalization capability in the majority of cases.

Yi et al [23] projected a modified Radial Basis kernel Function (U-RBF), through the mean and mean square diversity values of feature attributes inserted in Radial Basis kernel Function (RBF). They recommended an enhanced incremental kernel function U-RBF, which is based on Gauss kernel function. This method decreases the noise between attributes, so the recognition rate of the U - RBF is elevated than RBF. U-RBF plays significant role in saving the training and testing time. This technique is unsuccessful to discover user to root (U2R) and remote to local (R2L) attacks.

Lu and Traore [24] proposed Genetic Programming (GP) which is an extension of GA used for intrusion detection. The major difference among GP and GNP with GA is that GP and GNP offer a particular way for individual (chromosome) representation. In GA, preset length of vectors is utilized to represent a solution while GP encodes every chromosome utilizing a parse tree. However, GP and GNP have the flexibility to represent extremely complex individuals. In the perspective of rule supported intelligent schemes, though Genetic Algorithm is frequently utilized to efficiently extract simple rules, GP and GNP might also be utilized in its place since GP and GNP have the flexibility to represent complex rules.

The Genetic Network Programming (GNP) based fuzzy class association rule mining with sub attribute utilization was proposed by Mabu et al [25] to detection network intrusion. They utilized a graph based evolutionary optimization technique for GNP, which directs to improve the representation capability with compact programs obtained from the reusability of the nodes in the graph.

Khayyam et al [26] proposed IDS to detect the network intrusion detection using three detectors namely maximum entropy detector, rate limiting detector and credit-based threshold detector. In their work, the entropy threshold was used mainly to detect the intruders.

Apart from the above discussed IDSs, there are numerous, helpful open sources and commercial products to offer different intrusion detection performs of network based. A few of such products of IDS are as follows: SNORT [27] is one of the most excellent recognized lightweight IDSs, which targeted on performance, flexibility, and ease. It is an open source IDS that is currently in quite widespread utilize. SNORT is a network based IDS which utilizes signature based recognition techniques. It might identify various attacks and probes. Therefore, SNORT is an example of dynamic intrusion detection systems (IDS) that identifies probable attacks or access violations even as they are happening in actuality.

Cisco IOS (IDS module) offers price efficient way to organize firewall with network based intrusion detection capabilities [28]. In addition to the firewall characteristics, Cisco IOS Firewall has 59 included, static signatures to identify ordinary attacks and misuse efforts. The IDS procedure on the firewall router examines packet headers for intrusion recognition by using those 59 signatures. In a few cases routers might inspect the entire packet and preserve the state information for the connection. Upon attack recognition, the firewall might be configured to log the incident, drop the packet, or reset the connection.

The Self Monitoring Analysis and Reporting Technology (SMART) scheme observes the make for anything appears unusual, reports the information, and examines the data [29]. It will throw out alarms if some unusual functions are identified, creating parameter list for the user or system manager. Currently, the SMART scheme might identify about 70% of all hard drive errors. SMART is a hardware and software combined observe that could identify the unusual state of the hard disk according to differ 'health' attributes, and this might offer huge support to anomaly detection.

3. Machine Learning Algorithms

We checked our proposed technique on various machine learning algorithms and calculate the results of categorization, we enlighten these machine learning algorithms themselves. The machine learning algorithms illustrated below will form the base for the researches executed in our proposed technique.

BayesNet: BayesNet learns Bayesian networks under the assumptions like nominal attributes and no missing values. These two are completely dissimilar elements for estimating the conditional probability tables of the network. In our research we tend to execute BayesNet with the Simple Estimator and K2 search algorithm while not utilize AD Tree. The K2 algorithm executes as follows. Assume we identify the complete ordering of the nodes. At the beginning each node has no folks. The algorithm then incrementally appends the parent whose addition will enhance most of the score of the ensuing structure. Once no addition of one parent will enhance the score, it stops appending parents to the node. Since an ordering of the nodes is understood beforehand, the search domain below this constraint is far smaller than the complete domain. And that we don't have to be compelled to check for cycles, since the complete ordering guarantees that there's no cycle within the deduced structures.

