International Journal of Intelligent Information Systems
2017; 6(6): 67-71
http://www.sciencepublishinggroup.com/j/ijiis
doi: 10.11648/j.ijiis.20170606.11
ISSN: 2328-7675 (Print); ISSN: 2328-7683 (Online)

SciencePG
Science Publishing Group

Classification for DGA-Based Malicious Domain Names with Deep Learning Architectures

Feng Zeng, Shuo Chang, Xiaochuan Wan

Han Sight (Beijing) Software Technology Co., Ltd, Beijing, China

Email address:
zengfengscu@outlook.com (Feng Zeng), shuo.sight@gmail.com (Shuo Chang), wanxc@hansight.com (Xiaochuan Wan)

To cite this article:
Feng Zeng, Shuo Chang, Xiaochuan Wan. Classification for DGA-Based Malicious Domain Names with Deep Learning Architectures. *International Journal of Intelligent Information Systems*. Vol. 6, No. 6, 2017, pp. 67-71. doi: 10.11648/j.ijiis.20170606.11

Received: September 29, 2017; **Accepted**: October 8, 2017; **Published**: December 6, 2017

Abstract: The preemptive defenses against various malware created by domain generation algorithms (DGAs) have traditionally been solved using manually-crafted domain features obtained by heuristic process. However, it is difficult to achieve real-world deployment with most research on detecting DGA-based malicious domain names due to poor performance and time consuming. Based on the recent overwhelming success of deep learning networks in a broad range of applications, this article transfers five advanced learned ImageNet models from Alex Net, VGG, Squeeze Net, Inception, Res Net to classify DGA domains and non-DGA domains, which: (i) is suited to automate feature extraction from raw inputs; (ii) has fast inference speed and good accuracy performance; and (iii) is capable of handling large-scale data. The results show that the proposed approach is effective and efficient.

Keywords: Domain Generation Algorithm (DGA), Recurrent Neural Network (RNN), Deep Learning Architecture, Classification, Transfer Learning

1. Introduction

Recently, the percent of various malware families regenerated by domain generation algorithms (DGAs) has been creeping upward [1-2]. DGAs are used to periodically produce a large number of domain names that can be used as rendezvous points with their command and control (C2) servers. Fighting against DGAs becomes increasingly difficult since DGAs can use some feeds to generate domains every day or even more frequently – every hour, which leads to generate domains increase dynamically. That gives attackers the advantage of setting up the C2 server just for a day in order for infected machines – after that has occurred, the attackers can shut down the C2 server and set it up again as the need arises.

Compared with normal domain names, these malicious domain names generated by DGAs have significant differences. Current researches mainly use clustering-classification process to detect abnormal domain names. Some of them use failed (NXDomain) traffic clustering, other many researches based on the classification of statistical features, such as length, bigram, entropy, life span and character frequency distribution, as well as shared hosts requesting the domain [3].

Unfortunately, much of the previous work in DGA detection does not meet the needs of many security applications that require real-time detection and prevention [4-5] because i) these techniques are too slow for most real-world deployments and often take hours to detect malicious domains; ii) the performance of these methods are quite poor in terms of false positives and true positives; and iii) it is often unrealistic for many security applications to use contextual information.

Recent research suggests that deep learning methods [6-11] are achieving state-of-the-art results across a range of difficult problem domains. Deep learning architectures such as deep neural networks, deep belief networks and recurrent neural networks have been applied to fields including computer vision [9] and speech recognition [12] where they produced results comparable to and in some cases superior to human experts.

However, research about deep learning applying to security system is relatively few, especially classification for DGA-based malicious domain names. Motivated by this idea, a new approach based on current popular advanced deep learning architectures [6-10] is introduced into discriminating DGA domains from non-DGA domains in this paper. In summary, this paper makes the following several core contributions.

