ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

Extreme Learning Machines for Internet Traffic Classification

Joseph Ghafari[1,2], Emmanuel Herbert[1], Stephane Senecal[1], Daniel Migault[1], Stanislas Francfort[1] and Ting Liu[1]

1- Orange Labs
38-40 rue du General Leclerc, 92130 Issy-les-Moulineaux - France

2- Ecole des Mines de Nantes - GIPAD Dept.
4 rue Alfred Kastler, 44300 Nantes - France

Abstract. Network packet transport services (namely the Internet) are subject to significant security issues. This paper aims to apply Machine Learning methods based on Neural Networks (Extreme Learning Machines or ELM) to analyze the Internet traffic in order to detect specific malicious activities. This is performed by classifying traffic for a key service run over the internet: the Domain Name System (DNS). The ELM models and algorithms are run on DNS traffic data extracted from operating networks for botnet detection.

1 Introduction

Nowadays telecommunication devices, and especially computers connected to the Internet, are likely to be the target of malicious activities. According to [1], 5% to 10% of connected devices are infected with some form of malware. "Botnets" are one of the most common network security threats. A botnet is a network of infected computers that are remotely under the control of a hacker. Botnets are mostly used for illegal activities that require distributed resources. 27% of all malicious connection attempts can be attributed to botnet-related activities, cf. [2]. In order to put up a fight against botnets, several approaches have been considered. One of them consists in mining traces of DNS (Domain Name System [3]) traffic data in order to detect botnet activities. This solution has been implemented by Exposure [4], a system that employs large-scale, passive DNS analysis techniques for the detection, as well as in the Notos system [5]. These approaches use different properties of DNS names and the ways they are queried. Exposure classification methodology for instance is based on decision trees [6]. Our work differs from these existing solutions in the sense that we base our classification methodology on Artificial Neural Network models and that we consider different network features.

This paper is organized as follows. In section 2 we introduce the Internet DNS traffic dataset used to perform the detection. Then, in section 3, the structure and basic principles of the Extreme Learning Machines (ELM) method are presented from a high-level point of view. An ELM-based algorithm is thus run on the dataset for the task of botnet detection and the results are presented in section 4. Section 5 concludes the paper.

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

2 DNS traffic data

In this section, we describe the Internet DNS traffic data considered to conduct the botnet detection task. The DNS maps a name (like www.google.com) and an IP address. The name is also referred to as Fully Qualified Domain Name (FQDN). Practically it works as follows: an end user queries a DNS server, the DNS server performs the resolution and returns the mapping to the end user. The end user is then able to locate and contact the Internet resource wanted. Our dataset is extracted from raw traffic captures on a DNS resolving platform acting as a DNS server for the Internet Service Provider Orange. This dataset is a subset of a day-long traffic capture of about 650 millions of DNS queries.

The network traffic is captured from the DNS server using a packet capture (pcap) library and stored as .pcap files. A typical DNS packet is divided into several features, among which we consider: the IP source of the DNS query, an ID given to the query, a code representing errors that have occurred in the process, the FQDN associated with the query, the TTL (i.e. the "Time To Live") which represents the time, in seconds, for which the answer is considered valid and kept in cache, and the IP address sent as the answer to the query. In addition to these parameters, we consider the time at which events such as querying, answering and resolution take place, also called "Timestamps".

The dataset has been constructed from features mentioned above in three main steps: first, parameters are extracted from each DNS packet. Results are then grouped and processed to build a table with the FQDN in the first column and the other parameters in the other columns. Secondly, these FQDN are labelled as "Black" and "White" using domain name lists. Namely, Black lists such as Abuse.ch (Zeustracker and SpyEye), Malware Domains and Compuweb, and the White list dmoz, cf. [7]. The first 12 hours of the 24 hour long traffic capture is split into 12 equal parts of 1 hour each and only the first 15 minutes of each part is kept. Finally, the dataset, cf. table 1, is then filtered using the Black list and White list mentioned above. Because of the strong imbalance nature of the classification problem, the majority class (White) is downsampled. All the black-labelled entries are kept, and the white-labelled entries are thus downsampled to the size of the black-labelled list. The **qt**, **id**, **ans**, **eu**, **rt** and **ttl** parameters, cf. table 1, are statistical values (number of occurences, minimum, maximum and quartiles) representing the distributions of the lists of end user queries.

3 Extreme Learning Machines

The Multi-Layer Perceptron (MLP) model is widely used in Machine Learning. However, this approach raises issues such as its relatively slow learning speed and the relatively high number of parameters needed to be tuned. The Extreme Learning Machines (ELM), a recent learning paradigm, has been developed by

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

Name	Description
hour	indicates at which hour of the day the fqdn queries were issued
total_queries	total number of queries issued during the considered period
query time (qt)	time at which the query has been issued to the DNS server
id	ID of the query
answer (ans)	IP address given by the DNS server for the FQDN
end user (eu)	IP address of the end user issuing the query
response time (rt)	time at which the answer has been given by the DNS server
time to live (ttl)	the time to live of the answer
no_error	number of times the resolution was successful
error	number of times an error has occurred during resolution

Table 1: Dataset.

