Towards using Transfer Learning for Botnet Detection

Basil Alothman
Faculty of Technology
De Montfort University
Leicester LE1 9BH, Great Britain
p14029266@my365.dmu.ac.uk

Prapa Rattadilok
Faculty of Technology
De Montfort University
Leicester LE1 9BH, Great Britain

Abstract—**Botnet Detection has been an active research area over the last decades. Researchers have been working hard to develop effective techniques to detect Botnets. From reviewing existing approaches it can be noticed that many of them target specific Botnets. Also, many approaches try to identify any Botnet activity by analysing network traffic. They achieve this by concatenating existing Botnet datasets to obtain larger datasets, building predictive models using these datasets and then employing these models to predict whether network traffic is safe or harmful. The problem with the first approaches is that data is usually scarce and costly to obtain. By using small amounts of data, the quality of predictive models will always be questionable. On the other hand, the problem with the second approaches is that it is not always correct to concatenate datasets containing network traffic from different Botnets. Datasets can have different distributions which means they can downgrade the predictive performance of machine learning models. Our idea is instead of concatenating datasets, we propose using transfer learning approaches to carefully decide what data to use. Our hypothesis is "Predictive Performance can be improved by using transfer learning techniques across datasets containing network traffic from different Botnets".**

Keywords-component; Botnet-Detection; Transfer-Learning; data-distribution; improve-predictive-performance; network-traffic-analysis)

I. Introduction

Traditional machine learning algorithms use datasets separately to create predictive models. In transfer learning, which is a subfield of machine learning, a group of datasets are used together to enhance the quality of such predictive models [1]. In more detail, transfer learning attempts to learn one or more tasks (known as the source tasks) and use the knowledge learnt to improve learning in another task (known as the target task). The source and target tasks are usually related.

We can summarise our contributions in the following two points: 1) We demonstrate that the distribution of data that belong to different Botnet types is different, which means concatenating such data without care is not always the right decision. 2) We demonstrate that using Transfer Learning, instead of traditional machine learning, can enhance the performance of predictive models, which leads to more accurate Botnet detection.

II. The Data

A. Obtaining the Data

We have downloaded the data that was used in [2]. The data is in Packet Capture (PCAP) format. Details of this data can found in [3]. We have mainly worked with the Testing Dataset because it has more Botnets than the Training Dataset. Because the data is in PCAP format, we needed to transform it into a format that machine learning platforms such as WEKA [7] or SciKit Learn [8] understand (i.e. into Character Separated Values, or CSV, format). Therefore, we have downloaded and used FlowGenerator [4] which reads in a directory that contains one or more PCAP files and transforms them into CSV files. It generates several attributes (features) such as Source Port, Destination Port, Protocol, Flow Duration, Flow Bytes per second and Flow Packets per second. The original number of features generated by FlowGenerator is 26 and the total number of instances we obtained is ~309000.

B. Labelling the Data

After obtaining the CSV file, we have labelled the data using the source and destination IP address fields as explained in [3]. The distribution of the Botnet and Normal traffic in the data varies. For example, the number of instances that belong to Botnet Osx_trojan was as little as 28. Also, the number of instances that belong to Botnet Weasel Botmaster was 40. On the other hand, the number of instances that belong to Botnet Virut was 42254 and the number of instances that belong to Botnet Neris was 24071. Furthermore, the number of instances that belong to Normal traffic was 149727.

C. Data PreProcessing

All features generated by FlowGenerator are Numeric which makes it easy to process data. The fields Source Port, Destination Port and Protocol are essentially categorical so we have used the one hot encoding technique to transform these into Dummy Variables. This has resulted in a very large dataset with > 60000 features.

D. Outlier Removal

In order to obtain clean data, we have used the One-class SVM [9] method for outlier and novelty detection on the large

978-1-908320/93/3/$31.00©2017 IEEE

dataset we obtained from the previous step. We have removed all the instances that were deemed Outliers.

III. Principal Component Analysis (PCA)

In data exploration and analysis, Principal Components Analysis [5] is a technique used to identify a smaller number of uncorrelated features (i.e. attributes or variables). These uncorrelated variables are usually known as the "principal components". Its main objective is to explain the highest possible amount of variance with the smallest possible number of principal components. It is commonly used as a dimensionality reduction procedure as well as an exploratory procedure to examine whether there is separation amongst instances that belong to different classes.