1) It applies featureless deep learning architectures to the process of features extractor from raw domain names inputs without manual creation of features from domain names and high-parameter tuning during training.
2) The data have 34,000,000 unique samples (i.e., 1000,000 non-DGA domains and 33,000,000 DGA domains). The DGA domains contain 64 malware families collected from real-world mutilple malware feeds, which, to the best of our knowledge, is the first application of deep learning network to such a big dataset on security system.
3) This method is designed to be distributed and capable of handing large-scale data. It supports GPU learning and parallel, which operates significantly faster inference speed and good accuracy performance than existing implementation.
4) It presents a comparative study of five deep learning networks, namely Alex, VGG, Squeeze Net, Inception, Res Net on speed performance, memory usage and accuracy to classify DGA domains based on MX Net deep learning framework [13] for DGAs classification, which can assist with selecting good architectures in the first place.

2. Background

2.1. Domain Generation Algorithms

Over the last few years, most malware families have begun to use a different approach to communicate with their remote servers. Instead of using hardcoded server addresses, some malware families now use a domain generation algorithm (DGA). DGA is a class of algorithm that takes a seed as an input, outputs a string and appends a top level domain (TLD) such as.com,.ru,.uk, etc. It allows malware to periodically create a list of tens of thousands of new DNS names in order to dynamically determine remote download server address and C2 server addresses at run time. As a result, there is a significant increase in the amount of requests that were being directed at the failback DGA domains used by the malware. It also raises the bar for researchers to sinkhole the server addresses. This would prevent attackers from gaining access to the infected machines, making the DGA algorithm quite limitless.

A kind of successful approach to combating these malware is to build a DGA classifier. With the DGA classification it is also possible to see links between different malware samples of the same family. Normally, DGA classifiers are grouped into two categories depend on the features: using additional contextual information from the DNS queries and using domain names only. In the past research, most DGA detection algorithms pay attention to gathering domains and network data nearby DNS queries, then trying to calculate statistical properties to determine if they are DGA generated. However, this way may be not effective in real world because network data gathering will be challenging in many real deploy. In contrast, machine learning has advantage over previous classical DGA algorithms on accuracy performance and time consuming.

2.2. Recurrent Neural Network

Unlike traditional feed forward neural networks (FFNN), Recurrent neural networks (RNNs) [14] contain cycles that feed the network activations from a previous time step as inputs to the network to influence predictions at the current time step. Unfortunately, in practice, training conventional RNNs with the gradient-based back-propagation through time (BPTT) technique is challenging due to the well-known vanishing gradient and exploding gradient problems as the length of the training sequence grows.

To overcome these weaknesses of RNNs, an elegant RNN architecture Long Short-Term Memory (LSTM) was designed by Hochreiter and Schmidhuber [15]. The LSTM architecture performs better than traditional RNNs on tasks involving long time lags tasks that have never been solved by previous RNNs such as learning context-free and context-sensitive languages and image generation. Authors in [16] presented a DGA classifier that leverages LSTM networks for real-time prediction of DGAs without the need for contextual information or manually created features, which to our knowledge, is the first application and in-depth analysis of deep learning to this field.

2.3. Deep Learning Architectures

Deep learning is overwhelmingly revolutionizing in a broad range of applications. In particular, deep Convolutional Neural Networks (CNNs) have gaining huge success in tacking many applications domains, such as computer vision, video analysis and natural language processing over the past few years, where they have achieved significant performance improvements compared to state-of-art methods in the respective domains. Several popular and advanced architectures will be introduced as follows.

Alex Net (2012) [6], proposed by Alex Krizhevsky, was the first famous convolutional neural network to push ImageNet Classification accuracy by a significant stride in comparison to traditional methodologies. It used Re Lu (Rectified Linear Unit) for the non-linear part, instead of a Tanh or Sigmoid function which was the earlier standard for traditional neural networks. It achieved a winning top-5 test error rate of 15.3% in the ILSVRC-2012 competition.

