Noname manuscript No.
(will be inserted by the editor)

Two-Stage R2L and U2R classifier using Feature Selection

Ajay Kamath · **Praveen Kaushik**

Received: date / Accepted: date

Abstract Anomaly based Intrusion Detection System (IDS) uses machine learning techniques to detect novel attacks. Naive Bayes is one such classifier which predicts the class of an attack. While it detects DoS (Denial of Service) and probe attack with reasonable accuracy, its detection rate fails miserably in case of novel R2L (Remote to Local) and U2R (User to Root) attacks. One solution to improve the detection rate of R2L and U2R attacks is to use a two-stage classifier with feature selection in the second stage. The first stage uses a regular Naive Bayes classifier. Genetic Algorithm employing entropy based fitness function is used as a feature selection technique. Experiments were conducted on the NSL-KDD dataset using WEKA machine learning tool. The proposed approach detects 86.2% of the R2L attacks and 91% of the U2R attacks with a reduced false positive rate of 1.3% and 2.1% for both respectively. It is found that the detection rate of the proposed approach is the lowest for UDP based Snmpguess and Snmpgetattack.

Keywords Intrusion Detection · Naive Bayes classifier · Feature selection · Genetic algorithm

1 Introduction

With the advancement in network based services and information transfer, it is highly imperative to provide security. Generally, when an application is connected to a network, it becomes vulnerable to attacks by malicious intruders who try to steal information. An intruder masquerading as a legitimate user tries to steal information and compromise the security. Organizations generally have certain amount of critical data and applications that need to be secured to protect their confidentiality, integrity and availability.

Intrusion Detection System (IDS) is a mechanism that detects malicious and unauthorized activity inside a network. Firewalls are not enough to protect networks. Firewalls protect only the boundaries of the network, detect limited attacks and their database needs to be updated regularly. An Intrusion is an illegal activity that is done by penetrating the system boundaries with the intent of stealing vital information, denying access to a legitimate user or to infect the system with malicious packets. An Intrusion Detection System (IDS) detects any intrusions and unauthorized activity inside a system. It raises alarms when any suspicious activity is found which can then be supervised upon by the network security administrator or the Intrusion Prevention System (IPS). With bulky data commonly traversing across the network, the amount of alerts generated by these systems is very high. Moreover, these systems are not perfect in detecting attacks with increasing number of novel attacks found every day. They are also optimized with latest state of the art techniques to automate and mitigate the task of the security expert. Nowadays, organizations use Intrusion Detection Systems coupled with a firewall to secure the network.

Intrusion Detection approaches are traditionally studied from two major viewpoints, signature detection and anomaly detection [1]. Signature detection or misuse detection compares the incoming packets with the existing database of attacks. It uses pattern matching algorithms and rule based approaches to find a match in the previously defined rules. The detection rate is very high in this case whereas the false positive rate is very low. The main disadvantage of this detection is that it cannot detect previously known attacks. In stark

Ajay Kamath
Department of Computer Science and Engineering
Maulana Azad National Institute of Technology, Bhopal
E-mail: ajaytaker3@gmail.com

Praveen Kaushik
Department of Computer Science and Engineering
Maulana Azad National Institute of Technology, Bhopal

contrast to this approach is the anomaly detection method which assesses the incoming packets for any abnormality. An Anomaly is a deviation from the normal behavior. It compares the packet with the normal behavior that is predefined. For e.g. If the number of failed login attempts is more than the specified threshold, then it can be termed as an anomaly. The main advantage of using this detection is that it can detect previously unknown attacks. However, due to novelty in detection, the error rate is very high. Error is introduced in terms of the false positive rate. False positives are those normal packets that are wrongly classified as attacks. Researchers have classified the attack into mainly four different categories (DoS, U2R, R2L and Probe). While it is seen that DoS and probe attacks are easily detected by the Naive Bayes classifier, the detection rate of U2R and R2L attacks is very low [1,2].

A number of machine learning techniques like Classification and clustering, Support Vector Machines (SVM), Decision trees, Artificial Neural Networks (ANN), Markov Models, Swarm Intelligence etc. are used as predictive models to classify the activity as normal or an attack. Bayesian based classifier is a supervised learning algorithm and is based on the Bayes theorem of conditional probability [3]. It assumes mutual independence amongst the features. It calculates the probabilities for attack classes and in normal TCP traffic for different features. The classifier is trained by giving it classified traffic. It then corrects the probabilities for each feature. During the test phase, the classifier estimates the probabilities for each TCP connection and predicts the class of the each instance.

