Chapter 3

Intrusion Detection System Using Bagging of Partial Decision Tree Base Classifier

Selection of appropriate technology and method for intrusion detection system is a challenging task. There are many methods which can be used to implement offline or inline intrusion detection system. Data mining and machine learning are mostly used in intrusion detection. Recently, soft computing techniques have been used which increase the accuracy and reduce the false positive rate. Although, such techniques are well advanced, there still arises a need to devise a proper technique and method to implement intrusion detection system. Now-a-days, the ensemble method of soft computing is widely used to implement intrusion detection system.

In this chapter, we have planned and implemented the intrusion detection system using the ensemble method of soft computing. We make use of bagging ensemble method to implement the system. The Partial Decision Tree has been used due to its simplicity and fast generation of rules. This experimental work has been implemented and tested offline. The offline test dataset has been used
to test the performance of the system. The Partial Decision trees are ensemble to increase the accuracy of the system. The system architecture and algorithm of the proposed IDS are explained throughout the chapter.

### 3.1. Introduction

Due to the rapid growth of the Internet, more and more systems encounter intruders in the network. These intruders can access, manipulate and disable computer systems through the Internet. The intrusion detection system is used to detect illegal access to the computer systems in a network. The malicious activities are detected by analysing the packets to prevent damage from the attack. Generally, intrusion detection techniques are categorized into two methods; anomaly and misuse detection. It is used to detect attacks based on the known pattern of attack. They are used to detect known attacks effectively with low errors. They are unable to detect unknown attacks because new attack does not have a similar pattern to the known attack. Anomaly detection technique is a profile based which analyses normal traffic. It effectively detects the unknown packets in the network, but they are not so effective in the detection rate. They also provide high false positive rates. To resolve the disadvantages of the anomaly detection technique of intrusion detection, soft computing techniques have been used by many researchers. The ensemble method of machine learning is more efficient, which can decrease the false alarms and raise the classification accuracy. There are three methods of ensemble methods, namely, Bagging, Boosting and Stacking [53] [54]. Boosting and Bagging ensemble methods are broadly used to implement the intrusion detection system as compared to Stacking. The stacking of weak classifiers requires more time, so they are practically not effective for intrusion detection.

The major contribution of this chapter is that, we have proposed a novel offline intrusion detection system using the bagging method of machine learning. Partial Decision Tree has used as weak classifiers to implement the
offline intrusion detection system. The selection of relevant features from the
dataset is essential to improve the classification accuracy and reduce the false
positives. The relevant features are selected based on their vitality to identify
the types of attacks. Genetic Search algorithm has used to select relevant
features from NSL_KDD training and test dataset. Finally, the performance of
the system is evaluated in term of false positive, classification accuracy and
model building time. Experimental result show that the proposed technique
for intrusion detection system outperforms as compared with earlier
approach.

3.2. Related works
Sengupta [55] had proposed IDS using modified Q-learning and RST. In this
work, author had reduced the dimensionality of training dataset to increase
accuracy with fewer discretized features. Hanging and Yu [56] had used
apriori algorithm which generate to recognize a variety of attacks, increase
the overall performance of detection system. Kim et.al, [57] had proposed
hybrid intrusion detection method. The method integrates the anomaly and
misuse detection in hierarchical manner. A misuse detection model is based
C4.5 classifier. The one-class SVM models are trained using subsets of training
dataset which reduce false positive rate effectively. Wang et.al, [58] had
proposed automatic intrusion detection system using dynamic clustering
method. It is online and adaptive intrusion detection system. Feng et al. [59]
had combined SVM method and clustering based on self-organised Ant colony
Network to implement the intrusion detection system. The proposed method
returns the benefits of both SVM and Clustering based Self-organized Ant
Colony Network which avoids their weakness. Kuanga et.al, [60] had
proposed a novel hybrid KPCA-SVM with Genetic algorithms model for
intrusion detection system. In this model, KPCA is used to extract the principal
feature of intrusion detection system. The SVM multilayer classifier is used to
identify an attack.
Earlier techniques used for intrusion detection system are based on combination of different family of machine learning classifiers [57] which uses C4.5 and SVM to develop a hybrid classifier which is more complex and it take require more time to build the model. The C4.5 requires complex optimization stage for forming rule set. The rules formed by C4.5 rule learner are complex and not effective for intrusion detection system. Hence to improve the accuracy, and model building time, and reduce false positive rate, we have offered a new approach for intrusion detection system using Partial Decision Tree. Bagging of similar classifier require less time as compared with heterogeneous classifier.