Huang and coworkers (cf. [8]) in order to deal with these issues. The main concept behind ELM lies in the random initialization of the incoming weights and biases of the hidden layer. The structure of an ELM consists in a classic feedforward layered network with only one hidden layer. This architecture is called "Single Layer Feedforward Network" (SLFN, as depicted in figure 1). The ELM model relies on the universal approximation capabilities of feedforward neural networks (cf. [6]). Furthermore, Huang and Babri showed in [9] that, with N distinct samples, the SLFN with a number of hidden nodes $\tilde{N} < N$ and almost any activation function $f(.)$ can learn a model with an arbitrarily small error. Based on this result, an extremely efficient learning algorithm has been developed which makes fast learning possible.

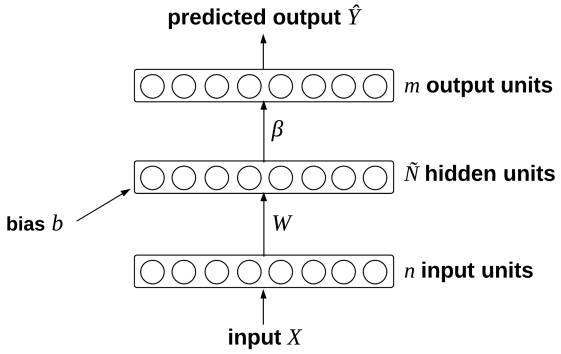


Fig. 1: Extreme Learning Machines' SLFN Architecture.

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

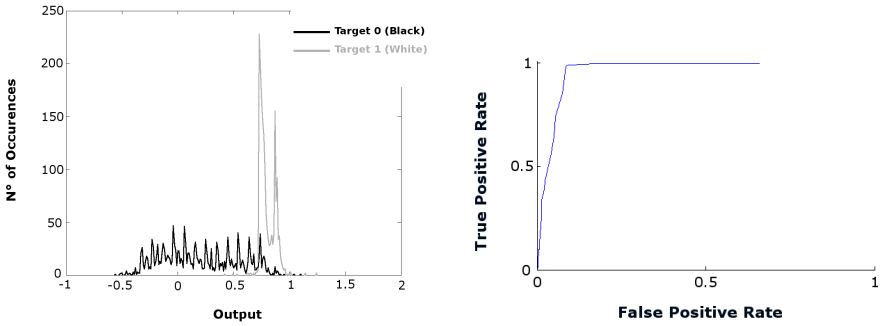
The training procedure for ELM can be decomposed in three steps: **(i)** random generation, **(ii)** propagation, **(iii)** inversion. The first step **(i)** consists in randomly generating the incoming weight matrix W and the bias vector b of the hidden layer (cf. figure 1). It was shown that the choice of the probability distribution for this random generation does not affect the overall performance of the network after training. Generating values from a uniform distribution has the advantage of being fast, simple and implemented in almost every programming language.
The second step **(ii)** requires to propagate the input up to the hidden layer to obtain the hidden layer output matrix H.
The third step **(iii)** consists in determining β as depicted in figure 1 so that it minimizes the error between the predicted output $\hat{Y} = H\beta$ and the target or expected output T.
This error or cost function is defined as $E = \|\hat{Y} - T\| = \|H\beta - T\|$.
β should be a solution of $H\beta = T$. In practice, zero error cannot be obtained. However, the aim is to get the error as close to zero as possible. Thus, the optimal weight matrix $\hat{\beta}$ should be a least-squares solution to $H\beta = T$. Furthermore, to ensure better generalization performance, $\hat{\beta}$ should also have the smallest norm possible. Such matrix is called the "minimum-norm least-squares solution" of the equation. This solution is unique and is given by $\hat{\beta} = H^{\dagger}T$ where $H^{\dagger}$ is the Moore-Penrose pseudo-inverse of the hidden layer output matrix H. Also, this computation can be improved by considering L_2-regularization for a Ridge regression model.
By using this algorithm, a fully trained network is obtained with very few steps and very low computational cost. This network is usually trained several orders of magnitude faster than a classical feedforward network with gradient-based learning.

In our application case, the model takes $X \in \mathbb{R}^{N \times n}$ as an input representing the N FQDN with their associated n features. $W \in \mathbb{R}^{n \times \tilde{N}}$ and $b \in \mathbb{R}^{1 \times \tilde{N}}$ are randomly generated. H is computed as $H = f(XW + B)$ where $f(.)$ is the activation function of the hidden layer (sigmoid in our case) and $B \in \mathbb{R}^{N \times \tilde{N}}$ having each row equal to b. The target $T \in \{0, 1\}^{N \times 1}$ specifies if each FQDN is "Black" (0) or "White" (1). $\beta \in \mathbb{R}^{\tilde{N} \times 1}$ is computed as $H^{\dagger}T$ (cf. above).