NaiveBayes Classifier: Along with decision trees, neural networks, nearest neighbors, one of the most practical learning techniques is NaiveBayes Classifier. NaiveBayes Classifier is utilized in when moderate or huge training set obtainable or when Attributes that explain instances are provisionally independent given classification suppose objective function $f : X \rightarrow V$, where every instance x explained by attributes $\{a1, a2, \ldots\ an\}$.

$$v_{NB} = \arg\max_{v_j \epsilon V} P(v_j) \prod_i P(a_i|v_j)$$

J48 Decision Trees: The J48 Decision tree is a predictive machine learning model that decides the target value of a new example based on different attribute values of the obtainable data. The internal nodes of a decision tree stand for the distinctive attributes, while the branches between the nodes provides information about the possible values that these attributes can attain in the observed samples, while the terminal nodes informs about the final value (classification) of the dependent variable. In J48 method, in order to classify a new item, it first needs to create a decision tree based on the attribute values of the obtainable training data. So, as quickly as it encounters a set of items (the training set) it recognizes the attribute that discriminates the other instances most evidently. This feature that is able to identify the data most accurately 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 falls inside its type have the similar value for the objective variable, then we terminate that branch and assign to it the target value that we have obtained.

J48graft:- J48graft produces a grafted Decision Tree from a J48 tree. The graft method appends nodes to an existing decision tree with the endeavor of reducing prediction errors. These algorithms identifies areas of the instance space that don't appear to be occupied by training instances, or occupied exclusively by misclassified training instances, and obtain into account different classifications for those areas. In alternative words, replacement tests are performed within the leaf, generating new branches that may cause new classifications. Graft is an algorithmic rule for addition of nodes to the tree as a postprocessor. Its reason is to widen the possibility of justly categorizing instances that plunge outside the areas covered by the training information. Graft might be a postprocessor which will be applied to decision trees. Its endeavor is to reduce prediction error by reclassifying regions of the instance space wherever no training information exists or wherever there's solely misclassified knowledge. Its aim is to seek out the most effective matched cuts of existing leaf regions and branches intent on produce new leaves with alternative classifications than the initial. Although tree becomes a lot of advanced, however here solely branching that doesn't introduce any classification errors in knowledge already justly classified is taken into account. Fresh generated tree therefore cut back errors rather than introduce them.

Random Forests: Random Forests grows a large number of classification trees. To classify a new object from a given input vector, the input vector is put down to each of the trees in the forest constructed before. Each tree gives its own classification, and we say that the tree has “voted” for that class. The forest chooses the classification having the maximum votes (over all the trees present in the forest). Each tree is grown in the following manner: If the number of cases in the training set is P, sample P cases at random - but with replacement, from the original data. This sample would be the training set for increasing the tree. If there are Q input variables, a number q¡¡Q is specified such that at each node, q variables are selected at random out of the Q and the best split on this q is used to split the node. The value of q is held constant during the forest growing. Each tree is grown to the largest extent possible. There is no reducing. When the training set for the present tree is drawn by sampling with replacement, about one-third of the cases are left out of the sample. After each tree is built, all of the data are run down the tree, and proximities are calculated for every pair of cases. If two cases engage the similar terminal node, their proximity is bigger by one. At the end of the execution, the proximities are normalized by separating by the number of trees. Proximities are utilized in replacing missing data, locating outliers, and producing illuminating low dimensional views of the data. Based on these proximities that categorization is complete.

4. PROPOSED WORK

The following algorithm 1 has been used as proposed algorithm. The KDD cup99 dataset contains a lot of individual attacks like apache2, back, xterm etc, initially the dataset is trimmed in order to remove the outliers. After the outliers are removed then, the individual attacks are replaced by their category as shown in Algorithm I.

