# On the Application of Deep Learning Techniques to Website Fingerprinting Attacks and Defenses

# On the Application of Deep Learning Techniques to Website Fingerprinting Attacks and Defenses

Marc Juarez
imec-COSIC KU Leuven
Leuven, Belgium
marc.juarez@kuleuven.be

Vera Rimmer
imec-Distrinet KU Leuven
Leuven, Belgium
vera.rimmer@cs.kuleuven.be

## ABSTRACT

Several studies have shown that traffic metadata can be exploited by a network-level adversary to identify the websites that users are visiting over Tor. The success of such attacks, known as Website Fingerprinting attacks, heavily depends on the particular set of traffic features that are used to distinguish websites. Typically, these features are manually engineered and static which makes them fragile to changes in the Tor protocol and the deployment of defenses. In this work we evaluate a traffic analysis attack based on *deep learning* techniques that allows us to extract features automatically. We show that our attack's performance is comparable to that of traditional attacks, while eliminating the need for feature design and selection. We argue that this may be a game-changer in the arms-race between Website Fingerprinting attacks and defenses.

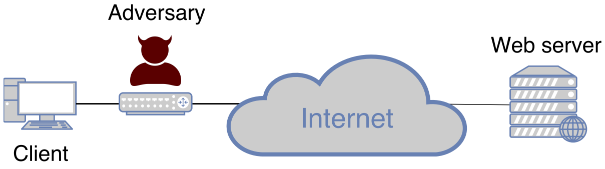

**Figure 1: Representation of the WF threat model. The adversary has access to the communication between the user and the web server.**

## 1 INTRODUCTION

Since the first studies on *Website Fingerprinting* (WF) were published, in the mid-nineties, this problem has received increasing attention. One of the reasons for such interest is the magnitude of the threat that WF poses: if successful, WF allows to recover the browsing history of Internet users, circumventing encryption and anonymization technologies that may be in place. In particular, the Tor network, the nowadays most popular anonymity system, has been shown to be vulnerable to WF in several studies [1–5]. Yet, the practical feasibility of WF techniques is an open question. There are several studies that dispute the effectiveness of the attacks in practice. A 2014 study criticizes attack evaluations for making assumptions that give an unrealistic advantage to the adversary, overestimating the actual effectiveness of the attacks [6]. In particular, most evaluations limit the set of pages that a user can visit – from billions of pages to a few thousands – which may bias conclusions drawn from such evaluations. Moreover, recent studies have investigated the scalability of the attacks in larger sets of pages and suggested that the attacks may not scale [4, 5, 7].

Another reason for such a growing interest in WF is the adoption of machine learning methods in security and privacy research. The use of machine learning techniques in WF has shaped research in the field as an arms-race between attacks and defenses: new attacks defeat defenses because they exploit web traffic features that had not been considered before and, conversely, new defenses are designed to conceal the features that those attacks exploited. This arms-race is caused in part by the methodology followed in WF research for feature selection: features are defined based solely on intuition and expert knowledge and are fixed for a given attack.

*Deep learning* (DL) is the natural next step in this arms race. As deep learning has been shown to outperform traditional machine learning models in other fields such as image and speech recognition, it is reasonable to think these techniques will also boost the performance of traffic analysis techniques such as WF. Moreover, DL techniques can also provide *automatic* and efficient methods to generate website-identifying features, eliminating the need for a human expert. The fact that these feature sets might be optimal and complete can settle the arms-race between attacks and defenses, as no new attacks can be proposed from the discovery of new features.

With this paper we contribute to research in WF by reviewing the prior work on WF, including our work that presents a WF attack based on deep learning [8] – an extended version of this work will appear in NDSS 2018. We also discuss the implications of these works for the practicability of the attacks and defenses and identify what are the challenges for future work on DL as applied to WF.

## 2 BACKGROUND

WF attacks apply supervised classifiers to network traffic traces to identify patterns that are unique to a web page. These attacks can circumvent the protection afforded by encryption and the metadata protection of anonymity systems such as Tor. To carry out the attack the adversary first visits a representative set of websites, records the network traffic of his own visits, and extracts from it a template or fingerprint for each site. Later, when the victim user connects to the site, the adversary observes the victim's traffic and tries to find a match with previously recorded templates.

