Poster: Experimental Evaluation of Website Fingerprinting using Deep Learning

Payap Sirinam
Center for Cybersecurity
Rochester Institute of Technology
Rochester, New York 14623
Email: payap.sirinam@mail.rit.edu

Mohsen Imani
Department of Computer Science and Engineering
The University of Texas at Arlington
Arlington, Texas, 76013
Email: mohsen.imani@mavs.uta.edu

Matthew Wright
Center for Cybersecurity
Rochester Institute of Technology
Rochester, New York 14623
Email: matthew.wright@rit.edu

***Abstract*—The *website fingerprinting (WF)* attack is one of the most dangerous threats against the *Tor* anonymity system. The attack enables an adversary who can locally observe user's traffic to learn about that user's online browsing activity. Previous studies have shown high rates of attacker effectiveness by applying machine learning techniques such as k-NN and SVM classifiers. The attacker, however, must carefully design the features to be used in these classifiers. In this paper, we explore the use of deep learning for the WF attack, which offers the advantage of not requiring features to be hand crafted. In particular, we applied a Stack Denoising Autoencoders (SDAE) to construct the classifier. Our experimental evaluations show that this technique is fairly effective, at least for a closed-world setting. We believe that further refinement is possible to improve the attack and will explore this in future work.**

I. Introduction

Since so much of users' personal lives is conducted online, privacy for web browsing has become increasingly important.A widely used technology for protecting the privacy of users' web browsing behavior is the Tor anonymity systems [2]. Unfortunately, an adversary can use a website fingerprinting (WF) attack to break Tor's privacy protections. Fig. 1 shows the main WF attack model in which the adversary is monitoring the network between the client and the *guard*, the first Tor node on the client's path, to observe the client's traffic patterns. WF then feeds these traffic patterns as input to a machine learning classifier that aims to identify which website the user has visited. In particular, the adversary trains his classifier to recognize a set of monitored websites. Recent studies have an accuracy of over 90% accuracy against Tor using classifiers such as k-NN [4], SVM [7] and k-FP [9]

While these classifiers are widely used, deep learning has become the state-of-art machine learning technique in many domains, such as speech recognition, visual object recognition, and object detection [5]. Furthermore, deep learning does not require selecting and fine-tuning features by hand. In the WF domain, there is only one work that we know of on applying deep learning, in which Abe and Goto applied a Stacked Denoising Autoencoders (SDAE) [10]. This direction seems promising, given that Vincent et al. show that SDAE appears to mostly achieve performance with lower classification error compared with Support Vector Machine (SVM), Deep Belief Networks (DBN), and Stacked Autoencoders (SAE) [3].

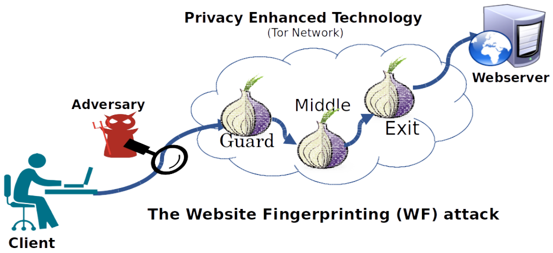
Privacy Enhanced Technology
(Tor Network)
Webserver
Adversary
Middle
Guard
Exit
The Website Fingerprinting (WF) attack
Client

Fig. 1. The website fingerprinting adversary model.

In this paper, we conduct a preliminary experimental evaluation of WF attacks using deep learning. We begin by working with SDAE and following what Abe and Goto proposed [10] to gain a better understanding of how SDAE can be applied to WF. We have tried to reproduce their work and thoroughly investigated on how to apply SDAE to maximize its performance. As future work, we plan to apply other deep learning techniques such as Convolutional Neural Networks (CNN) and to measure the performance of attack against the state-of-art defensive mechanisms such as WTF-PAD [8]. We discuss our current works and preliminary results in the following section.

II. WF Attack using Deep Learning

A. Dataset

We use the same dataset as Wang et al. used for their study [4]. The dataset contains 100 monitored websites, where each website was downloaded 90 times. Each download generates one *instance*, and each instance comprises a list of (time, direction) pairs, where the time indicates when the Tor cell was seen and the direction indicates whether it was sent or received at the client.

Following standard machine learning techniques, we split the instances for each website into three groups: 58 instances for training data, 14 instances for validation data, and 18 instances for testing data. Abe and Goto categorized data into 72 instances for training and 18 instances for testing [10], with no validation data.

B. Dataset pre-processing

Before start training the classifier, we checked the dataset and found some invalid instances that contain only a few Tor

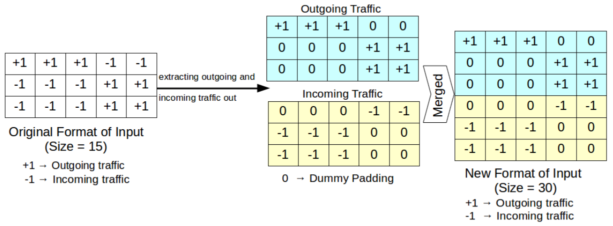

Fig. 2. Customized input to the classifier by first separating outgoing and incoming traffic and then concatenating them.

cells, e.g. some instances contain only ten lines. These cases appear to indicate errors that occurred during data gathering. We thus decided to filter these files out of the dataset. We set the filter value to exclude any instance that has fewer than 200 cells. Moreover, we set the truncate inputs to a fixed maximum size of 5000. If the input contains less than 5000 cells, we pad the input with 0's to create a vector of size 5000, making the input size uniform across instances and sites.