### 3.3. Partial Decision Tree as Base Classifier

There exists many rule learner algorithms of soft computing and machine learning to generate rules from decision trees. The C4.5 and RIPPER are two main schemes for rule learning. Both the schemes operate in two stages. The C4.5 first induces an initial rule set and then refines the rule set using complex optimization stage by discarding the individual rule. RIPPER does the same thing by adjusting individual rules. These two schemes can be combined to produce optimal rule sets. This combination of two schemes of rule learning is called as a PArtial Decision Tree (PART). This combined scheme does not require any complex optimization stage. The algorithm to combine C4.5 and RIPPER is very simple, effective and straightforward. Initially, it builds a pruned decision tree for the current set of instances. The leaf (best) with the largest coverage is converted into the rule, and the decision tree is discarded by removing cover instances from the training dataset. This process is repeated for all set of instances of the training dataset. This process is called separate-and-conquer strategy. PART algorithm produces rule sets which are more accurate than RIPPER’s rule set. PART’s rule sets are as accurate as C4.5’s rule set and the size of rule sets of PART is about of the same size of the C4.5 rule set. The performance of PART is fast because it does not need any post processing [62]. These features of partial decision tree enable us to implement
intrusion detection system. The intrusion detection is implemented using partial decision tree which has exhibited very exciting classification accuracy and false positive rate. The main drawback of this system is that it takes more time to generate the rule sets.

3.4. Architecture of the Proposed Intrusion Detection System

The work reported in [57] has implemented by using C4.5 and SVM which is more complex and it take more time to build model. The generated by C4.5 are complex and it requires more complex optimization stage. To overcome these problems, the bagging of very simple and fast rule leaner has applied to implement the intrusion detection system. The system architecture of the proposed IDS has reported in [63]. The proposed system architecture is depicted in Figure 3.1.

![System Architecture of the Proposed Intrusion Detection System](image)

In this work, we present two main contributions; one is to select the relevant features from NSL_KDD99 dataset and reduce the false positives. The Genetic
Algorithm has used to select the relevant features. Following subsection describe the feature selection process in detail.

### 3.4.1. Selection of Relevant Features

We provide a brief explanation of feature selection methods used in this contribution. The online available datasets provided by DARPA 1998, NSL-KDD99 and KDD99 are mostly used as a training dataset for intrusion detection system. In this contribution, the NSL-KDD99 dataset has been used to generate rules for intrusion detection application. Actually, the NSL-KDD dataset is a reduced version of KDDCup’99 dataset. This reduced version helps in reduction of time for rule generation. NSL_KDD dataset suggests 41 features for intrusion detection system. Feature selection (FS) is used to select a subset of features among the accessible features to obtain the improved result. Feature selection (FS) is needful because of following reasons.

- All features may not be useful and some features may be redundant
- Some features may cause confusion during learning and some features increase complexity of feature space
- Leads to more computation time for learning and finding solution

To avoid these things, the pre-processing of training dataset is essential before rule generation. Normally, a pre-processing step is required for reducing dimension, boosting generalization capability, accelerating learning and enhancing model interpretation [60]. In view of this, the Genetic Algorithm has been applied to NSL_KDD dataset. The Genetic Algorithm selects 15 relevant. Table 4.1 shows the list of features to train Partial Decision Tree (PART) and run out the experiments.

In the second contribution, the bagging ensemble method of machine learning has used to reduce the variance. The bagging method with Partial Decision tree as a base classifier is used to reduce the false positive and
increase the classification accuracy. On training, the rule set formed and stored in model. The performance of rule model is evaluated using cross-validation of 10-fold and test dataset. The test dataset is used to test the accuracy of the model. Each test samples are matched with rules in model. If the test sample is match with rule of abnormal rule, then it is declared as abnormal event or normal event. The following subsection describes the proposed ensemble algorithm for intrusion detection system.

<table>
<thead>
<tr>
<th>Sr. No</th>
<th>Relevant Features</th>
<th>Sr. No</th>
<th>Relevant Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Flag</td>
<td>9</td>
<td>Src_serro_rat</td>
</tr>
<tr>
<td>2</td>
<td>Src_bytes</td>
<td>10</td>
<td>Same_srv_rate</td>
</tr>
<tr>
<td>3</td>
<td>Dst_bytes</td>
<td>11</td>
<td>Diff_srv_rate</td>
</tr>
<tr>
<td>4</td>
<td>Wrong_fragment</td>
<td>12</td>
<td>Dst_host_count</td>
</tr>
<tr>
<td>5</td>
<td>Hot</td>
<td>13</td>
<td>Dst_host_srv_diff_host_rate</td>
</tr>
<tr>
<td>6</td>
<td>Logging_in</td>
<td>14</td>
<td>Dst_host_serror_rate</td>
</tr>
<tr>
<td>7</td>
<td>Num_file_creation</td>
<td>15</td>
<td>Dst_host_srv_serror_rate</td>
</tr>
<tr>
<td>8</td>
<td>Count</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.4.2. Algorithm of the Proposed Bagging for Intrusion Detection System

The bagging is a kind of voting algorithm which takes a base classifier and training set as input. It runs for multiple times by changing the distribution of instances in the training dataset. Each trained base classifier is then combined to generate a classifier that is used to classify the test dataset. Bagging is also called as Bootstrap Aggregating. In the voting method, classifiers are generated by different bootstrap samples Sm. The samples are generated by uniform sampling n instances from the training set with replacement. The classifiers C1; C2; C3;…; Cm are built using m boot-strap samples S1; S2; S3;…; Sm. The final classifier C* is built from the C1, C2, C3,…,Cm whose output the most often predicted by the base classifier. The basic procedure for the proposed intrusion detection system is summarized in Algorithm 1. The main reason for choosing PART is that it is simple, effective and straightforward decision tree.
Algorithm 1: Bagging of Partial Decision Tree for Intrusion Detection System.