WF can be deployed by *local* adversaries who have access to the first link of the communication between the user and the web server (see Figure 1). There are many entities in a position to access this communication, including wireless router owners, local network administrators or eavesdroppers, Internet Service Providers (ISPs), and Autonomous Systems (ASes), among other. Note that to deploy the attack, the adversary only requires access to record the network packets and does not require to modify, drop or add packets.

Tor is an anonymous communications network specially designed for low-latency applications such as web browsing. In its default mode of operation, Tor routes connections through three-hop circuits and encrypts the traffic in layers using onion routing [9], so that none of the relays can know both the origin and the destination of the communication at the same time. The Tor network is designed to protect its users from local network eavesdroppers and, thus, should protect against WF. However, several studies have shown that an adversary sitting between the user and the entry (*guard*) to the Tor network can deploy a WF attack successfully [1–5].

Evaluations of WF attacks have been criticized for making unrealistic assumptions on the experimental settings that give an unfair advantage to the adversary compared to real attack settings [6]. For instance, they evaluated the attacks on small datasets and assumed the adversary can train on all the sites that the victim may visit. This assumption is known as the *closed-world* assumption. A more challenging scenario is the *open world*, where the user can visit any website even if the attacker has not trained the classifier on it. We have evaluated our deep-learning-based attack in both open- and closed-worlds but only report in this paper the latter. We refer the reader to our technical report for more details [8].

As neural network models for the implementation of the attacks we used Stacked Denoising Autoencoders (SDAE), Long Short-Term Memory (LSTM) units and Convolutional Neural Networks (CNN). We refer the reader to our technical report for a complete background of these techniques and our methodology to tune their parameters [8].

# 3 RELATED WORK

The first WF attack against the Tor network was based on a Naive Bayes classifier and the features were the frequency distributions of packet lengths [10]. Even though their evaluation showed the attack achieved an average accuracy of 3%, the attack was improved by Panchenko et al. using a Support Vector Machine (SVM) [11]. In addition, Panchenko et al. added new features that were exploiting the distinctive *burstiness* of web traffic and increased the accuracy of the attack to more than 50% accuracy.

These works were succeeded by a series of studies that claimed to improve the accurady of the attacks over 90% success rates. First, Cai et al. [1] used an SVM with their custom kernel based on an edit-distance and achieved more than 86% accuracy for 100 sites. The edit distance allowed for delete and transpose operations, that are supposed to capture drop and retransmission of packets respectively. Following a similar approach, Wang and Goldberg [2] experimented with several custom edit distances and improved Cai et al.'s attack to 91% accuracy for the same dataset.

The three last attacks outperform all the attacks described above and, for this reason, we have selected them to compare with our attack. Each attack uses a different classification model and feature sets and work as follows:

**Wang-kNN [3]:** this attack is based on a k-Nearest Neighbors (k-NN) classifier with more than 3,000 traffic features. They proposed to weigh the features of a custom distance metric, minimizing the distance among traffic samples that belong to the same site. Their results show that this attack achieves 90% to 95% accuracy on 100 websites [3].

**CUMUL [4]:** CUMUL is based on an SVM with a Radial Basis Function (RBF) kernel. CUMUL uses the cumulative sum of packet lengths to derive the features for the SVM. Their evaluations demonstrate an attack success of 90-93% on 100 sites.

**k-Fingerprinting (k-FP) [5]:** Hayes and Danezis's k-FP attack is based on Random Forests (RF). Their feature sets include 175 features developed from features available in prior work, as well as novel timing features. They use the leafs of the random forest to encode a new representation of the sites they intent to detect that is relative to all the other sites in their training set. The new representation of the data is fed to a k-NN classifier for the actual classification. Their results show the attack is as effective as CUMUL and achieves similar accuracy scores for the same number of sites.

All these attacks have selected their features mostly based on expertise and their technical knowledge on how Tor and the HTTP protocol work and interact with each other. Some of these works have also used basic feature selection algorithms to determine the relevance of classifier features. For instance, Hayes and Danezis used the random forest classifier to rank their features [5] and previous studies had analyzed traffic features with regard to WF attacks and defenses [11–13].