Algorithm 1:

```
Input: KDD cup 99 Dataset
Output: WEKA compatible .ARFF file in which all attacks are classified individually
Step 1: Outlier removal from dataset
Step 2: If attack_read == ‘apache2’ then // search for all ‘apache2’ attacks
replace attack by Cat1 in KDD dataset
Step 3: else If attack_read == ‘back’ then // search for all ‘back’ attacks
replace attack by Cat2 in KDD dataset // repeat this step for all individual attacks in KDD cup99 dataset
Step n: else If attack_read == ‘xterm’ then // search for all ‘xterm’ attacks
replace attack by Catn in KDD dataset
```

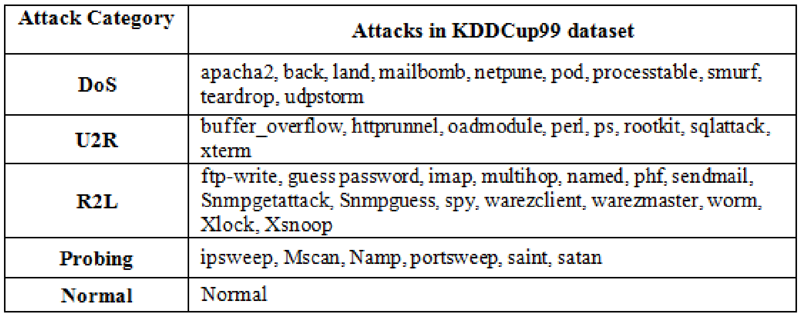
Step n+1: else Mark as Normal in KDD cup dataset
Step n+2: compile the file and save as individual_attacks.arff
After getting individual_attacks.arff, we execute it on WEKA tool to verify the efficiency of classification of our proposed method. For this purpose we used BayesNet, NaiveBayes, J48, J48graft, Random Forests classifiers.

5. Result Analysis

To evaluate proposed method we used WEKA 3.7.12 tool (Waikato Environment for Knowledge Analysis) is an open source machine learning software scripting in Java, designed at the Waikato University, New Zealand [30].

Our technique was analyzed utilizing the KDD Cup99 network intrusion dataset [31]. It arrives from DARPA 98 Intrusion Detection assessment knobbed by Lincoln laboratory at MIT. This is a reviewed set of standard dataset which includes training and testing set. Training set is about 4 GB of compacted binary TCP chuck data from 7 weeks of network traffic with about 5 million connections. Similarly, test set includes two weeks of network traffic with around 3 million connections [32]. In this work, we used KDD data set (corrected.zip) which consists of (311029 records). Table II shows the number of samples. Then, 10% of data is extracted by sampling. 66% of this new set belonged to training set, and 34% dedicated to test set. Furthermore, in KDD Cup dataset, there are 37 attack types which are available in test set but only 23 of them are found in training set. Therefore, test set can be used to estimate the detection rate for new or unknown attacks.

Table I: Attack Categories of KDDcup99 dataset



Attack Category	Attacks in KDDCup99 dataset
DoS	apacha2, back, land, mailbomb, netpune, pod, processtable, smurf, teardrop, udpstorm
U2R	buffer_overflow, httprunnel, oadmodule, perl, ps, rootkit, sqlattack, xterm
R2L	ftp-write, guess password, imap, multihop, named, phf, sendmail, Snmpgetattack, Snmpguess, spy, warezclient, warezmaster, worm, Xlock, Xsnoop
Probing	ipsweep, Mscan, Namp, portsweep, saint, satan
Normal	Normal

Table II. Number of Samples in the Kddcup99 Test dataset

Attack Category	Number of Samples
Normal	60589
DoS	229853
R2L	16179
U2R	228
Probe	4165
Total	311014

Accuracy and Detection Rate of proposed method

The metrics utilized for measuring the performance of proposed technique are accuracy (i.e., Precision) and recognition rate (i.e., Recall). Precision is the percentage of the total number of attacks that are properly detected. It is determined by the following equation:

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

Detection Rate is described as the number of attacks detected by the proposed technique to the total number of attacks truly there.

$$\text{DetectionRate (Recall)} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Here, TP is True Positive, FP is False Positive, and FN is False Negative. A TP is a case, which is truly an attack and is acknowledged as attack by the proposed technique. A FP occurs when proposed technique treats a normal action as attack action. A FN arises when the proposed technique treats an attack action as normal. A measure that merges precision and recall is the harmonic mean of precision and recall is recognized as F measure.