The main aim of this paper is to improve the detection rate of U2R and R2L attacks by using the two-stage Naive Bayes Classifier using the Feature Selection method. Genetic Algorithm is used to remove unwanted features from the dataset. The rest of the paper is organized as follows. Topic 2 discusses the related work followed by a brief description of the NSL-KDD dataset in 3. The proposed solution to improve the detection rates of attacks is discussed in 4 followed by the experimental setup in 5 and the improved results in 6. The paper is concluded in Topic 7.

2 Related Work

Several classification techniques were proposed earlier and are discussed below. Hesham Altwaijry and Saeed Algarny in [3] proposed a Naive Bayes Classifier that detected DoS and probe attacks with better accuracy than R2L and U2R attacks. They used the KDD Cup dataset to test the efficiency of their algorithm. The classifier was tested with selected features to find that the detection rate increases. For R2L attacks, features 23, 24 and 31 were found to be most important for R2L attacks. For R2L and U2R attacks, they proposed a multi-stage Naive Bayes Classifier in [4] where each stage was trained to detect a different type of attack. They achieved a significant improvement with a detection rate of 85.35% for R2L attacks. Detection of DoS and probe attacks is easier as compared to R2L and U2R attacks. This is because the DOS and probing attacks involve many connections to some host(s) in a very short period of time, but the R2L and U2R attacks are embedded in the data portions of packets, and normally involve only a single connection [2].

One of the interesting fields of research in the area of Intrusion Detection is feature selection. Removal of select features from the dataset significantly reduces the complexity and the running time of the algorithm and improves the detection rate of attacks. Guo et.al [5] proposed a feature selection method based on the rough set theory and genetic algorithm to detect attacks. Rough set theory was used to find the feature subset. Improved Genetic Algorithm based on clustering was used to find the optimal subset. They found features 1,2,4,5,6,11,22,23,31,33 and 35 to be the most relevant. Feature selection based on meta-heuristic evolutionary techniques provides better results than wrapper methods in finding an approximate solution. Zhou et.al [6] used genetic algorithm to remove redundant features and then applied Support Vector Machine as a classifier to improve detection rates. Researchers have used different ranker methods and wrapper methods in [7] to evaluate the best features that majorly contribute to R2L and U2R attacks. Generally, the novel attacks are variations of known attacks, particularly employing the same features as those in the known attacks. Hence, the involvement of each feature for a particular attack must be thoroughly evaluated before carrying out the actual classification task. Current advancements in IDS show that a probabilistic classifier like Simple Nave Bayes, Artificial Neural Networks (ANN) and decision trees dont have good detection rates for R2L and U2R attacks [12].

Deshmukh et.al in [8] tested the Naive Bayes Classifier on the NSL-KDD dataset with Correlation based Feature Selection (CFS) to improve the overall detection accuracy from 76.56% in [9] to 88.2%. However, significant improvement could be made in improving the number of R2L records detected. They found that NB Tree provided the best results but was computationally more expensive than Simple NB. Ibrahim et.al [11] used Self-Organizing Maps (SOM) based Artificial Neural Networks (ANN) and discovered that while SOMs possess fast conversion rates, for NSL-KDD dataset, the overall detection rate of attacks is 75.49%. Bhupendra Ingre and Anamika Yadav in [15] used ANN to evaluate the NSL-KDD dataset. Their Proposed method uses Levenberg-Marquardt (LM) and BFGS quasi-Newton backpropagation for updating the weight and bias. They obtained an improved detection rate of 34.6% for R2L and 10.5% for U2R attacks.

After extensive research, it was found that a two-stage classifier could produce better results with each stage fil-

Table 1 List of features in NSL-KDD dataset

ID	Name	ID	Name
1	duration	22	is_guest_login
2	protocol_type	23	count
3	service	24	srv_count
4	flag	25	serror_rate
5	src_bytes	26	srv_serror_rate
6	dst_bytes	27	rerror_rate
7	dos	28	srv_rerror_rate
8	wrong_fragment	29	same_srv_rate
9	urgent	30	diff_srv_rate
10	hot	31	srv_diff_host_rate
11	num_failed_logins	32	dst_host_count
12	logged_in	33	dst_host_srv_count
13	num_compromised	34	dst_host_same_srv_rate
14	root_shell	35	dst_host_diff_srv_rate
15	su_attempted	36	dst_host_same_src_port_rate
16	num_root	37	dst_host_srv_diff_host_rate
17	num_file_creations	38	dst_host_serror_rate
18	num_shells	39	dst_host_srv_serror_rate
19	num_accesss_files	40	dst_host_rerror_rate
20	num_outbound_cmds	41	dst_host_srv_rerror_rate
21	is_host_login		

tering attacks. Naive Bayes was used because of its faster speed of execution, its ease of construction and reasonable accuracy. Similarly, the false positive rate could be reduced by using removing unwanted features in the training set.