Nevertheless, none of these methods provide any guarantees that the feature sets they provide are optimal. Since the features are designed based on heuristics, this leaves an opportunity for improvement by automatically searching for new, perhaps even more effective features for WF.

The first attempt to apply a DL-based approach to WF was made by Abe and Goto [14], where they evaluated a SDAE on the Wang-kNN's dataset [3]. Their classifiers do not outperform the state-of-the-art, but nevertheless achieve a convincing 88% on a closed world of 100 classes. We re-evaluated the reported models in Abe and Goto's paper on the same dataset with support of the authors to confirm their results. It is fair to assume that the lower performance is due to the lack of a sufficient amount of training data for a deep neural network, which, as we confirm in our experiments is deemed necessary for the DL algorithms. In this paper we explore other DL models apart of SDAE and further tune the attacks to perform as accurately as current traditional WF attacks.

# 4 EVALUATION

## 4.1 Datasets

We have followed prior work's methodology for data collection [2, 6]. For the closed-world dataset, we visited the homepage of each of the top Alexa 1,200 sites 3,000 times and dumped the traffic generated by each visit separately. The crawls were parallelized but the visits in each node were sequential and were visited in round-robin order, as described by Wang and Goldberg's batched methodology [2]. After cleansing the datasets by removing erred visits and ensuring we have the same number of visits per site, we ended up having 900 websites, with 2,500 valid network traces each.

## 4.2 Comparison with state-of-the-art attacks

Figure 2 shows the closed world classification accuracy obtained through cross-fold validation for the three traditional WF attacks on a dataset with 100 traces per website ($CW_{100}$). For the same set of website instances, the $k$-NN attack achieves a classification accuracy of 92.87% on our dataset, whereas for CUMUL and $k$-FP we obtain accuracy results of 95.43% and 92.47%, respectively. The obtained results are in line with those originally reported by the authors themselves albeit on other datasets. For this particular setup, the CUMUL attack turned out to be the most successful one.

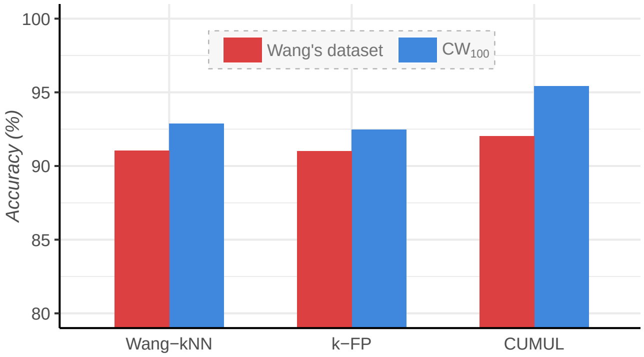

**Figure 2: Re-evaluation of traditional WF attacks on new data**

In the second experiment, we evaluate the same traditional methods on 100 websites, but with a growing number of traces per website, to investigate whether the classification accuracy improves significantly when provided with more training data and whether any WF attack method is consistently better than another.

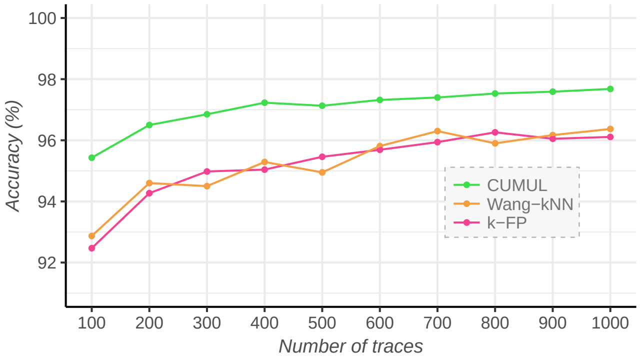

**Figure 3: Impact on the classification accuracy for a growing number of website traces**

In Figure 3, we depict the classification accuracy in a closed-world experiment where the number of website instances grows from 100 to 1,000 traces. Our results show that the CUMUL attack consistently outperforms the two other methods. For all methods, the improvement becomes less evident after 300 website traces. Another interesting observation is that each WF attack – when given sufficient training data – converges to a classification accuracy of approximately 96 − 97%. However, we experienced scalability issues with the $k$-NN based attack by Wang et al., given that the classification running times were at least an order of magnitude higher than those of CUMUL and $k$-FP.