We have Run the PCA algorithm on the dataset we obtained after performing all the preprocessing steps we explained. Fig. 1 shows a scatter plot of the first and second principal components. To preserve display space, we only show points (i.e. instances) that belong to Botnets: TBot, Zero access and Zeus.

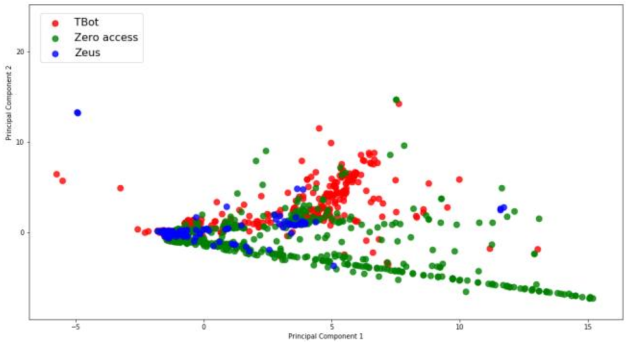

Figure 1: PCA Plot

It can be seen in the Fig.1 that there is separation between instances that belong to different Botnet types. Instances belonging to the same Botnet type tend to form a cluster and this solidifies the argument of this work.

IV. Transfer Learning Plan: Implementation and Evaluation

Since transfer learning uses related but separate datasets, we have split the dataset resulting from the steps explained above into smaller datasets. The split is based on the Botnet label. This means the split resulted in a separate dataset for each Botnet. We have split the Normal data (i.e. instances that belong to Normal traffic) into non-overlapping chunks and added one chunk to each Botnet dataset so that we have positive and negative examples in each dataset (we made sure that the classes in the resulting datasets are balanced). We are currently evaluating the performance of several machine learning models on these datasets and we will choose the best classifier to use in our proposed Transfer Learning experiments. For transfer learning itself, we will use the TrAdaBoost algorithm that can be found in [6]. TrAdaBoost is based on the classical AdaBoost algorithm. It works when the source and target tasks have the same set of features, but different data distributions. In addition, TrAdaBoost assumes that some of the data in the source task can be useful (i.e. leads to positive transfer) and some can be harmful (i.e. leads to negative transfer). The idea is to assign weights to data from the source task in such a way that useful data can have more effect than harmful data. The author has made the java implementation of this approach publicly available. As the java source code of TrAdaBoost is freely available on the internet, we have already downloaded and integrated it into WEKA [7]. We are currently setting up the environment for running experiments with relatively large Botnet datasets as source tasks and smaller Botnet datasets as target tasks.

V. Conclusion

Datasets that contain network traffic data belonging to different types of Botnets should not always be concatenated. In this work we have demonstrated that such data can have different distribution. Therefore, our suggestion to use transfer learning, instead of traditional machine learning, seems to be a reasonable method to enhance the performance of models used for Botnet identification and detection.

References

[1] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Trans. Knowl. Data Eng., vol. 22, no. 10, pp. 1345–1359, 2010.

[2] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," Computers & Security, vol. 31, no. 3, pp. 357–374, 2012.

[3] Beigi, Elaheh Biglar, et al. "Towards effective feature selection in machine learning-based botnet detection approaches." Communications and Network Security (CNS), 2014 IEEE Conference on. IEEE, 2014.

[4] ISCX/ISCXFlowMeter: ISCX Flow Meter. [ONLINE] Available at: https://github.com/ISCX/ISCXFlowMeter [Accessed on 12May 2017].

[5] I. T. Jolliffe, (2002), Principal Component Analysis (Springer Series in Statistics) Hardcover.

[6] W. Dai, Q. Yang, G. Xue, and Y. Yu, "Boosting for Transfer Learning," Proc. 24th Int'l Conf. Machine Learning, pp. 193-200, June 2007.

[7] Eibe Frank, Mark A. Hall, and Ian H. Witten (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.

[8] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[9] Larry M. Manevitz and Malik Yousef. 2002. One-class svms for document classification. J. Mach. Learn. Res. 2 (March 2002), 139-154.