Collaborative Filtering and Random Forest classification Algorithm
For PROBE Attacks Detection in a Network

Madhuri Nallamothu, Mrs D.N.V.S.L.S.Indira
1 M.Tech (CSE), Gudlavalleru Engineering College, Gudlavalleru
2 Associate professor, Gudlavalleru Engineering College, Gudlavalleru.

Abstract: During the past few years huge amount of network attacks have been increased the requirement of efficient network intrusion detection techniques for detecting attacks. In the existing approach, different classification techniques are used for identifying various real time network attacks using data mining. However, most of the algorithms fail to classify the different types of attacks due to absence of collaborative filtering technique and robust classifiers. In Proposed System Robust collaborating filter Algorithm is an optimization method used for fine-tuning of the features whereas Random forest (RF), a highly accurate classifier, is designed here for Probe kinds of attacks classification.

Keywords— Random forest, self organizing map, intrusion detection, filtering, Normalization.

I. INTRODUCTION

With the tremendous growth of network-based services and sensitive information on networks, the number and the severity of network-based computer attacks have significantly increased. Completely preventing breaches of security is unrealistic by security technologies such as information encryption, access control, and intrusion prevention. Thus, Intrusion Detection Systems (IDSs) play a vital role in network Security Network Intrusion Detection Systems (NIDSs) detect attacks by observing various network activities, while Host-based Intrusion Detection Systems (HIDSs) detect intrusions in an individual host.

There are two major intrusion detection techniques: misuse detection and anomaly detection. Misuse detection determines intrusions by patterns or signatures which can represent attacks. Thus, misuse based systems can detect known attacks like virus detection systems, but they cannot detect unknown attacks since these detection systems are usually prone to higher detection rate and lower false positive rate than anomaly detection. Another disadvantage of misuse detection is high detection speed due to low complexity of detection algorithms. Anomaly detection usually has high computational complexity, especially for unsupervised algorithms such as clusters, outlier detection of the random forests algorithm, and Self-Organizing Map (SOM).

Therefore, misuse detection is more suitable for on-line detection than anomaly detection. There have been numerous techniques for modeling anomalous and normal actions for intrusion detection. The signature-based and supervised anomaly detections are commonly deployed and commercially accessible. The signature based detection ingredients attribute of the system information. It detects intrusions by comparing the feature values onto a group of attack signatures given by human experts. Unfortunately, it can simply identify previously known intrusions through a signature. The signature database has got to feel manually revised for each new kind of discovering attacks. Throughout the other hand, the supervised anomaly detection trains versions of labeled information (i.e., Information pre-classified as an attack or not) and checks how well the new data set into the model[1]. Definitely, it can not be promptly modified to new kinds of intrusion and do not have sufficient labeled information available. As a whole, a really large amount of system information should be handled and classified. Thus, it is impractical to classify them manually.

One of several challenges in IDSs is feature selection. Many algorithms are sensitive to the amount of attributes. Hence, feature selection is essential for improving recognition speed. The raw information format of system traffic is certainly not suitable for detection. IDSs should build features from raw system traffic data, and it involves a lot of computation. Thus, feature selection can help lower the computational price for feature construction by decreasing the sheer number of features. However, in a lot of current data-mining based IDSs, feature selection is based on domain knowledge or intuition. We make use of the feature selection algorithm that can give bids of what attributes are essential inside the classification. Another challenge of intrusion recognition is imbalanced intrusion. Some intrusions like denial of provider (DoS) [2] have much more connections than others (e.g., user to root). Nearly all of the information excavation algorithms try to minimize the overall error rate, but this leads to growing the error pace of minority intrusions. However, in real-world system environments, minority attacks are more risky than bulk attacks. In this paper, we improve the recognition performance for minority intrusions.