3 Analysis of the NSL-KDD dataset

The KDD 99 cup dataset [11] is a standard benchmark for testing out new Intrusion Detection algorithms. Each record is a TCP connection between two hosts and is classified as a normal connection or into four attack types; DoS, U2R, R2L and probe.

DoS (Denial of Service): These types of attacks prevent the legitimate users the access to a resource or a particular service.

U2R attack: A local user who does not have super-user privileges tries to gain root access.

R2L attack: A remote attacker who does not have access to the local machine tries to gain local access.

Probe: Stealth scanning and surveillance of the hosts on the network for open IP addresses and ports.

Although KDD 99Cup dataset is a standard for testing Intrusion Detection algorithms, it suffers from certain drawbacks defined in [10]. It is found that a number of DoS and probe attack packets are redundant so that any classifier produces biased results towards them. To overcome these drawbacks, the NSL-KDD dataset (2010) is used nowadays [14]. Although it is not a real representation of a network, it does not include the redundant records in the train set so that the classifier will not be biased towards the more frequent records. There are 125973 record connections in the train set with four attack classes mentioned above.

Table 2 Number of records for each class

Class type	Number of records in Training set	Number of records in Test set
Normal	67343	9711
DoS	45927	7458
U2R	52	200
R2L	995	2754
Probe	11656	2421
Total	125973	22544

Table 3 List of Attacks

Attack type	List of Attacks in Training Set	List of new Attacks in test set
R2L	FTP Wrtie, Guess Password, Imap, phf, spy, multihop, warezclient, warezmaster	Sendmail, named, snmpgetattack, snmpguess, xlock, xsnoop, worm
U2R	Buffer Overflow, Loadmodule, perl, rootkit	httptunnel, ps, xterm, sqlattack

Each connection in the dataset consists of 41 features that are mainly divided into four categories: Basic features (Features 1-9), Content Based features (Features 10-22) , Time-based Traffic features (Features 23-31) and Host-based Traffic features (Features 32-41). The detailed names of all features are given in Table 1. There are 22,544 record instances in the test set (Table 2). There are a total of 23 attack types in the training set of the NSL-KDD dataset with additional 17 attack types in the test set. The list of R2L and U2R attack types are given in Table 3.

4 Feature Selection

Selecting subset of features for use in the classifier model construction is called as feature selection. There are features that majorly contribute to a particular attack, the rest are redundant features. Removal of these redundant features will shorten the training times; improve the prediction accuracy rate of the classifier as well as simplify the classifier model construction [5–8]. Each feature selection technique employs a search algorithm for finding the subset and an evaluation function to give a score to that subset. Filter methods find an approximate subset whereas wrapper methods use a classifier to evaluate the subset. Wrapper methods give an optimal solution but take longer time to produce an output than the filter methods [7].

4.1 Feature Selection using Genetic Algorithm

Genetic Algorithm is a model of machine learning which derives its behavior from the process of evolution in nature. A population of candidate solutions to an optimization problem is evolved toward better solutions. Initially a population of randomly generated individuals is selected and they

are operated upon by an iterative process to create a new population. Each newly generated population is called as a generation. The populations of individuals are represented by Chromosomes. Each chromosome is made up of genes. In essence, a genetic algorithm characteristically results in an offspring that is genetically identical to a parent. From the population of individuals, fitter parents are selected for crossover to produce fitter individuals. The fitness of an individual is determined by a known fitness function. It retains the fitter solutions in the population whereas removes the less fit solutions from the population. With successive generations, we can see that the average fitness of the population increases and the algorithm can be terminated when the solution is found. Mutation is another operation performed on individuals and it is used to maintain the genetic diversity from one generation to the other. It involves arbitrary changing of the genes of the chromosome. It is necessary to consider two things in general: exploration and exploitation. Exploration means searching for the solution throughout the search space and is provided by the crossover operation while the latter involves producing better solutions within the local region and is provided by mutation.

Using Genetic Algorithm as a feature selection technique is time consuming. However, features are selected in the training stage where the time taken to produce an output is not a factor which needs to be improved. GA was used because of its effectiveness in finding a globally optimal solution. The proposed algorithm does not consider nominal features 2,3,4 and they can be removed from the selection process. As a result, the algorithm works on 38 features making use of the initial solutions from the dataset to determine their relevance to attacks. The first step in a Genetic Algorithm is the encoding. In the proposed approach, for each record connection, the value of a feature was converted to a bit binary gene value 1 or 0 indicating that a feature was selected or rejected respectively [6]. If the value of the feature is nonzero, then the gene value is 1, otherwise it is 0. For e.g. A record with values (0,tcp,ftp data,SF,491,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0,0,0,0,1,0,0,150,25,0.17,0.03,0.17,0,0,0,0.05,0) is encoded as
01000000000000000000110000100111111000 10.
The proposed feature selection technique based on Genetic Algorithm assigns weights to each feature to calculate the fitness of the chromosome. Weights are assigned using the concept of Information Theory. According to this concept, each feature carries a certain amount of information and provides a contribution to the class attribute. This information is not the actual value for that particular feature, but depends on the frequency with which a particular attribute is found to be contributing to a particular attack [5].