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

Receiver Operator Characteristics (ROC): ROC illustrates the exchange among sensitivity and specificity. ROC curves plot the TP rate vs. the FP rate, at differing threshold cutoffs.
The experimental results of precision with various Classifiers used are shown in Table III. As can be seen the performance of NaiveBayes Classifier is below par. Similarly Tables IV - VI show the performance of Recall, F – measures, ROC of the Bayes Net, NaiveBayes, J48, J48 graft and Random Forrest for detection of individual attacks. Thus three J48, J48 Graft and Random Forrest classifier detect the various attack categories of KDDCup'99 dataset efficiently. As a result, these two algorithms can be successfully deployed on any machine learning based IDS in order to detect the attack categories shown in Table I.

Table III. PRECESSION of all Classifier for Individual Attacks in kddcup99 dataset

	Bayes Net	**NaiveBayes**	**J48**	**J48 Graft**	**Random Forest**	**Individual attack Class**
	1	0.861	0.982	0.982	0.996	**apache2**
	0.995	0.992	0.997	1	1	**back**
	0.097	0.014	0.417	1	0.5	**buffer_overflow**
	0.999	0.876	0.996	0.998	0.999	**guess_passwd**
	0.644	0.487	0.963	0.981	1	**httptunnel**
	0.514	0.196	1	1	0.989	**ipsweep**
	0.031	0	1	1	1	**land**
	1	0.943	0.998	0.996	1	**mailbomb**
	0.948	0.756	0.989	0.984	1	**mscan**
	1	1	1	1	1	**neptune**
	0.813	1	1	1	1	**nmap**
	0.998	0.999	0.953	0.953	0.954	**normal**
	0.567	0.071	0.868	0.892	0.971	**pod**
	0.818	0.79	0.991	0.991	1	**portsweep**
	1	0.992	1	1	1	**processtable**
	0.211	0.005	0.333	1	1	**ps**
	0.788	0.026	0.95	0.95	0.93	**saint**
	0.884	0.71	0.964	0.966	0.978	**satan**
	1	0.996	1	1	1	**smurf**
	0.411	0.383	0.618	0.618	0.618	**snmpgetattack**
	0.928	0.195	0.999	0.999	1	**snmpguess**
	0.804	0.748	0.989	0.988	0.991	**warezmaster**
Weighted Avg.	**0.981**	**0.965**	**0.981**	**0.981**	**0.981**	

Table IV. RECALL of all Classifier for Individual Attacks in kddcup99 dataset

	Bayes Net	NaiveBayes	J48	J48 Graft	Random Forest	Individual attack Class
	0.996	0.996	0.996	0.996	0.996	apache2
	1	1	0.995	0.995	1	back
	1	0.714	0.714	0.571	0.714	buffer_overflow
	0.993	0.844	0.994	0.994	1	guess_passwd
	0.983	0.949	0.881	0.881	0.983	httptunnel
	0.979	0.968	0.979	0.979	0.979	ipsweep
	1	0	1	1	1	land
	1	0.976	1	1	1	mailbomb
	0.995	0.978	0.986	0.986	1	mscan
	0.996	0.995	1	1	1	neptune
	1	1	1	1	1	nmap
	0.797	0.636	0.947	0.947	0.948	normal
	0.971	0.914	0.943	0.943	0.943	pod
	0.968	0.904	0.92	0.928	0.976	portsweep
	0.996	0.98	0.996	0.992	1	processtable
	0.444	0.778	0.111	0.111	0.333	ps
	0.746	0.02	0.93	0.93	0.922	saint
	0.927	0.973	0.976	0.974	0.958	satan
	1	0.999	1	1	1	smurf
	0.992	0.606	0.654	0.654	0.65	snmpgetattack
	0.996	0.971	0.996	0.996	0.996	snmpguess
	0.998	0.442	0.988	0.989	0.993	warezmaster
Weighted Avg.	**0.958**	**0.909**	**0.98**	**0.98**	**0.98**	