The KDD CUP’99 dataset that is produced by MIT Lincoln Lab under contract to Defense Advanced Research Projects Department (DARPA) is usually familiar with examining the performance of the IDS. There are 42 attributes and millions of connecting records within the dataset. Unfortunately, the tall dimensional feature area might include many redundant or sound attributes that could result in not only decreasing classification accuracy but additionally increasing training time and space complexity of the classifier. Hence, feature selection happens to be an effective way to choose the essential feature space which
probably can improve the standard of detecting attacks via classification.

II. RELATED WORK

Within a classification problem, the quantity of features can be very large, numerous of that are irrelevant or redundant. Since the amount of audit data that an IDS must examine is really big even for a small system, classification by hand is impossible. Feature reduction and feature selection improves classification by on the lookout for the subset of features, which best classifies the training information. Some of the important attributes an intrusion recognition program should have include refer in Srilatha et al. [3].

Most intrusions happen via the network using the network protocols to attack their targets. Twycross [4] suggested a new paradigm in immunology, Danger Principle, to feel used in developing an intrusion recognition program. Alves et al. [5] exhibits a classification-rule discovery algorithm integrating artificial resistant systems (AIS) and fuzzy methods[8]. For instance, throughout a certain intrusion, a hacker follows fixed methods to attain his attention, first sets up a connection between a source IP address on to a target IP, and sends information to attack the target[6,7,8]. Generally, there are four areas of attacks. They are: 1) DoS (denial-of-service), for illustration ping-of-death, teardrop, smurf, SYN flood, and the like. 2) R2L: unauthorized access from a remote machine, for example guessing account, 3) U2R : unauthorized access to the local super user (root) privileges, for instance, various "buffer overflow" attacks, 4) PROBING: surveillance along with other probing, for instance, port-scan, ping-sweep, etc. A few of the attacks (like DoS, and PROBING) might use hundreds of network packets or connections, while however attacks like U2R and R2L typically utilize sole one or maybe a few Connections.

III. OVERVIEW OF THE FRAMEWORK

The suggested framework applies collaborating filters along with robust classifies to identify the intrusions. The framework is shown in Figure 1. The NIDS catches the network traffic and constructs dataset by pre-processing. Thereafter, the random forest is designed to progress the service-based patterns. Proposed approach contains the collaborative filter onto the probe attacks after filtering is matched, the outcome is targeted against both classifiers then the results are compared with existing approach results. The system will effectively classify the probe kinds of attacks.

Dataset and Preprocessing:

The DARPA dataset is common to evaluate almost all of IDSs. The KDD’99 dataset is a subsection, subdivision, subgroup, subcategory, subclass associated generally DARPA dataset prepared caused by Sal Stofo and Wenke Lee. This dataset is a preprocessed dataset consisting of 41 features (e.g., Protocol , provider, and flag) extracted from the tcp dump information inside the 1998 DARPA dataset[10].

<table>
<thead>
<tr>
<th>Dst_host</th>
<th>Src_bytes</th>
<th>Protocol</th>
<th>Serv_error</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Tcp</td>
<td>0.3</td>
<td>Normal</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>Udp</td>
<td>0.5</td>
<td>Probe</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>Udp</td>
<td>0.3</td>
<td>Probe</td>
</tr>
<tr>
<td>0.3</td>
<td>0.3</td>
<td>Udp</td>
<td>0.5</td>
<td>Probe</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Tcp</td>
<td>0.9</td>
<td>Normal</td>
</tr>
<tr>
<td>.09</td>
<td>.09</td>
<td>Icmp</td>
<td>0.45</td>
<td>Probe</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>Udp</td>
<td>1.35</td>
<td>Normal</td>
</tr>
<tr>
<td>1.35</td>
<td>1.35</td>
<td>Udp</td>
<td>0.3</td>
<td>Normal</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>Tcp</td>
<td>.09</td>
<td>Probe</td>
</tr>
<tr>
<td>0.45</td>
<td>0.45</td>
<td>Icmp</td>
<td>0.5</td>
<td>Probe</td>
</tr>
<tr>
<td>0.3</td>
<td>0.45</td>
<td>Icmp</td>
<td>0.4</td>
<td>Probe</td>
</tr>
</tbody>
</table>