We also assess how the classification accuracy drops when the number of websites increases for a fixed amount of training instances. Given that the CUMUL attack consistently outperformed the other two methods on our dataset, and was superior in resource consumption, we only report the results for CUMUL. We reevaluate the CUMUL classifier on our closed worlds $CW_N$, where $N$ is the number of sites in the world, with 300 traces per website.

Table 1 illustrates that the CUMUL attack obtains a reasonable 92.73% 10-fold cross-validation accuracy for 900 websites using 300 instances each, and a parameter combination of $log_2(C) = 21$ and $log_2(\gamma) = 5$. In general, we observe that the performance degrades gradually with a growing size of the closed world. Moreover, doubling the initial amount of instances gives an advantage of up to 2%, while the amounts higher than 300 stop providing any significant improvement.

| Dataset | CUMUL (100tr) | CUMUL (300tr) | CUMUL (best) |
|---|---|---|---|
| $CW_{100}$ | 95.43% | 96.85% | 97.68% (2000tr) |
| $CW_{200}$ | 93.58% | 95.93% | 97.07% (2000tr) |
| $CW_{500}$ | 92.30% | 94.22% | 95.73% (1000tr) |
| $CW_{900}$ | 89.82% | 92.73% | 92.73% (300tr) |

**Table 1: CUMUL accuracy for a growing closed world (with 100 traces per website, 300 traces, and the best achieved accuracy for a varying number of traces).**

We choose CUMUL as the reference point for comparing our proposed method with the state-of-the art, as CUMUL performed the best on our closed worlds, and proved to be practically feasible.

## 4.3 DL-based WF attacks

In this study, we evaluate the SDAE, CNN and LSTM networks on four closed worlds of different number of websites. We use the models selected by performing hyperparameter tuning on the $CW_{100}$ dataset. We estimate the models' performance by conducting a 10-fold cross-validation on each dataset. We use two performance metrics to evaluate and compare the models with each other: the test *accuracy* and *loss* functions.

The aspect that had the greatest impact on the performance of the attack was the amount of training data (i.e., the amount of traffic traces for each website). For every closed world experiment, we observed significant improvements for a growing amount of traces. One example of this trend is given in Figure 4 for the $CW_{100}$ dataset, where we vary the amount of instances from 100 to all available 2,500 per class.

First and foremost, from these results we can confirm the *feasibility of the WF attack based on a DL approach with automatic feature learning*. We observe how classification accuracy and loss function gradually improve for all models, in the end reaching the 95.46, 96.26 and 94.02% success rate for SDAE, CNN and LSTM model accordingly. These results are comparable to the ones achieved by traditional attacks.

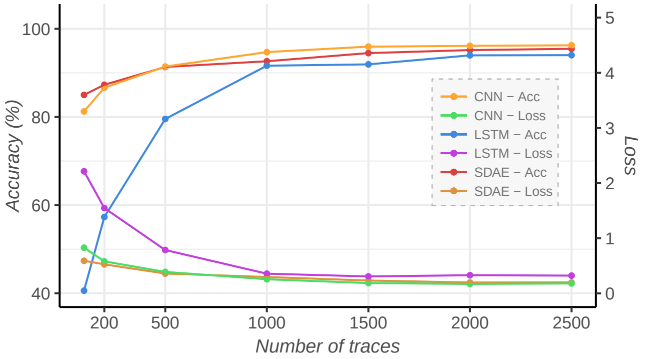

**Figure 4: Accuracy, loss and evaluation time of the DL models (SDAE, CNN, LSTM) for $CW_{100}$ and a growing number of traces**

If we compare the three DNNs with each other, we observe that the SDAE and CNN networks consistently perform better than the LSTM in terms of classification accuracy, with CNN outperforming the other two. Nevertheless, given that LSTM classifies traffic traces based solely on their first 150 Tor cells – compared to the SDAE and CNN that use up to 5,000 and 3,000 cells from each trace –, the achieved performance still appears promising. We interpret that even a small part of the traffic trace is sufficient for website recognition up to 94% accuracy when deploying a model that is able to exploit temporal dependencies of the input sequence. Notably, LSTM performs much poorer when trained on fewer traffic traces than SDAE and CNN, but later gains comparable recognition rate at 1,000 training instances per class.