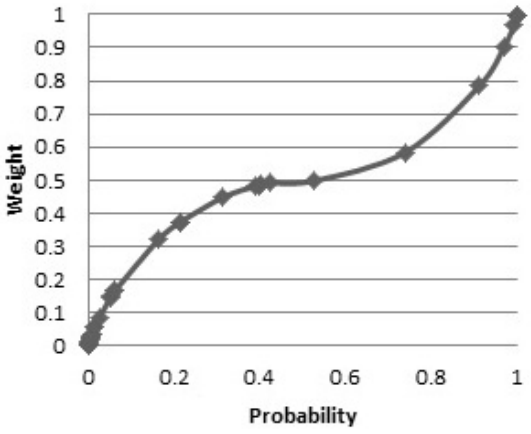

Fig. 1 Weight distribution of features for R2L attacks

The average amount of information contained in a feature is called as entropy and for a discrete feature it is defined by the formula (1).

$$H(X) = \sum_i P(X_i).log(P(X_i)) \tag{1}$$

Where, X is a particular feature and $P(X_i)$ is the probability that X takes a value i (0 or 1) P(X = i). The entropy of a feature has a minimum value of 0 and a maximum value of 1. When its value is 0, it means that the feature does not occur in all record connection or that it occurs most frequently in all record connection. When the entropy has a value 1, it means the feature has a 0.5 probability of occurrence within all record connections. Formula (2) derived from Formula (1) is applied after the encoding step.

$$H(X) = -1.(\frac{n0}{n0+n1}.\log\frac{n0}{n0+n1} + \frac{n1}{n0+n1}.\log\frac{n1}{n0+n1}) \tag{2}$$

Where n1 represents the number of 1s in a chromosome and n0 represents the number of 0s in a chromosome. More the number of 1s for a particular feature for all chromosomes, higher is the weight. The weight of a feature is calculated using formula (3). Hmax is the maximum entropy of the feature in the feature set.

$$\begin{aligned} w_i &= \frac{H(X_i)}{2} && if \;\; 0 < P(X_i = 1) \leq 0.5 \\ &= \frac{2H_{max} - H(X_i)}{2} && if \;\; 0.5 < P(X_i = 1) \leq 1 \end{aligned} \tag{3}$$

The Fitness Function is calculated in formula (4)

$$fitness = \frac{Lx}{N} + XW^T$$

$$Where, W = \frac{w_i}{\sum_{i=1}^{N} w_i} \tag{4}$$

Where, W is the Weight vector assigned using (3) for R2L and U2R attacks and w_i is the weight of the i^{th} feature. Higher the fitness of a chromosome better is the chance of survival in the next generation. N is the total number of features, Lx is the number of 1s in the chromosome; X is the gene vector where the i^{th} gene corresponds to the i^{th} feature in the feature vector. The crossover probability simply indicates the ratio with which the parents are chosen for crossover and the mutation probability is the probability with which a particular gene is mutated. Based on trials, it was found that single point crossover with the pivot arbitrarily in the 25th position gives best results. Crossover probability is 1. Rank Selection is used for selecting the parents for crossover. Mutation probability is chosen to be 0.05. After certain number of generations, the algorithm converges to a solution. The feature selection algorithm is run separately for R2L and U2R attacks over 300 generations to find out the optimal feature subset.

The psuedocode for feature selection is given below. encode[] stores the encoded initial solutions obtained from the dataset. chromosome[] is a vector which stores the 38 bit chromosomes used by the GA. children[] is a vector which stores the offspring resulting from the crossover operation.