Table V. F-Measure of all Classifier for Individual Attacks in kddcup99 dataset

	Bayes Net	NaiveBayes	J48	J48 Graft	Random Forest	Individual attack Class
	0.998	0.924	0.989	0.993	0.996	apache2
	0.997	0.996	0.996	0.997	1	back
	0.177	0.028	0.526	0.727	0.588	buffer_overflow
	0.996	0.859	0.995	0.996	0.999	guess_passwd
	0.779	0.644	0.92	0.929	0.991	httptunnel
	0.674	0.326	0.989	0.989	0.984	ipsweep
	0.06	0	1	1	1	land
	1	0.959	0.999	0.998	1	mailbomb
	0.971	0.853	0.988	0.985	1	mscan
	0.998	0.998	1	1	1	neptune
	0.897	1	1	1	1	nmap
	0.886	0.777	0.95	0.95	0.951	normal
	0.716	0.132	0.904	0.917	0.957	pod
	0.886	0.843	0.954	0.959	0.988	portsweep
	0.998	0.986	0.998	0.996	1	processtable
	0.286	0.01	0.167	0.2	0.5	ps
	0.766	0.023	0.94	0.94	0.926	saint
	0.905	0.821	0.97	0.97	0.968	satan
	1	0.997	1	1	1	smurf
	0.582	0.469	0.636	0.636	0.634	snmpgetattack
	0.961	0.324	0.998	0.998	0.998	snmpguess
	0.89	0.556	0.989	0.989	0.992	warezmaster
Weighted Avg.	**0.964**	**0.926**	**0.98**	**0.98**	**0.981**	

Table VI. ROC of all Classifier for Individual Attacks in kddcup99 dataset

	Bayes Net	NaiveBayes	J48	J48 Graft	Random Forest	Individual attack Class
	1	0.999	0.998	0.998	1	apache2
	1	1	1	0.997	1	back
	1	0.975	0.928	0.856	0.929	buffer_overflow
	1	0.994	0.997	0.997	1	guess_passwd
	0.998	0.997	0.961	0.942	0.992	httptunnel
	0.993	0.988	0.994	0.994	0.998	ipsweep
	1	1	1	1	1	land
	1	0.999	1	1	1	mailbomb
	1	1	0.997	0.999	1	mscan
	1	0.999	1	1	1	neptune
	1	1	1	1	1	nmap
	0.997	0.98	0.998	0.998	0.998	normal
	1	0.976	0.971	0.971	1	pod
	1	0.999	0.988	0.988	1	portsweep
	1	0.998	0.998	0.998	1	processtable
	0.997	0.992	0.886	0.83	0.944	ps
	0.999	0.969	0.992	0.988	0.988	saint
	1	0.999	0.996	0.994	0.994	satan
	1	0.997	1	1	1	smurf
	0.989	0.977	0.99	0.99	0.99	snmpgetattack
	1	0.993	0.999	0.999	1	snmpguess
	1	0.997	0.997	0.998	1	warezmaster
Weighted Avg.	**0.999**	**0.994**	**0.999**	**0.999**	**0.999**	