Next, we assess whether the selected DL models tuned on $CW_{100}$ perform similarly when applied to larger datasets: $CW_{200}$, $CW_{500}$ and $CW_{900}$. The closed world evaluation results remain comparable to CUMUL's results presented in Figure 5 compares the DL-based methods to CUMUL. This comparison illustrates that our DL-based attack can indeed successfully eliminate the complex feature engineering.

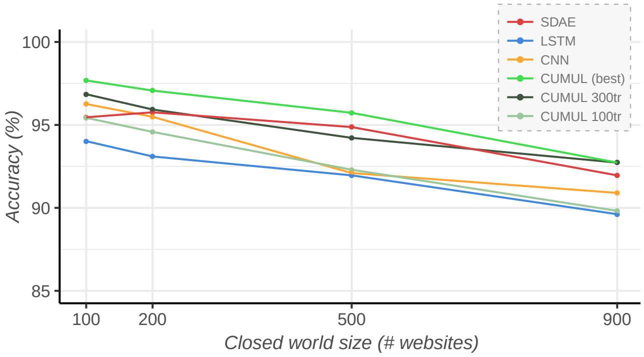

**Figure 5: DL (SDAE, CNN, LSTM) vs. CUMUL for a growing size of the closed world from 100 to 900 websites.**

## 5 DISCUSSION

While DL eliminates the need for feature engineering, the learning method does not produce an explicit representation of the features that can be easily interpreted by a human analyst. The next step would be to produce countermeasures against techniques exploiting these implicit features. Therefore, future work should focus on interpreting these features and compare them with existing ones. However, this is a complex and unsolved task common in DL and ML in general [15].

One approach to confirm the presence of new features would be to analyze the impact that defenses have on concealing existing features in Tor traffic. For instance, an interesting direction for future work could be investigating to what extent the proposed WF defenses [16–18] can be adapted to isolate single known features as a way to identify the new ones.

Another line of research for future work is to investigate defense strategies specifically designed to mitigate the attack that we have presented in this work. Such research could base on the latest work on *adversarial examples* [19]. Adversarial examples are attacks that consist in crafting classification examples that fool the neural network into classifying them into a wrong class. In DL-based WF attacks, adversarial examples can be seen as WF defenses that optimize the bandwidth overhead. We propose to build upon work [20] and design a server-side defense that applies adversarial examples techniques to change the hosted site and protect it against the attacks presented in this paper. Further, that would enable the application of Generative Adversarial Networks (GAN) in the WF problem [21].

## 6 CONCLUSION

In this study, we propose and evaluate a WF attack based on DL. The main objective was to assess the feasibility of WF through automated feature learning. We show that deep neural networks are capable of fingerprinting websites with an accuracy that is comparable to the best-performing approaches among numerous research efforts in recent years. In conclusion, using DL gives an adversary major advantages, resulting in accurate and efficient traffic deanonymization. We hope our work will encourage future research on DL as applied to WF an other traffic analysis problems.

## ACKNOWLEDGEMENTS

A special acknowledgment to Davy Preuveneers, Tom Van Goethem and Wouter Joosen for their contribution to this research. This research is partially funded by the Research Fund KU Leuven. This material is based upon work supported by the European Commission through KU Leuven BOF OT/13/070, H2020-DS2014-653497 PANORAMIX, and H2020-ICT-2014-644371 WITDOM. This material is based upon work supported by the Research Fund KU Leuven. Marc Juarez is funded by a PhD fellowship of the Fund for Scientific Research - Flanders (FWO).

## REFERENCES

[1] Xiang Cai, Xin Cheng Zhang, Brijesh Joshi, and Rob Johnson. Touching from a distance: Website fingerprinting attacks and defenses. In *ACM Conference on Computer and Communications Security (CCS)*, pages 605–616. ACM, 2012.

[2] Tao Wang and Ian Goldberg. Improved website fingerprinting on Tor. In *ACM Workshop on Privacy in the Electronic Society (WPES)*, pages 201–212. ACM, 2013.