Algorithm 1 Feature Selection Pseudocode

```
Separate R2L and U2R attack instances
Set population_size_of_r2l=995, population_size_of_u2r=52
pop[i][j] ← i^th recordand j^th feature
Remove features 2,3,4 from Table 1
for i = 0 to pop.length − 1 do
    if valueof j^th feature > 0 then
        encode[i][j] ← 1
    else
        encode[i][j] ← 0
    end if
end for
for specific no. of generations do
    chromosome[i] ← encode[i][j]
    Calculate fitness[i] using Formula (4)
    Sort chromosome[i] based on fitness[i]
    Assign ranks to chromosome[i] in descending order
    Remove 10% of less fit chromosome[i]
    Copy 10% best chromosome[i] to next generation
    for all chromosome[i] as parent do
        Single Point Crossover on parents
            children[i] ← chromosome[i]
            children[i+1] ← chromosome[i+1]
        %Mutation
        for each bit do
            Generate random variable, 0 ≤ r ≤ 1
            if r ≤ 0.05 then
                flip bit
            else
                retain bit
            end if
        end for
    end for
end for
```

Table 4 Results of Feature Selection

Attack Type	Features selected	No. of features selected
R2L	1,2,3,4,5,6,10,12,23,24,25, 26,27,28,29,32,34,36,38,39,41	21
U2R	1,2,3,4,5,6,10,11,12,14,15,17, 18,19,23,24,29,31,32,33,34,36	22

The chromosome with highest fitness is the feature subset vector. In this vector, bit 1 indicates that a feature is selected, whereas bit 0 indicates that a feature is rejected.

4.2 Results of Feature selection

The proposed approach for feature selection combines the search technique and the evaluation function with the fitness function acting as the latter. The results of feature selection algorithm are summarized in Table 4.

5 Experimental Setup

The workflow diagram of the proposed classifier system is shown in Fig 2. It is a two-stage classifier with the first stage using the Simple Naive Bayes filter. The classifier is trained with all features selected and then the trained filter is evaluated against the test set. The output obtained from the first stage consisting of misclassified instances. i.e. only those records whose predicted class value is not the same as the actual class value, are given to the second stage. The classifier accuracy is calculated. In the second stage, the training set is trained only for the R2L attacks and U2R attacks separately. The Naive Bayes model filters the records into either two classes (R2L and others or U2R and others). In the second stage, the classifier is trained with the features selected by the Genetic Algorithm. The filter is evaluated against the misclassified records obtained from stage 1.

The experiments were run on a Windows platform PC with Intel Core i5-3210 Processor running at 2.50 GHz and 4GB RAM. The classifications were performed in WEKA (Waikato Environment for Knowledge Analysis) version 3.7. 12, a standard tool for running data mining tasks and machine learning algorithms. The following expressions are used to analyze the data:-

True Positives (TP): Percentage of correctly detected R2L and U2R records. (Actual class = Predicted class)

False Positive (FP): Percentage of records belonging to some other attack class that are classified as either R2L or U2R records. (False Alarm)

True Negative (TN): Percentage of records belonging to some other attack class that are correctly detected.

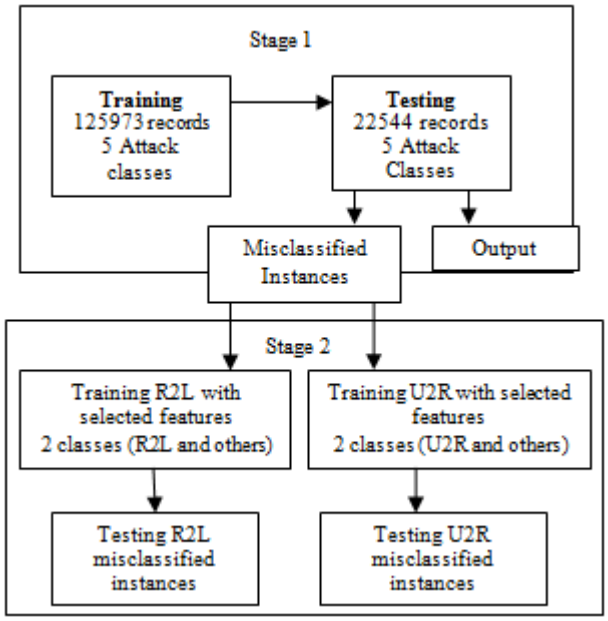

Fig. 2 Workflow of the proposed two-stage classifier

	Predicted Class	
Actual Class	True Positive (TP)	False Negative (FN)
	False Positive (FP)	True Negative (TN)

Fig. 3 A confusion matrix

False Negative (FN): Percentage of records belonging to R2L and U2R attack class that are not detected.

A confusion matrix visualizes the performance of a supervised learning algorithm displaying the true positives, true negatives, false positives and false negatives. This is shown in the Fig 3.