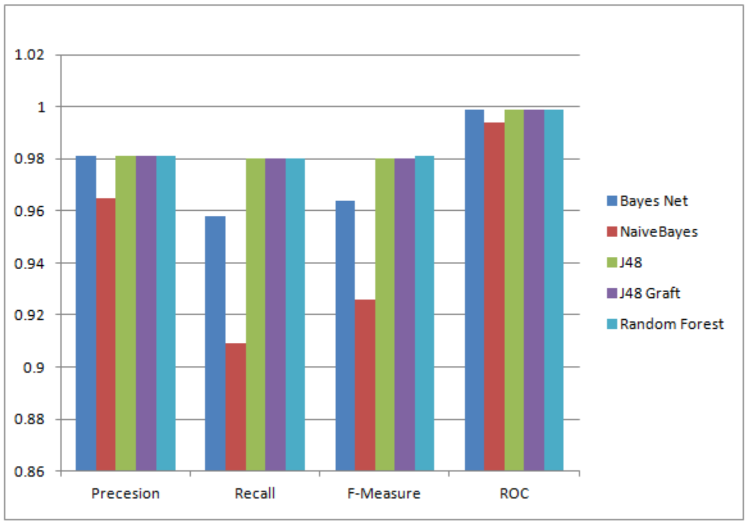


Fig. 1: Accuracy of classification of all classifiers with proposed method

6. CONCLUSION & FUTURE WORK

Presently so many techniques, method and tools are uses to detect Intrusion in computer network and continues research is being carried out to make them even better to recognize intrusion. But simultaneously new attacks arrived which will difficult to Handel because they continues changes their behaviours. In this research work we proposed "an improved method to detect intrusion using machine learning algorithms" with KDDCUP 99 dataset, which is simulated on WEKA tool. The proposed method detects individual attacks presents in KDDCUP 99 dataset fast and efficiently. Detection rate of all three machine learning algorithm J48, J48 Graft and Random Forrest is above 96 %. These algorithms can be modified according to environment of network to make better detection rate and time. In future we can modify default WEKA parameters with reduction of features of KDDCUP 99 dataset. Apart from combination of Machine learning algorithms and data mining methods we could use artificial intelligence and soft computing methods like neural networks etc which would helpful to reduce false alarm rate in intrusion detection.

REFERENCES

[1] Sourcefire. Snort open source network intrusion prevention and detection system (ids/ips). http://www.snort.org.

[2] Liao, Hung-Jen, Chun-Hung Richard Lin, Ying-Chi Lin, and Kuang-Yuan Tung. "Intrusion detection system: A comprehensive review." Journal of Network and Computer Applications 36, no. 1 (2013): 16-24.

[3] Debar, Herve. "An introduction to intrusion-detection systems." Proceedings of Connect 2000 (2000).

[4] Liao, Hung-Jen, Chun-Hung Richard Lin, Ying-Chih Lin, and Kuang-Yuan Tung. "Intrusion detection system: A comprehensive review." Journal of Network and Computer Applications 36, no. 1 (2013): 16-24.

[5] Jyothsna, V., VV Rama Prasad, and K. Munivara Prasad. "A review of anomaly based intrusion detection systems." International Journal of Computer Applications 28, no. 7 (2011): 26-35.

[6] Bashah, Norbik, Idris Bharanidharan Shanmugam, and Abdul Manan Ahmed. "Hybrid intelligent intrusion detection system." World Academy of Science, Engineering and Technology 11 (2005): 23-26.

[7] Ghosh, Anup K., Aaron Schwartzbard, and Michael Schatz. "Learning Program Behavior Profiles for Intrusion Detection." In Workshop on Intrusion Detection and Network Monitoring, vol. 51462. 1999.

[8] Xia, Tao, Guangzhi Qu, Salim Hariri, and Mazin Yousi. "An efficient network intrusion detection method based on information theory and genetic algorithm." In Performance, Computing, and Communications Conference, 2005. IPCCC 2005. 24th IEEE International, pp. 11-17. IEEE, 2005.

[9] Caulkins, Bruce D., Joohan Lee, and Morgan Wang. "A dynamic data mining technique for intrusion detection systems." In Proceedings of the 43rd annual Southeast regional conference-Volume 2, pp. 148-153. ACM, 2005.

[10] Gudadhe, Mrudula, Prakash Prasad, and Kapil Wankhade. "A new data mining based network intrusion detection model." In Computer and Communication Technology (ICCCT), 2010 International Conference on, pp. 731-735. IEEE, 2010.

[11] Pan, Zhi-Song, Song-Can Chen, Gen-Bao Hu, and Dao-Qiang Zhang. "Hybrid neural network and C4. 5 for misuse detection." In Machine Learning and Cybernetics, 2003 International Conference on, vol. 4, pp. 2463-2467. IEEE, 2003.

[12] Gaddam, Shekhar R., Vir V. Phoha, and Kiran S. Balagani. "K-means+ id3: A novel method for supervised anomaly detection by cascading k-means clustering and id3 decision tree learning methods." Knowledge and Data Engineering, IEEE Transactions on 19, no. 3 (2007): 345-354.