[3] Tao Wang, Xiang Cai, Rishab Nithyanand, Rob Johnson, and Ian Goldberg. Effective attacks and provable defenses for website fingerprinting. In *USENIX Security Symposium*, pages 143–157. USENIX Association, 2014.

[4] Andriy Panchenko, Fabian Lanze, Andreas Zinnen, Martin Henze, Jan Pennekamp, Klaus Wehrle, and Thomas Engel. Website fingerprinting at internet scale. In *Network & Distributed System Security Symposium (NDSS)*, pages 1–15. IEEE Computer Society, 2016.

[5] Jamie Hayes and George Danezis. k-fingerprinting: A robust scalable website fingerprinting technique. In *USENIX Security Symposium*, pages 1–17. USENIX Association, 2016.

[6] Marc Juarez, Sadia Afroz, Gunes Acar, Claudia Diaz, and Rachel Greenstadt. A critical evaluation of website fingerprinting attacks. In *ACM Conference on Computer and Communications Security (CCS)*, pages 263–274. ACM, 2014.

[7] Andriy Panchenko, Asya Mitseva, Martin Henze, Fabian Lanze, Klaus Wehrle, and Thomas Engel. Analysis of fingerprinting techniques for tor hidden services. In *to appear in the proceedings of the ACM Workshop on Privacy in the Electronic Society (WPES)*, page 11. ACM, 2017.

[8] Vera Rimmer, Davy Preuveneers, Marc Juarez, Tom Van Goethem, and Wouter Joosen. Automated feature extraction for website fingerprinting through deep learning. In *to appear in the proceedings of the Network & Distributed System Security Symposium (NDSS)*, page 13. IEEE Computer Society, 2017.

[9] Roger Dingledine, Nick Mathewson, and Paul F. Syverson. "Tor: The second-generation Onion router". In *USENIX Security Symposium*, pages 303–320. USENIX Association, 2004.

[10] Dominik Herrmann, Rolf Wendolsky, and Hannes Federrath. Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial Naïve-Bayes classifier. In *ACM Workshop on Cloud Computing Security*, pages 31–42. ACM, 2009.

[11] Andriy Panchenko, Lukas Niessen, Andreas Zinnen, and Thomas Engel. Website fingerprinting in onion routing based anonymization networks. In *ACM Workshop on Privacy in the Electronic Society (WPES)*, pages 103–114. ACM, 2011.

[12] Kevin P. Dyer, Scott E. Coull, Thomas Ristenpart, and Thomas Shrimpton. Peek-a-Boo, I still see you: Why efficient traffic analysis countermeasures fail. In *IEEE Symposium on Security and Privacy (S&P)*, pages 332–346. IEEE, 2012.

[13] Xiang Cai, Rishab Nithyanand, Tao Wang, Rob Johnson, and Ian Goldberg. A systematic approach to developing and evaluating website fingerprinting defenses. In *ACM Conference on Computer and Communications Security (CCS)*, pages 227–238. ACM, 2014.

[14] K. Abe and S. Goto. Fingerprinting attack on tor anonymity using deep learning. In *in the Asia Pacific Advanced Network (APAN)*, 2016.

[15] David Gunning. Explainable artificial intelligence (xai). *Defense Advanced Research Projects Agency (DARPA), nd Web*, 2017.

[16] Xiang Cai, Rishab Nithyanand, and Rob Johnson. CS-BuFLO: A congestion sensitive website fingerprinting defense. In *Workshop on Privacy in the Electronic Society (WPES)*, pages 121–130. ACM, 2014.

[17] Marc Juarez, Mohsen Imani, Mike Perry, Claudia Diaz, and Matthew Wright. Toward an efficient website fingerprinting defense. In *European Symposium on Research in Computer Security (ESORICS)*, pages 27–46. Springer, 2016.

[18] Tao Wang and Ian Goldberg. Walkie-talkie: An efficient defense against passive website fingerprinting attacks. In *USENIX Security Symposium*, pages 1375–1390. USENIX Association, 2017.

[19] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy (S&P)*, pages 39–57, 2017.

[20] Giovanni Cherubin, Jamie Hayes, and Marc Juarez. "Website fingerprinting defenses at the application layer". In *Privacy Enhancing Technologies Symposium (PETS)*, pages 168–185. De Gruyter, 2017.

[21] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.