The Detection Accuracy Rate (DAR) refers to the total number of attacks records belonging to a particular class that are detected successfully. Parameters that used to measure the performance of the algorithm are Precision and Recall. Precision is the number of attack records detected correctly to the total number of records detected (TP + FP). High Precision means that the algorithm returned more relevant results than irrelevant for a particular class. Recall is the number of attack records detected correctly to the total number of records belonging to that class (TP + FN). High recall means that the algorithm returned most of the relevant results.

$$precision = \frac{TP}{TP+FP} recall = \frac{TP}{TP+FN} \quad (5)$$

$$DetectionAccuracyRate(DAR) = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

Table 5 Results of Stage 1 Classifier

Class	TP Rate(%)	FP Rate(%)	Precision (%)	Recall (%)
Normal	86.9	23.9	73.3	86.9
DoS	71.1	4.1	89.6	71.1
U2R	25.5	8.5	2.6	25.5
R2L	10.5	1.3	52.2	10.5
Probe	81.6	3.2	75.4	81.6
Weighted Average	71.2	12.2	74.7	71.2

6 Results

As seen in Table 1, the NB classifier has a reasonable accuracy compared to other [9] over the NSL-KDD dataset. For R2L attack, the precision is found to be 52.2% and a very low recall of 10.5% with 288 records of the total R2L records detected. The filter produces a very low precision of 2.6% for U2R attack records indicating that 97.4% of the records detected for this class were false positives. Recall for U2R records is found to be 25.5% (detects 51 out of 200 records).

As seen from Table 5, normal records have the high detection rate as well as very high false positive rate. The false positive rate of U2R attack is 8 times of R2L attack indicating that the NB classifier misclassifies higher percentage of U2R records. The FP rate of 1.3% is the lowest for R2L and significanly less than the mean FP rate of 12.2%. Comparing the recall of all attacks, it is seen that the recall of DoS and probe attacks is significanly higher than R2L and U2R attack. This is because the DoS and probing attacks involve many connections to the same host in a very short period of time and so the proper training of the temporal features will help in easily detecting these attacks. However, with R2L and U2R attacks, it is necessary to find sequential patterns in payload, which is a difficult task. It was found that the classifier misclassifies 6489 out of the total 22544 test records. Of these misclassified records, 149 were U2R attack instances and 2466 were R2L attack instances. Stage 2 results are tabulated in Table 6 and Table 7.

As seen in Table 6, 2086 R2L records were detected by the stage 2 classifier with an accuracy of 84.59%. Of the 149 misclassified U2R records, 143 were detected by the new algorithm with an accuracy of 95.97%. This shows that the training the selected features in the second stage helped in enhancing the capture of malicious records. Out of the 6 record connections that were not classified, 4 of them belonged to the buffer overflow attack, 1 belonged to the xterm attack and 1 belonged to the httptunnel attack.

The false positive rate is reduced to 1.3% for R2L attack and 2.1% for U2R attack in both stages. The proposed algorithm produced the lowest detection rate of 28% for snmpguess attack and 82.58% for snmpgetattack. This can be attributed to the fact that these attacks work on the UDP protocol, whereas all the attacks in the training set were

Table 6 Results of Stage 2 R2L Classifier

Class	TP Rate(%)	FP Rate(%)	Precision (%)	Recall (%)
R2L	84.6	0	1	84.6
Others	0	15.4	0	0
Correctly Classified Instances			2086	84.5904%
Incorrectly Classified Instances			380	15.4096%
Classifier Accuracy				84.59%

Table 7 Results of Stage 2 U2R Classifier

Class	TP Rate(%)	FP Rate(%)	Precision (%)	Recall (%)
U2R	95.97	0	1	95.97
Others	0	4.02	0	0
Correctly Classified Instances			143	95.97%
Incorrectly Classified Instances			6	4.02%
Classifier Accuracy				95.97%