The VGG architecture (2014) [7] was from Oxford. It makes the improvement over Alex Net by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with very small (3 × 3 kernel-sized) convolution filters one after another, which enables it to learn more complex features, and that too at a lower cost.

The Squeeze Net network (2016) [8] was proposed to focus directly on the problem of identifying a CNN architecture with fewer parameters but equivalent accuracy to a well-known model. It achieved Alex Net-level accuracy on ImageNet with 50x fewer parameters.

As the dramatically increased use of computational resources, the Goog Le Net architecture (2015) [9-10] was built on the idea that most of the activations in deep network are either unnecessary (value of zero) or redundant because of correlations between them. It used one deeper and wider Inception network with slightly superior quality, but adding it to the ensemble seemed to improve the results only marginally, which obtains a top-5 error of 6.67% on both the validation and testing data, ranking the first among other participants in the ILSVRC 2014 Classification challenge.

Residual Network (2015) [11], proposed by Microsoft Research, inserted shortcut connections which still performs identity mapping, with extra zero entries padded for increasing dimensions. It is a residual net with a depth of up to 152 layers-8x deeper than VGG nets but still having complexity, which achieved 3.57% error on the ILSVRC 2015 classification task.

3. Method

3.1. Transfer Learning

The pre-trained models are trained on the large hand-labeled Image Net dataset (10,000,000 labeled images depicting 10,000+ object categories). These network gains knowledge from the data, which is complied as “weights” of networks. However, the objective of this paper is to classify DGAs domain names rather than images. This is relatively large in size and very different in content compared to the original image dataset. It is expected that it transfers the learned weights from the Imge Net-trained network to classify beneficially DGAs domain names. This is known as “transfer learning” [17]. Transfer learning is used to improve a high-performance learner for a target domain by transferring information from a related source domain.

In order to transfer the pre trained weights from images to DGAs domain names efficiently and effectively, several useful strategies are proposed as follows.

1) Turn the string type of domain names into byte array type first to match the learned ImageNet models input.
2) Resize the input size [25x25x3] instead of original image size [224x224x3] to save memory usage because the upper bounded length from domain names string is smaller than 25. A clear example is shown in Figure 1.
3) Remove the top output layer and extracted at the antepenultimate layer of deep learning architectures as features because the last layer of pre trained nets are too "overfitted", lower-layer features can are more suitable for classification.

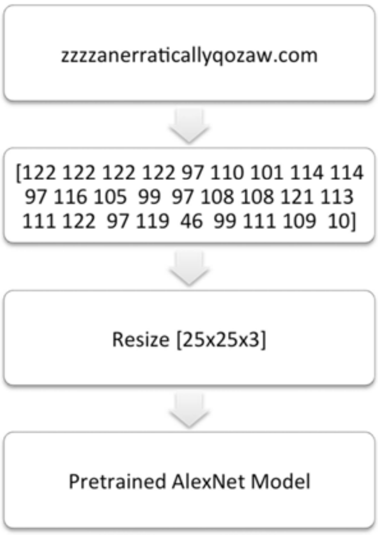

Figure 1. *Process from raw input domain names called zzzzanerraticallyqozaw.com to Alex Net models.*

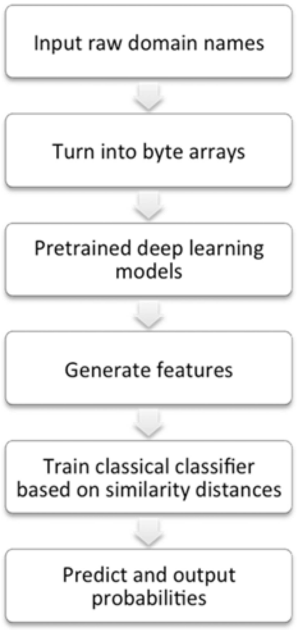
Input raw domain names
Turn into byte arrays
Pretrained deep learning models
Generate features
Train classical classifier based on similarity distances
Predict and output probabilities

Figure 2. *The proposed algorithm steps.*

3.2. Classification

After the features are extracted, a classification module is trained with the domains and their associated labels. A few examples of this module are SVM, Logistic Regression, Random Forest classifier, Decision Trees etc. Additionally, the similarity score between two domain names with the Euclidean distance is used to improve accuracy. The process is show in Figure 2.