Table 1: Dataset
<table>
<thead>
<tr>
<th>Classification of Attacks</th>
<th>Attack Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Denial of Service</td>
<td>Neptune, Snarf, Pod, Teardrop, Land, Beck, Apache2, Udpstorm, Process-table, Mailbomb</td>
</tr>
<tr>
<td>Remote to User</td>
<td>Guesspassword, Flywirl, Knap, Pat, Multidop, Warnzeister, Warnseeker, Scangetattack, Named, Xlock, Xscan, Bendmail</td>
</tr>
<tr>
<td>User to Super User</td>
<td>Bufferoverflow, Loadmodule, Pod, Rootkit, Xworm, Ps, Hsp-tunnel, Sljattach, Worm, SnapGuess</td>
</tr>
<tr>
<td>Probing</td>
<td>Portswipe, IPswipe, Nmap, Satan, Saint, Macam</td>
</tr>
</tbody>
</table>

### IV. PROPOSED ALGORITHMS

The standard filtering algorithms are not adaptive to currently the conditions when kdd99 dataset is large. This might generate the false recommendations. In this paper, a new recommended collaborating filter can be used in order to get active probe attacks dynamically based on the network feature changing by using the dynamic likeness. It can detect the target probe type of attacks based on the network remaining features. The process for the proposed recommendation is as follows.

**Step1.** Determining the actual load W(u,i) This action is used to have the best attributes for probe kind of attacks detection using W(u,i)=best feature selection method(data) Step2. Using approved Pearson’s correlation to gauge the similarity Pearson’s correlation, as follows, measures the linear correlation between 2 vectors of ratings.

\[
\text{sim}(i, j) = \frac{\sum (R_{i,c} - A_i)(R_{j,c} - A_j)}{\sqrt{\sum (R_{i,c} - A_i)^2}(\sum (R_{j,c} - A_j)^2)}
\]

Where \( R_{i,c} \) is the rating of the probe kind of attack c by system protocol i, \( A_i \) is the average rating of network protocol i for all the co-rated system attributes, and \( R_{i,j} \) is the probe attack set both ratings by system protocol i and protocol j.

**Step3.** Neighbor Selection
A set of K probe attacks is found, which is formed according to the degree of similarity between each of the network attacks with the target probe attack.

**Step4.** Prediction
To generate a prediction on a probe attack rating prediction formula is used. Since we have got the probe attack features based on the protocol and size of the source bytes, we can calculate the weighted average of probe attack rating. The producing prediction formula as following:

\[
P_{ui} = A_u + \sum_{m=1}^{n} (R_{ui} - A_u)(\text{sim}(u, m))
\]

Ai is the average rating of network protocol i for all the co-rated network features, \( R_{ui} \): the rating of the probe attacks to the attack i, Am: average ratings of the probe attacks m to the protocols, \( \text{sim}(u, m) \): the similarity of the probe attack and the network attacks m, n: the number of the closeness of the attack similarity.

### Random Forests:

The random forests are an ensemble of unpruned classification or regression trees whose literature in relevance to intrusion detection. Random forest generates many classification trees. A tree classification algorithm is commonly employed to construct a tree with different bootstrap sample by means of original data. When the formation of forest is completed, a new object as well as to be classified is taken from each tree active in the forest. A vote is given by each tree which indicates the decision of the tree decision relating to the class of the particular object. The forest selects the class with the most votes just for the object.

A very important options that come with random forests algorithm are listed as follows:

1. It may be unsurpassable in accuracy one of the several current data mining algorithms.
2. It shows efficient performance on large data sets with lots of features.
3. It is able to a number of circumstances estimate of what features must be present.
4. It easily has no nominal data problem and doesn’t overfit.
5. It might possibly handle unbalanced data sets.

In random forests, there isn’t necessity for cross validation or any test set for an unbiased estimate of the coming test error. Since each tree is constructed by using the bootstrap sample, approximately one-third all over the total cases are omitted away from the bootstrap samples and they also try not to appear in ideal to start.