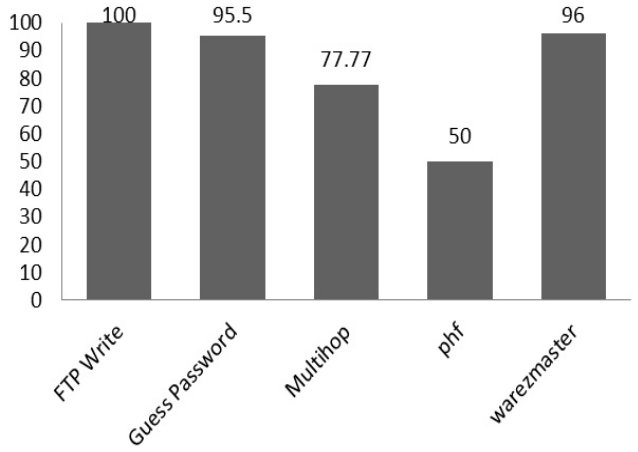

Fig. 4 Detection rate of different R2L attack types

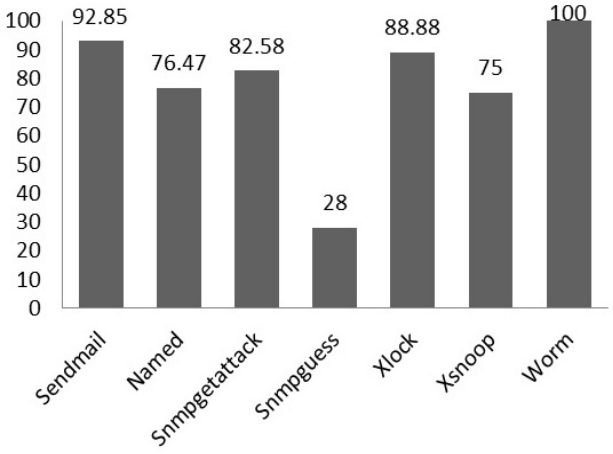

Fig. 5 Detection rate of novel R2L attack types

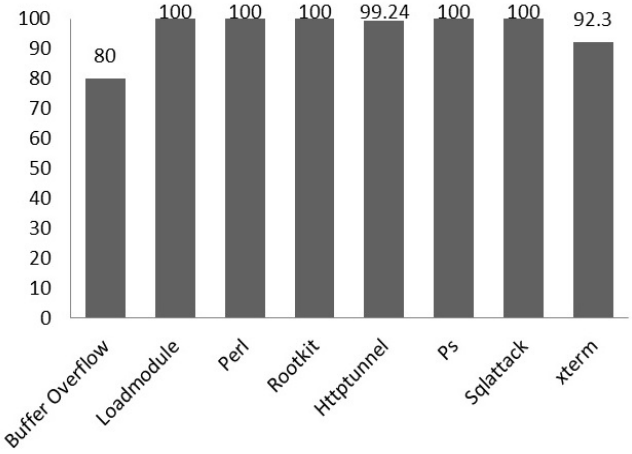

Fig. 6 Detection rate of different U2R attack types

based on the TCP protocol. As a result, the classifier was not trained properly to detect UDP based attacks and can be concluded as one of the flaws in the dataset construction. 1176 out of 1231 Guesspassword attack records and 907 out of 944 warezmaster records were detected. Except for these two attacks, the performance of the proposed approach novel R2L attacks is significantly improved compared to earlier approaches. The detection rates of different attacks can be shown in Fig. 4, Fig. 5 and Fig. 6. The overall detection rate of an attack is the ratio of the number of attacks instances detected in both stages to the total attack instances of that attack type.

The overall detection accuracy rate of the proposed approach for R2L attacks is obtained to be 86.2% and for U2R attacks it is obtained to be 91%. Comparing the results with those obtained by Ingre & Yadav in [15], it can be seen that Naive Bayes classifier provides better results in classifying instances of the NSL-KDD dataset as compared to Artificial Neural Networks. The proposed approach significantly improves the detection rate of R2L and U2R attacks as shown in Fig 7.

Table 8 Detection Rates of different classifiers in stage 2

Classifier	R2L records detected (out of 2466)	Detection Rate (%)	U2R records detected (out of 149)	Detection Rate (%)
Proposed Approach	2086	84.59	143	96
C4.5	45	1.82	7	4.7
Random Forest	307	12.44	2	1.3
Random Tree	12	0.486	6	4
Simple K-Means	876	36	149	100
NB-Tree	2	0.0811	6	4
Bayes Network	321	13.017	6	4
SVM [9]	64.3%			
SOM [11]	75.49%			

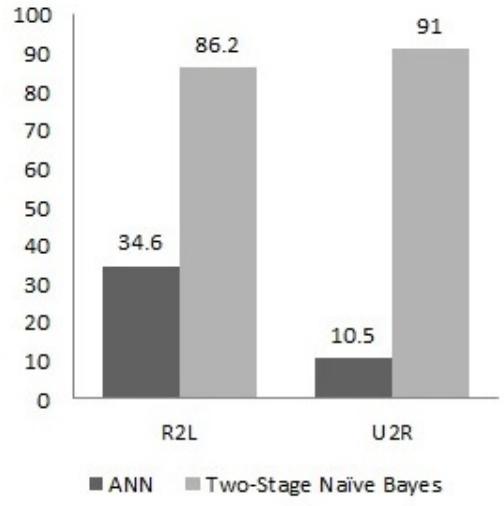

Fig. 7 Comparison of the detection rate of the proposed approach with ANN

The detection rate of different classifiers for stage 2 of testing is shown in Table 8.