C. Experimental Evaluation

1) SDAE configuration: We apply SDAE by using Theano, a Python library for fast numerical computation for deep learning that can be run on either a CPU or GPU [11]. We study the effect of different parameters, including the number of layers in the neural network, the number of hidden layers, the number of hidden units in each layer, the pre-training learning rate, the learning rate, the batch size, and so on.

2) Data representation: We investigate how to customize the representation of the input data to maximize the performance of the classifier. Even though deep learning is known to be a powerful machine learning method for various types of data, we have found that changing how the data is represented can substantially improve the performance of the classifier. We measured the effects of how different data representations affect the performance of the classifier. We find that separating the outgoing and incoming traffic and concatenating them as two consecutive chunks (outgoing as the first half of the input and incoming as the second half) as shown in Fig. 2 provides the highest performance for our experiment.

3) Results: We investigate the performance of the classifier in the *closed-world* scenario, in which the user is known to be visiting one of the monitored sites. We tested SDAE with both two layers and three layers using the data representation described above. The output layer was realized by a *softmax* function representing labeled monitored websites as class 0 to class 99. We set fundamental SDAE configurations for training including *finetune learning rate* = 0.5, *pre-train learning rate* = 0.001, *batch size* = 50, *pre-training epoch* = 10, *training epoch* = 500. These values come from the configurations providing the highest performance in our tests.

We investigate the accuracy of the classifier with different numbers of neurons in each hidden layer. Table 1 shows the results of the *closed-world* scenario on different numbers of hidden neurons on each hidden layer; performance with three layers was similar. The maximum classifier performance is around 84% for multiple configurations.

TABLE I
RESULTS OF DIFFERENT NUMBERS OF HIDDEN NEURONS ON THE 1st AND 2nd HIDDEN LAYERS (CLOSED-WORLD SCENARIO)

Accuracy		1st layer			
		250	500	750	1000
2nd layer	125	80.07±0.95	82.33±1.02	82.86±0.71	83.66±0.27
	250	81.25±0.62	83.99±0.26	84.17±0.39	83.82±0.50
	500	-	84.33±0.39	84.07±0.95	84.60±0.48
	750	-	-	84.61±0.36	84.59±0.44
	1000	-	-	-	84.93±0.49

III. Conclusion and Future Works

In this paper, we describe the preliminary results of an experimental evaluation of website fingerprinting using deep learning. We applied SDAE with different configurations and find that it works well, though not as effectively as the state of the art. As future work, we plan to examine ways to further improve the performance of the attack, including using other deep learning methods and input representations, studying the performance of the attack in more the realistic open-world scenario, and examining its effectiveness against defenses such as WTF-PAD and Walkie-Talkie [6].

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grants No.1423139 and 1619067.

References

[1] I. Goldberg. "Privacy-Enhancing Technologies for the Internet, II: Five years later," in *Privacy Enhancing Technologies*, pp. 1-12, 2003.

[2] R. Dingledine, N. Mathewson, and P. Syverson. "Tor: The Second-Generation Onion Router," in *Proceedings of the 13th USENIX Security Symposium*. USENIX, 2004.

[3] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol. "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion," *Journal of Machine Learning Research* vol.11, pp. 3371-3408, 2010.

[4] T. Wang, X. Cai, R. Nithyanand, R. Johnson, and I. Goldberg. "Effective Attacks and Provable Defenses for Website Fingerprinting," in *Proceedings of USENIX Security Symposium*. USENIX, pp. 143-157, 2014.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature* vol. 521, pp. 436-444, May 2015.

[6] T. Wang and I. Goldberg. "Walkie-Talkie: An Effective and Efficient Defense against Website Fingerprinting," Technical Report 2015-09, CACR, 2015. [Online]. Availabe: http://cacr.uwaterloo.ca/techreports/2015/cacr2015-09.pdf.

[7] A. Pachenko, F. Lanze, A. Zinnen, M. Henze, J. Pennekamp, K. Wehrle, and T. Engel. "Website Fingerprinting at Internet Scale," In *Proceeding of the 23rd Internet Society (ISOC) Network and Distributed System Security Symposium (NDSS)*, 2016

[8] M. Juarez, M. Imani, M. Perry, C. Diaz, and M. Wright. "Toward an Efficient Website Fingerprinting Defense," in *Proceeding of 21st European Symposium on Research in Computer Security*, pp. 27-46, 2016.

[9] J. Hayes and G. Danezis. "k-fingerprinting: a Robust Scalable Website Fingerprinting Technique," in *Proceeding of 25th USENIX Security Symposium*, pp. 1187-1203, 2016

[10] K. Abe and S. Goto. "Fingerprinting Attack on Tor Anonymity using Deep Learning." in *Proceedings of the APAN - Research Workshop*, 2016.

[11] Theano. [Online]. Available: http://deeplearning.net/software/theano.