Comparison of Machine Learning Algorithms and Their Ensembles for Botnet Detection

Songhui Ryu
Dept. of Computer and Information Technology,
Purdue University
ryu26@purdue.edu

ABSTRACT

A Botnet is a network of compromised devices that is controlled by malicious 'botmaster' in order to perform various tasks, such as executing DoS attack, sending SPAM and obtaining personal data etc. As botmasters generate network traffic while communicating with their bots, analyzing network traffic to detect Botnet traffic can be a promising feature of Intrusion Detection System(IDS). Although IDS has been applying various machine learning (ML) techniques, comparison of ML algorithms including their ensembles on Botnet detection has not been figured out yet. In this study, not only the three most popular classification ML algorithms – Naïve Bayes, Decision tree, and Neural network are evaluated, but also the ensemble methods known to strengthen ML algorithms are tested to see if they indeed provide enhanced predictions on Botnet detection. This evaluation is conducted with CTU-13 public dataset, measuring running time of each ML and its f measure and MCC score.

1 INTRODUCTION

According to the explanation in [1], a botnet is a "network of compromised devices used by a botnet owner to perform various malicious tasks. The devices – often personal computers – called 'Bots' are under the remote control of a human operator called 'Botmaster'." The primary goals of botnets include sending SPAM, executing DoS attack, obtaining identity etc. [2].

A botmaster's communication with the botnet is carried out via Command and Control (C&C), which is the main feature that distinguishes botnets from other malwares. Most botnets tend to use IRC commands for their C&C communication, but some botnets use POP3, HTTP or P2P as well. In this regard, botnet detection can be done by observing network traffics generated when botnets communicating with their C&C servers [3].

To analyze network traffic in IDS, using one or more machine learning(ML) algorithms have been popular. However, comparisons of each machine learning algorithms including ensemble methods for botnet detection has not been studied yet.

In this study, the three most common ML algorithms for classification [4] - Naïve Bayes, decision tree, and neural network are evaluated using public botnet dataset, CTU-13. Furthermore, ensemble methods which are known to strengthen machine learning algorithm also be evaluated to see if they indeed do the help to botnet detection.

2 METHODOLOGY

2.1 Algorithms

Naïve Bayes (NB) classifier calculates the probability of each target class and selects the highest one based on Bayes' theorem. This is popularly used for SPAM filtering because it is relatively fast and easy to understand.

Decision tree (DT) where leaves represent class labels and branches do conjunctions of features that lead to those labels, it supports the decision with chances of target class and resource costs obtained from decision rules.

Neural networks (NN) consisting of several layers, once each feature is given to input layer per a record, the model calculates weights of hidden nodes to get the highest value in the output layer. The outcome is compared to the true value and its error term is used to adjust the weights in the hidden layer.

2.2 Ensemble methods

Known to strengthen ML algorithms combining predictors of each ML classifier with or without weights, ensemble methods can be categorized into three.

Voting. All different classifiers are trained separately with whole training data and its posterior probabilities are averaged.

Bagging. Multiple sub-datasets from the original dataset are sampled and fed to each classifier to use voting at the end.

Boosting. Each classifier is incrementally trained with the same dataset but weighted by an error of the last prediction.

2.3 Metrics

F1 score. Considering both the precision p and the recall r of the test, F1 score can be between 0 and 1 where 1 means its best value and 0 its worst.

$$F1 = 2 * \frac{p * r}{p + r} \quad where\ p = \frac{TP}{TP + FP},\ r = \frac{TP}{TP + FN}$$

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.

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

3 TEST AND RESULTS

3.1 Dataset and Test Environment

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

[5]. The characteristics of the datasets are explained in Table 1. Also, among 12 attributes of network flow, 6 attributes – Duration, protocol, flags, type of service, number of packets, number of bytes were used as features. Date & time values and IP addresses were excluded.

Table 1: *Network flow distribution of the dataset*

	#9	#10	#11
Total flows	2,753,885	1,309,792	107,251
Botnet flows	184,979 (6.68%)	106,352 (8.11%)	8,163 (7.60%)
Normal flows	43,340 (1.57%)	15,847 (1.2%)	2,718 (2.34%)
Background flows	2,525,565 (91.7%)	1,187,592 (90.67%)	96,369 (89.85%)
Total size of NetFlows	1.5GB	980MB	74MB
Total size of packets	94GB	73GB	5.2GB
Bot(# bots)	Neris(10)	Rbot(10)	Rbot(3)

The machine used for the estimation consists of Two 10-core Intel Xeon-E5 and 64GB of memory.