7 Conclusion

In this paper, a two-stage Naive Bayes Classifier was proposed to improve the detection rates of R2L and U2R attacks. Feature selection was implemented in the second stage of the classifier using Genetic Algorithm to select the most important features for the two classes of attacks. The Genetic Algorithm (GA) based feature selection combines the subset search and the evaluation function to give an approximate feature subset representing R2L and U2R attacks separately. The fitness function in GA uses entropy based weights to assign importance to a feature. Run over 300 generations, a total of 21 important features were selected for the R2L attacks, whereas the feature relevance for U2R was 22. The NSL-KDD dataset was used to test the efficiency of the proposed algorithm. The NSL-KDD dataset removes the redundant records so that the classifiers are not biased towards the more frequent records.

The proposed algorithm significantly enhanced the result as compared to the previous IDS techniques. For R2L attacks, the accuracy was achieved to be 86.2%, whereas for U2R it was 91%. The false positive rate reduced to 1.3% and 2.1% for both respectively It is found that the detection rate is lowest for the snmpguess and the snmpgetattack since the classifier was not trained properly to detect UDP based attacks. The NB Classifier is simple in execution and achieves faster results as compared to other classifiers. As a result, it can be used in parallel and in multiple stages to improve efficiency, however increasing the complexity. This feature selection and classification process can be performed on DoS and probe attacks as per the need. The main limitation of the proposed work is its inefficiency with real life traffic. This deployment of a two-stage Naive Bayes classifier can be a further topic of research in the field of IDS.

References

1. Hung-Jen Liao, Chun-Hung Richard Lin et.al, Intrusion Detection System: A comprehensive review, Journal of Network and Computer Appllications, 36(2013), pg 16-24.
2. Monowar H. Bhuyan, D.K Bhattacharyya; J.K. Kalita, Network Anomaly Detection: Methods, Systems and Tools, IEEE Communications Surveys & Tutorials, Vol 16, No 1, First Quarter 2014
3. Hesham Altwaijry, Saeed Algarny, Bayesian based intrusion detection system, Journal of King Saud University Computer and Information Sciences (2012), 24, pg 1-6
4. Hesham Altwaijry, Saeed Algarny, Multi-Layer Bayesian Based Intrusion Detection System, Proceedings of the World Congress on Engineering and Computer Science 2011, Vol II, WCECS 2011, Oct 19-21, 2011, San Francisco, USA.
5. Yuteng Guo, Beizhan Wang Et.al, Feature Selection Based on Rough Set and Modified Genetic Algorithm for Intrusion Detection, 5th International Conference on Computer Science & Education, Hefei, China, August 24-27th ,2010, pg 1441-1446.
6. Hua Zhou, Xiangru Meng, Li Zhang, Application of Support Vector Machine and Genetic Algorithm to Network Intrusion Detection, Wireless Communications, Networking and Mobile Computing 2007, WiCom 2007, 21-25 Sept. 2007, pg 2267-2269.
7. Megha Agarwal, Performance Analysis of Different Feature Selection Methods in Intrusion Detection, International Journal of Scientific & Technology Research Volume 2, Issue 6, June 2013.
8. Datta H.Deshmukh, Tushar Ghorpade, Puja Padiya, Intrusion Detection System by Improved Preprocessing Methods and Nave Bayes Classifier using NSL-KDD 99 Dataset, 2014 International Conference on Electronics and Communication Systems (ICECS), 13-14th Feb 2014, pg 1-7
9. M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, A Detailed Analysis of the KDD CUP 99 Data Set, Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009.
10. J. McHugh, Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by Lincoln laboratory, ACM Transactions on Information and System Security, Vol. 3, no. 4, pp. 262-294, 2000.
11. Laheeb M. Ibrahim, Dujan T. Basheer, Mahmood S. Mahmood, A Comparison Study for Intrusion Database(KDD99, NSL-KDD) Based on Self-Organization Map (SOM) Artificial Neural Network, Journal of Engineering Science and Technology, Vol. 8, No. 1(2013) 107-119.
12. B.Senthilnayaki, K.Venkatalakshmi, A.Kannan, An Intelligent Intrusion Detection System Using Genetic based Feature Selection and Modified J48 Decision Tree Classifier, 2013 Fifth International Conference on Advanced Computing (ICoAC), 18-20th Dec 2013, pg 1-7.
13. Dheeraj Pal, Amrita Parashar, Improved Genetic Algorithm for Intrusion Detection System, 2014 Internationl Conference on Computational Intelligence and Communication Networks (CICN), 14-16th Nov. 2014, pg 835-839.
14. NSL-KDD Dataset [Online]. Available: http://nsl.cs.unb.ca/NSL-KDD/
15. Bhupendra Ingre, Anamika Yadav, Performance analysis of NSL-KDD dataset using ANN, International Conference on Signal Processing and Communication Engineering Systems (SPACES), 2-3rd Jan 2015, pg 92-96.