This solution has 3 steps:

Step 1: Use the pre-trained deep learning modules to extract features from raw inputs.

Step 2: For each family, compute the average of its features. Use this average as the family feature.

Step 3: Train a Decision Trees classifier for binary-label and predict.

Table 1. *True Positive Rates, False Positive Rates and accuracy for Binary Classifiers.*

Architectures	True Positive Rates	False Positive Rates	Accuracy
Alex Net	0.967086	0.02391	0.97231
VGG16	0.97819	0.02125	0.97296
VGG19	0.97258	0.01714	0.97039
Squeeze Net	0.97461	0.01942	0.97198
Inception-BN-21k	0.97882	0.01831	0.97596
Inception-BN-1k	0.98519	0.01610	0.98196
Inception V4	0.99863	0.01128	0.98568
RseidulNet152	0.99317	0.01659	0.98273

4. Experiments

This section will turn the attention to evaluating the proposed approach and compare its performance to some previous methods on large-scale dataset in this section.

4.1. Datasets

To test the neural networks implementation on a diverse set of benchmarks, some experiments are designed by using data from two sources.

1) The filtered Alexa top 1000,000 domains [18] are used for training domains that are non-DGAs.

2) The 33,000,000 real-world DGA-based malicious domain names are used for DGAs. The data consists of 64 families of DGAs.

4.2. Experimental Setting

All of the experiments in this section and the other techniques were run on an 8-core Intel i7-6700K machine with 32 GB of memory, which supports for two GeForce GTX 1080 cards.

The data set is divided into three parts: 60% training set, 20% cross validation set and 20% test set. Then, run 10-fold cross validation, excluding a different fold from training each time. The final model accuracy (or other evaluation metrics) is defined as the average of the single accuracies.

4.3. Results

Table 1 consolidates the domain names classification results from the experiments. It summarizes true positive rates, false positive rates and the accuracy of different approaches at various computations levels. From the table, it concludes that the best performing Inception V4 module are compared to other CNN baselines.

The speed of inference for above deep learning methods is present in Figure 3. As can be seen, the method with one GPU achieved 1x speed improvement at least than their correspondent methods with CPU. The fastest deep learning architecture is Squeeze Net that can handle almost 5000,000 domain names per day under 2 GPUs.