The working of Random Forests can be as follows:

1. Choose T number of trees to grow
2. Choose m wide range of variables used to split each node. \( m << M \), where M is the number of input variables.
3. Grow trees, while growing each tree do the following:
   (a) Construct a sample of size N from N training cases with replacement and grow a tree because of this new sample.
   (b) When growing a tree at each and every node select m variables at random from M and bust them out to have the best split.
   (c) Grow the tree to a new maximal extent. There isn’t really pruning.
4. To classify point X collect votes from every tree on the inside forest and then suddenly use majority voting available to you the class label.
Existing Binary pso:

Particle swarm optimizer is a population-based optimization algorithm using multiple candidate solutions to find the global optimum of a search space. It is inspired mainly by social behavior of flock organisms, such as swarms of birds or schools of fishes. The population is called a swarm and an individual is called a particle. A particle moves with an adaptive speed with an attempt to find the global optimum through cooperating and competing with other particles. When a specific particle finds the best solution, other particles move closer to it. Each particle represents a candidate solution to the optimization problem. They collaborate in an attempt to uncover ever-better solutions. Each particle in the swarm has two associated characteristics, a current position and a velocity. The position of a particle is influenced by the best position visited by itself (Pbest) and the position of the best particle in its neighborhood. When the neighborhood of a particle is the entire swarm, the best position in the neighborhood is referred to as the global best (gbest) particle. When smaller neighborhoods are used, the algorithm is generally referred to as a local (lbest) PSO.

V. EXPERIMENTAL RESULTS

Performance of our system is calculated on the basis of the number of trees constructed during the training phase. More the number of trees constructed more the amount of accuracy with only a small reduction in the performance. Figure 2 shows the comparison of Random Forests with other algorithm. As can be seen, with the increase in the number of trees used in the forest, the false positive rate decreases while determining attacks. Figure 4 shows the comparison of Random Forests algorithms with several different models. It shows that the execution times for different models do not vary very significantly. As the number of trees increases, the execution time for a given test set increases. This reduction in performance is negligible upon consideration of reduction in the rate of false positives.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Existing Algorithm</th>
<th>Proposed Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total no of Instances</td>
<td>5291</td>
<td>5291</td>
</tr>
<tr>
<td>Correctly Classified Instances</td>
<td>4962</td>
<td>5155</td>
</tr>
<tr>
<td>Incorrectly Classified Instances</td>
<td>329</td>
<td>136</td>
</tr>
<tr>
<td>Mean Error Rate</td>
<td>14.2857 %</td>
<td>16.3701 %</td>
</tr>
<tr>
<td>After Cross Validation</td>
<td>14.2857 %</td>
<td>16.8912 %</td>
</tr>
</tbody>
</table>

In the above figure 2, we compare the performance of Intrusion Detection System (IDS) Classifiers using feature reduction algorithm. As shown in the figure proposed approach using collaborating filter has only higher performance in probe type of attacks.
VI. CONCLUSION AND FUTURE WORK

In this paper we apply binary particle swarm optimization and Random forest methods to intrusion detection to avoid a hard definition between normal class and certain intrusion class and could be considered to be in more than one category. We introduce the current status of intrusion detection systems (IDS) and BPSO based feature selection heuristics, and present some possible data mining random forest technique for solving problems. BPSO based method with data reduction for network securities are discussed. As can be seen, with the increase in the number of trees used in the forest, the false positive rate decreases while determining attacks. The Collaborative filtering technique and random forests algorithm has been successfully applied to find patterns that are suitable for prediction in large volumes of data. Basically, in intrusion prediction, we can predict a specific intrusion based on symptoms. Improvements can be made on the collaborative filtering algorithm in the subsequent researches in order to make sure the precision of data source and improve the mining efficiency.

In future this work is extended to detect attacks in real time lan’s or wan’s. Future enhancement will concentrate more on the incremental collaborating filtering approach to multiclass dataset.

REFERENCES

[10] KDD’99 datasets, The UCI KDD Archive,