3.3 Results

From the test that uses machine learning algorithms only from Scikit-learn, the elapsed time and accuracy scores are as follows. Outcomes from 5 different tests were averaged.

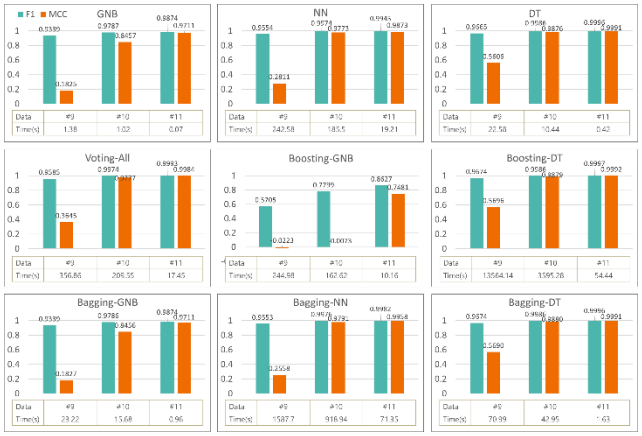

Figure 1: *Accuracy scores of each algorithm and ensembles implemented.*

For every data and algorithms, F1 score is higher than MCC score. This is because F1 score does not consider the true negatives. For this reason, MCC is preferred for a binary classification [7].

Among individual algorithms, for #10 and #11, NN and DT show decent accuracies over 0.97 in MCC even though NN takes about 20 times more time than DT. But for #9, the accuracy score dramatically goes down. Because the only differences among three datasets and their evaluation are that #9 is much larger than the others and it has different bot from the others. Possible reasons for this can be overfitting or the very nature of the different bots.

This pattern appears the same on Voting-All result. This is on the ground that voting works by averaging out each outcome from the model.

Boosting method does not help either GNB or DT. Because the nature of boosting is turning weak models, which has slightly better prediction than random, into a strong one [6]. In this regard, it obviously does not make DT strong as it already had a good accuracy. The interesting thing comes with boosting-GNB. For #9 and #10, the MCC scores are near zero which means the prediction is no better than random. At this moment, this outcome is put to the future work with the other problem following chapter.

Bagging each algorithm seems very similar to using a single classifier only for each dataset. While training a bagging model, multiple sub-datasets sampled out from the original dataset make their own classifier and then predictions from those classifiers are voted. This dataset, however, may not take benefit from sampling because the data is too sparse/skewed.

4 CONCLUSION

To detect botnet traffic out of all network traffic, decision tree without any ensemble method would be the most reliable approach. It runs much faster than NN alone, with the same accuracy. GNB even though it runs the fastest but the accuracy varies on the dataset. Unlike the common expectation, adopting ensemble methods on ML algorithms for botnet detection in a hope of enhancing the accuracy is not preferable because it does not give significantly more accurate result while taking much more time.

The questions that this evaluation gives are that why the accuracy scores drop down when a huge data used. This may lie in the nature of datasets or there might be overfitting issue. Evaluating other botnet network data to see if the prediction is data-dependent or changing parameters for the model can be a start point of future works. Also, the reason why boosting-GNB shows the poor result also could be handled together.

REFERENCES

[1] Sebastian Garc´ıa, Alejandro Zunino, and Marcelo Campo. 2014. Survey on network-based botnet detection methods. Security and Communication Networks 7, 5 (2014), 878–903.

[2] Julian B Grizzard, Vikram Sharma, Chris Nunnery, Brent ByungHoon Kang, and David Dagon. 2007. Peer-to-Peer Botnets: Overview and Case Study. HotBots 7 (2007), 1–1.

[3] Daniel Spokoyny. 2006. Taxonomy of botnet threats. (2006). http://www.cs.ucsb.edu

[4] Chakchai So-In, Nutakarn Mongkonchai, Phet Aimtongkham, Kasidit Wijitsopon, and Kanokmon Rujirakul. 2014. An evaluation of data mining classification models for network intrusion detection. In Digital Information and Communication Technology and it's Applications (DICTAP), 2014 Fourth International Conference on. IEEE, 90–94.

[5] Sebastian Garcia, Martin Grill, Jan Stiborek, and Alejandro Zunino. 2014. An empirical comparison of botnet detection methods. computers & security 45 (2014), 100–123.

[6] Breiman, L. (1998). Arcing classifier (with discussion and a rejoinder by the author). Annual Statistical Supplement ... to the Social Security Bulletin, 26(3), 801-849.

[7] Powers, D.M.W. (2011). Evaluation: from precision, recall and f-measure to ROC, informedness, markedness & correlation, Journal of Machine Learning Technologeis, 2(1), 37-63.