**Input:** NSL_KDD dataset, 15 relevant features

**Start:**
1. Let m=number of bootstrap samples
2. For i =1 to m do
3. Create a bootstrap samples S1; S2; S3:Sm(Sample with Replacement)
4. Train Partial Decision Tree as a base classifier (Ci) on bootstrap samples Sm
5. End for
6. $C^* (x) = \arg\max \Sigma \delta (C_i (x) = y)$ (the most often predicted label y)

**End**

**Output:** Trained $C^*$ (Ensemble) classifier

### 3.5. Experimental Results and Discussion

In this section, the performance of bagging of Partial Decision Trees has been presented. All experiments are performed by using an Intel(R) CORE™ i5-3210M CPU @ 2.50GHz, Installed 8 GB RAM and 32-bit Operating system. Feature selection using Genetic Algorithm plays a vital role in constructing classification systems. It reduces the dimension of data, time to generate the rules and reduces the computational costs. Fifteen features are selected before passing the data sets to the bagger of decision trees (PART).

The performance of the proposed bagger and earlier approach presented in [57] is given in Table. 3.2. The proposed approach is also compared with other approaches for intrusion detection system. The performances of the different approaches are presented in Table 3.2 and Table 3.3 on cross-validation and on test dataset respectively.
Table 3.2
Performance Evaluation of the Proposed Ensemble on Cross-validation.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>RMSE</th>
<th>True Positive</th>
<th>False Positives</th>
<th>Classification Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naïve Bayes</td>
<td>0.3148</td>
<td>0.896</td>
<td>0.114</td>
<td>89.60</td>
</tr>
<tr>
<td>PART</td>
<td>0.054</td>
<td>0.997</td>
<td>0.003</td>
<td>99.66</td>
</tr>
<tr>
<td>C4.5</td>
<td>0.0517</td>
<td>0.997</td>
<td>0.003</td>
<td>99.69</td>
</tr>
<tr>
<td>Bagging (Naïve Bayes)</td>
<td>0.3112</td>
<td>0.895</td>
<td>0.114</td>
<td>89.48</td>
</tr>
<tr>
<td>C.4.5+SVM Ref. [58]</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>97.40</td>
</tr>
<tr>
<td>Proposed Ensemble</td>
<td>0.0477</td>
<td>0.997</td>
<td>0.003</td>
<td>99.71</td>
</tr>
</tbody>
</table>

Table 3.3
Performance of the Proposed Ensemble on Test Dataset.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>RMSE</th>
<th>True Positive</th>
<th>False Positives</th>
<th>Classification Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naïve Bayes</td>
<td>0.5072</td>
<td>0.74</td>
<td>0.212</td>
<td>73.97</td>
</tr>
<tr>
<td>PART</td>
<td>0.4664</td>
<td>0.778</td>
<td>0.172</td>
<td>77.79</td>
</tr>
<tr>
<td>C4.5</td>
<td>0.4534</td>
<td>0.791</td>
<td>0.166</td>
<td>78.08</td>
</tr>
<tr>
<td>Bagging (Naïve Bayes)</td>
<td>0.5068</td>
<td>0.74</td>
<td>0.212</td>
<td>73.97</td>
</tr>
<tr>
<td>C.4.5+SVM Ref. [58]</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>75.55</td>
</tr>
<tr>
<td>Proposed Ensemble</td>
<td>0.4418</td>
<td>0.784</td>
<td>0.159</td>
<td>79.87</td>
</tr>
</tbody>
</table>

According to Table 3.2 and Figure 3.2, bagging of Partial Decision Trees exhibits the better accuracy as compared with earlier approach [58] and other standard classifiers on both cross validation and test dataset. The proposed ensemble gives of 99.71% accuracy on cross-validation and 79.87% accuracy on the test dataset. In Figure 3.3, RMSE, True Positive, and False positive rates are given on test dataset. According to Figure 3.3, the false positive rate is very low as compared with other classifiers.
Intrusion Detection System Using Bagging of Partial Decision Tree Base Classifier

3.6. Chapter Summary and Conclusion

In this chapter, the bagging method of machine learning has been used to implement the offline intrusion detection system. The proposed approach uses Partial Decision tree as a base classifier for forming the rules for intrusion detection system. Genetic Algorithm is used to select relevant features from
NSL_KDD99 dataset which have reduced the model building time and have improved the performance of intrusion detection system. The experimental results show that bagging of Partial Decision Tree exhibits better classification accuracy over combination of C4.5 and SVMs presented in [57]. The main disadvantage of the proposed approach is that it generates more number of rules. The rules formed by Partial Decision trees are simple as compare with C4.5, still there is a scope to simplify the rules for intrusion detection system. In testing phase, it takes more time to match query tuple with all rules. Therefore, it is still difficult to deploy online. This motivates us to use simple and advance rule learner which reduce the number of rules, which is discussed in the next chapter.