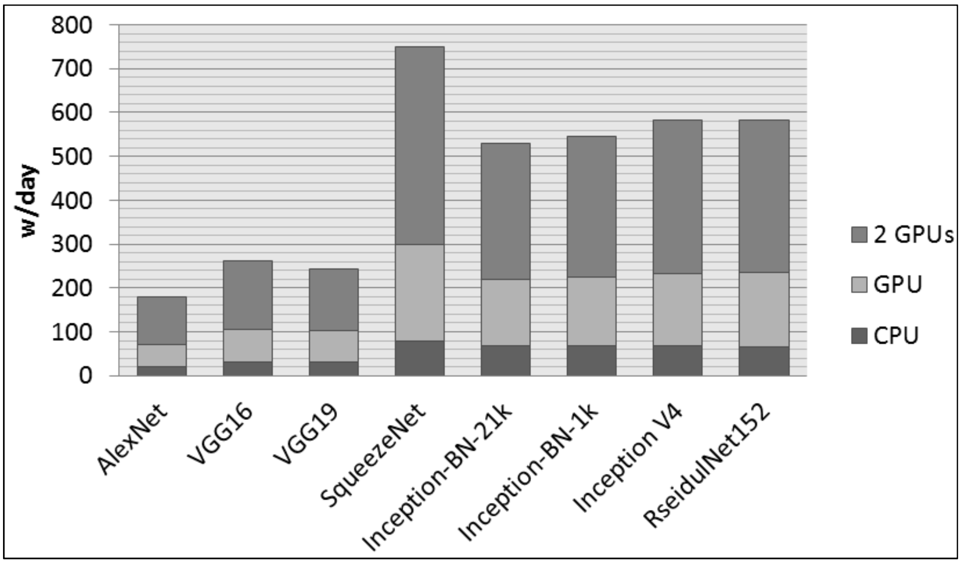

Figure 3. *Speed for Binary Classifiers.*

5. Conclusion

Motivated by the recent overwhelming success of important problems across a wide spectrum of domains, this paper proposed deep learning architectures including Alex, VGG, Squeeze Net, Inception, Res Net for classifying DGA domains and non-DGA domains. It used the pre-trained deep learning modules to extract features from raw inputs without manual creation of features from domain names and high-parameter tuning during training. To our knowledge, it is the first application deep learning architectures based on computer vision into security systems. Fortunately, the best experimental results achieved 99.86% true positive rates with a 0.011 false positive rate.

References

[1] Lever C, Kotzias P, Balzarotti D, et al. A Lustrum of Malware Network Communication: Evolution and Insights [C]. Security and Privacy. IEEE, 2017:788-804.

[2] Antonakakis M, Perdisci R, Nadji Y, et al. From throw-away traffic to bots: detecting the rise of DGA-based malware [C]. Usenix Conference on Security Symposium. 2012:24-24.

[3] Zhang Y, Zhang Y, Xiao J. Detecting the DGA-Based Malicious Domain Names [M]. Trustworthy Computing and Services. Springer Berlin Heidelberg, 2013:130-137.

[4] Woodbridge J, Anderson H S, Ahuja A, et al. Predicting Domain Generation Algorithms with Long Short-Term Memory Networks [J]. 2016.

[5] Anderson H S, Woodbridge J, Filar B. Deep DGA: Adversarially-Tuned Domain Generation and Detection [J]. 2016:13-21.

[6] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks [C]. International Conference on Neural Information Processing Systems. Curran Associates Inc. 2012:1097-1105.

[7] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition [J]. Computer Science, 2014.

[8] Forrest N. Iandola， Song Han， Matthew W. Moskewicz etc. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size [C]. International Conference on Learning Representations, 2016.

[9] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the Inception Architecture for Computer Vision [C]. Computer Vision and Pattern Recognition. IEEE, 2016:2818-2826.

[10] Szegedy C, Ioffe S, Vanhoucke V, et al. Inception-v4, Inception-Res Net and the Impact of Residual Connections on Learning [J]. 2016.

[11] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition [C]. Computer Vision and Pattern Recognition. IEEE, 2016:770-778.

[12] Hinton G, Deng L, Yu D, et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups [J]. IEEE Signal Processing Magazine, 2012, 29(6):82-97.

[13] Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems. In Neural Information Processing Systems, Workshop on Machine Learning Systems, 2015.

[14] Tang S, Han S. Generate Image Descriptions based on Deep RNN and Memory Cells for Images Features [J]. 2016.

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural Computation, 9(8):1735–1780, 1997.

[16] Woodbridge J, Anderson H S, Ahuja A, et al. Predicting Domain Generation Algorithms with Long Short-Term Memory Networks [J]. 2016.

[17] Zhao B, Huang B, Zhong Y. Transfer Learning With Fully Pre trained Deep Convolution Networks for Land-Use Classification [J]. IEEE Geoscience & Remote Sensing Letters, 2017, 14(9):1436-1440.

[18] “Does Alexa have a list of its top-ranked websites?” https://support.alexa.com/hc/en-us/articles/ 200449834-Does-Alexa-have-a-list-of-its-top-ranked-websites-. Accessed: 2016-04-06.