4 Numerical Experiments

This section introduces the numerical experiments performed on the DNS traffic dataset described in section 2 and measured from the Orange Internet Service Provider operational network. We run the simulations on a dataset which is comprised of around 10,000 entries. The input features are normalized (via the formula $X_{norm} = (X - X_{mean})/(X_{max} - X_{mean})$) to avoid large values from being overrepresented. All the results are given after a 3-fold cross validation phase. Furthermore, since the ELM model has randomly generated parameters, all results are given as averages over 20 trials. The size of the ELM neural network spans the interval $\{25, 50, 75, 100, 250, 500, 750, 1000\}$ and several feature

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.



(a) Output distribution. (b) ROC curve.

Fig. 2: Performance measures.

selections are tested. The results are given for the features combination and the hidden layer size yielding the best performance, according to our *a priori* networking expert knowledge. The associated sets of features are also given for these results.

Experiments performed on 512 different feature combinations showed that the best performance is obtained with 100 hidden units and with the following features for the input vector: [hour, ttl_{min}, ttl_{25}, ttl_{50}, ttl_{75}, ttl_{max}, no_error, error]. The TTL-related features are a statistical summary of the TTL distribution. The two classes "botnet-related" (Black) and "legitimate" (White) are respectively represented in the dataset by the values 0 and 1. The distribution of the network's outputs is given by figure 2a. The classification threshold used is equal to 0.7 and the corresponding confusion matrix is given in table 2. The Receiver Operating Characteristic (ROC) curve is given in figure 2b. The precision and recall are thus respectively equal to 0.92 and 0.99. The accuracy is equal to 5.06%, the False Positive rate is equal to 4.36% and the False Negative rate is equal to 0.7%. This means that there are only 0.7% of false alarms and that only 4.36% of the botnet-related traffic is not detected by the model. In addition, the precision and recall values are both high and close to 1, which means the model performs well on this dataset. The total training time for this dataset was roughly 13 seconds on average, compared to approximately 2 hours for MLP to reach the same accuracy level with appropriately tuned values for the learning rate and momentum term.

5 Conclusion

In this paper we aimed at applying Extreme Learning Machines (ELM) models for the task of detecting botnet-related activities on the Internet network by classifying Domain Name System (DNS) traffic. We recalled the main fuinctioning principles of DNS servers and also introduced the dataset we processed

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

Predicted \ Actual	White	Black	
White	1719	155	1874
Black	25	1660	1685
	1744	1815	3559

Table 2: Confusion matrix for the dataset.

to perform botnet detection. Our solution is based on ELM models and algorithms which are efficient implementations of single layered feedforward neural networks. Numerical experiments were presented to validate our proposed approach. ELM models performed well on the dataset while having a very high training speed, with performance comparable to other existing botnet detection solutions.

Future works include not only the improvement of the detection of botnets related activities on the Internet but also their prevention. In this aim, DNS reinforcement via name authentication (i.e. its secure extension DNSSEC) is a promising method for improving overall security.

References

[1] S. Greengard, *The war against botnets*, Communications of the ACM, Vol.55, No.2, pp. 16-18, 2012.

[2] M. A. Rajab, J. Zarfoss, F. Monrose and A. Terzis, *A multifaceted approach to understanding the botnet phenomenon*, Proceedings of the 6th ACM SIGCOMM conference on Internet measurement, pp. 41-52, 2006.

[3] P. Mockapetris, *Domain Names - Implementation and Specification*, RFC 1035, Network Working Group, IETF, 1987.

[4] L. Bilge, E. Kirda, C. Kruegel and M. Balduzzi, *Exposure: Finding malicious domains using passive DNS analysis*, Proc. of NDSS, 2011.

[5] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee and N. Feamster, *Building a Dynamic Reputation System for DNS*, Proc. of USENIX Security Symposium, pp. 273-290, 2010.

[6] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning*, Springer Series in Statistics, 2009.

[7] Malware Domains, *Malware Domain Block List*, http://www.malwaredomains.com, and Zeus Tracker, *Zeus IP & domain name block list*, https://zeustracker.abuse.ch, *dmoz*, http://www.dmoz.org, 2009.

[8] G.-B. Huang, Q.-Y. Zhu and C.-K. Siew, *Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks*, Proc. of IJCNN, 2004.

[9] G.-B. Huang and H. A. Babri, *Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions*, IEEE Transactions on Neural Networks, Vol. 9, No. 1, pp. 224-229, 1998.