Comparison of Machine Learning Algorithms and Their Ensembles for Botnet Detection

Songhui Ryu
Department of Computer and Information Technology, Purdue University, West Lafayette, IN 47907, USA
ryu26@purdue.edu

Botmaster

To compare accuracy of the algorithms, the following metrics are considered.

- Ensemble methods, which combine predictions of each ML classifier with or without weights, can be considered by making full use of
- Voting: As the simplest way, all different classifiers are trained separately with whole training data and its posterior probabilities are averaged.
- Boosting: To strengthen a classifier, Boosting incrementally builds an ensemble by training each model with the same dataset but weighted by error of the last prediction.

Accuracy Measurements

To compare accuracy of the algorithms, the following metrics are considered.

- **F1 score**: F1 score considers both the precision $\frac{TP}{TP + FP}$ and the recall $\frac{TP}{TP + FN}$ of the test. F1 score can be between 0 and 1 where 1 means its best value and 0 its worst.
- **Matthews correlation coefficient (MCC)**: With true and false positives and negatives, MCC can be between -1 and +1 where +1 means a perfect prediction, 0 no better than random prediction and -1 and inverse prediction.

## METHODOLOGY: ALGORITHMS & METRICS

### Machine Learning Algorithms & Ensemble Methods

The most common machine learning (ML) algorithms for classification are:
- Gaussian Naive Bayes (GNB),
- Neural Networks (NN), and
- Decision Tree (DT).

These all are supervised learning which need labeled data.

**Ensemble methods**, which combine predictions of each ML classifier with or without weights, can be distinguished as below.

- **Voting**: As the simplest way, all different classifiers are trained separately with whole training data and its posterior probabilities are averaged.
- **Boosting**: To strengthen a classifier, Boosting incrementally builds an ensemble by training each model with the same dataset but weighted by error of the last prediction.

## DATASET

The CTU-13 dataset is a public dataset featuring Botnet traffic mixed with normal and background traffic captured at the CTU university, Czech Republic in 2011. Among 13 different network traffic captures, #9, #10, and #11 were used.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#9</th>
<th>#10</th>
<th>#11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration(hrs)</td>
<td>5.18</td>
<td>4.75</td>
<td>0.28</td>
</tr>
<tr>
<td># Packets</td>
<td>115,415,321</td>
<td>90,389,782</td>
<td>6,337,202</td>
</tr>
<tr>
<td>Tot. size of Packets</td>
<td>94GB</td>
<td>73GB</td>
<td>5.2GB</td>
</tr>
<tr>
<td># NetFlows</td>
<td>2,753,885</td>
<td>1,309,792</td>
<td>107,252</td>
</tr>
<tr>
<td>Tot. size of Netflows</td>
<td>1.5GB</td>
<td>980MB</td>
<td>74MB</td>
</tr>
<tr>
<td># Botnet Flows</td>
<td>184,979</td>
<td>106,352</td>
<td>8,163</td>
</tr>
<tr>
<td># Normal Flows</td>
<td>43,340</td>
<td>15,847</td>
<td>2,718</td>
</tr>
<tr>
<td># Background Flows</td>
<td>2,525,565</td>
<td>1,187,592</td>
<td>96,369</td>
</tr>
<tr>
<td>Bot ( #Bots )</td>
<td>Neris(10)</td>
<td>Rbot(10)</td>
<td>Rbot(0)</td>
</tr>
<tr>
<td></td>
<td>Neris(10)</td>
<td>Rbot(10)</td>
<td>Rbot(0)</td>
</tr>
</tbody>
</table>

### Table 1. Data description for each dataset (Left)

<table>
<thead>
<tr>
<th>Feature</th>
<th>#9</th>
<th>#10</th>
<th>#11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Duration</td>
<td>Flow start time</td>
<td>Protocol</td>
<td>SInd Port</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 2. Features in each dataset (Right)

## RESULTS

### Data Preparation

Among 12 features, Date and time, IP address and port number and the number of flows were excluded because Date was all the same, and there were very specific IP addresses that were set to Bots. Each test was done by splitting the dataset randomly into training and test set in ratio of 8:2. The following is the averaged values of 5 runs for each algorithm on a single machine with 64GB of memory.

### Results

#### Accuracy scores of each algorithms and ensemble method

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>F1</th>
<th>MCC</th>
<th>GNB</th>
<th>NN</th>
<th>DT</th>
</tr>
</thead>
<tbody>
<tr>
<td>GNB</td>
<td>0.945</td>
<td>0.847</td>
<td>0.857</td>
<td>0.873</td>
<td>0.873</td>
</tr>
<tr>
<td>NN</td>
<td>0.925</td>
<td>0.824</td>
<td>0.834</td>
<td>0.843</td>
<td>0.843</td>
</tr>
<tr>
<td>DT</td>
<td>0.915</td>
<td>0.814</td>
<td>0.824</td>
<td>0.833</td>
<td>0.833</td>
</tr>
</tbody>
</table>

### Individual algorithms

- **DT >= NN > GNB in accuracy**
- **NN > DT > GNB in time consumption**

### Voting

- Better than GNB and NN, but worse than DT
- This is because combining all three algorithms works by averaging the predictions

### Boosting

- **Adaboosting does not help either GNB or DT.**
- Boosting works combining multiple ‘weak’ classifiers. DT is already strong enough.
- It takes much more time than sole algorithm and lowers the scores for GNB.

### Bagging

- **Bagging each algorithm seems very similar to using a single classifier.**
- Bagging DT does not provide considerably better results for each algorithm alone.
- Bagging works by sampling training data to iteratively train the model.
- But data itself is very sparse.

### Table 2. Interpretation of the results

## CONCLUSION

### Findings

- Decision tree without any ensemble method would the most preferable.
- Taking ensemble methods in a hope of enhancing the accuracy of machine learning algorithms for Botnet detection is not likely to be a help to the actual detection.
- When a real-time detection system is considered, taking GNB or DT without any ensemble methods could be a good option.

### Future works

- **Data dependency**: The accuracy results are much lower when it comes to the huge dataset. Possible reason can be overfitting or the very nature of the different bots (Neris & Rbot). Also, Boosting GNB shows near zero of MCC which means no better than random prediction. The reason for this is also to be studied.
- **Other algorithms / dataset can be evaluated**: SVM, k-NN or Random Forest algorithms can be tested to see if they are appropriate for botnet detection. Also, more network attribute can be considered by making full use of NetFlow features.
- **Scalability**: Spark MLlib also provides classification algorithms. But ensemble methods arenot developed yet. Finding out if it is feasible to implement the ensemble methods with Spark MLlib can be considered.