[13] Yasami, Yasser, and Saadat Pour Mozaffari. "A novel unsupervised classification approach for network anomaly detection by k-Means clustering and ID3 decision tree learning methods." The Journal of Supercomputing 53, no. 1 (2010): 231-245.

[14] Platt, J.C. “Fast Training of Support Vector Machines using Sequential Minimal Optimization", Advances in Kernel Methods: Support Vector Learning, pp. 185-208, 1998.

[15] Lin, Chun-Fu, and Sheng-De Wang. "Fuzzy support vector machines."Neural Networks, IEEE Transactions on 13, no. 2 (2002): 464-471.

[16] Tang, Hao, and Liang-sheng Qu. "Fuzzy support vector machine with a new fuzzy membership function for pattern classification." In Machine Learning and Cybernetics, 2008 International Conference on, vol. 2, pp. 768-773. IEEE, 2008.

[17] Kim, Dong Seong, Ha-Nam Nguyen, and Jong Sou Park. "Genetic algorithm to improve SVM based network intrusion detection system." In Advanced Information Networking and Applications, 2005. AINA 2005. 19th International Conference on, vol. 2, pp. 155-158. IEEE, 2005.

[18] Khan, Latifur, Mamoun Awad, and Bhavani Thuraisingham. "A new intrusion detection system using support vector machines and hierarchical clustering."The VLDB Journal—The International Journal on Very Large Data Bases 16, no. 4 (2007): 507-521.

[19] Guo, Jun, Norikazu Takahashi, and Wenxin Hu. "An efficient algorithm for multi-class support vector machines." In Advanced Computer Theory and Engineering, 2008. ICACTE'08. International Conference on, pp. 327-331. IEEE, 2008.

[20] Li, Lei, and Ke-nan Zhao. "A new intrusion detection system based on rough set theory and fuzzy support vector machine." In Intelligent Systems and Applications (ISA), 2011 3rd International Workshop on, pp. 1-5. IEEE, 2011.

[21] Mulay, Snehal, P. R. Devale, and G. V. Garje. "Decision tree based support vector machine for intrusion detection." In Networking and Information Technology (ICNIT), 2010 International Conference on, pp. 59-63. IEEE, 2010.

[22] Chen, Huanhuan, Qiang Wang, and Yi Shen. "Decision tree support vector machine based on genetic algorithm for multi-class classification." Systems Engineering and Electronics, Journal of 22, no. 2 (2011): 322-326.

[23] Yi, Yang, Jiansheng Wu, and Wei Xu. "Incremental SVM based on reserved set for network intrusion detection." Expert Systems with Applications 38, no. 6 (2011): 7698-7707.

[24] Lu, Wei, and Issa Traore. "Detecting new forms of network intrusion using genetic programming." Computational Intelligence 20, no. 3 (2004): 475-494.

[25] Mabu, S., Chen, C., Lu, N., Shimada, K. and Hirasawa, K. “An Intrusion-Detection Model Based on Fuzzy Class-Association-Rule Mining Using Genetic Network Programming”, IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews,Vol. 40, No 99, pp. 1-10, 2010

[26] Khayam, Syed Ali, Ayesha Binte Ashfaq, and Hayder Radha. "Joint network-host based malware detection using information-theoretic tools." Journal in computer virology 7, no. 2 (2011): 159-172.

[27] Roesch, Martin. "Snort: Lightweight Intrusion Detection for Networks." InLISA, vol. 99, no. 1, pp. 229-238. 1999.

[28] Cisco, I. O. S. "NetFlow." (2008).

[29] Rothberg, Michael S. "Disk drive for receiving setup data in a self monitoring analysis and reporting technology (SMART) command." U.S. Patent 6,895,500, issued May 17, 2005.

[30] Hall, Mark, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. "The WEKA data mining software: an update." ACM SIGKDD explorations newsletter 11, no. 1 (2009): 10-18.

[31] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[32] Wu, Su-Yun, and Ester Yen. "Data mining-based intrusion detectors." Expert Systems with Applications 36, no. 3 (2009): 5605-5612.