Performance Analysis of Tor Website Fingerprinting over Time using Tree Ensemble Models

Hyoungseok Oh
Department of Software Science
Dankook University
Yongin-si, South Korea
paris105992@gmail.com

Donghoon Kim
Department of Computer Science
Arkansas State University
Jonesboro, AR, USA
dhkim@astate.edu

Won-gyum Kim
AiDeep
Seoul, South Korea
wgkim@aideep.ai

Doosung Hwang
Department of Software Science
Dankook University
Yongin-si, South Korea
dshwang@dankook.ac.kr

***Abstract*—Tor (The Onion Router) ensures network anonymity by encrypting contents through multiple relay nodes. Recent studies on website fingerprinting (WF) showed that websites can be identified with high accuracy by analyzing traffic data. However, websites are changing over time by updating contents, which can significantly reduce the accuracy of WF attacks. This study analyzes the performance over time by using ensemble models with excellent WF attack performance. The experiment are conducted in two cases with the initial model. The *not updated* analyzes the accuracy of models made from initial data over time, whereas the *updated* adds data that has changed over time to update the model to analyzes the accuracy. The average accuracy of the initial ensemble models is over 90.0% and the Rotation Forest algorithm shows high performance of 93.5%. Comparing the models trained after 30 days with the initial model, the classification performance dropped in both cases; the *not updated* dropped by more than 30.0% and the *updated* dropped by about 10.0%. The experimental results suggest that WF using machine learning may require model learning on a regular basis.**

***Index Terms*—Tor network, fingerprinting, packet based feature, decision tree, tree ensemble**

I. Introduction

The Tor (The Onion Router)[1] is a Firefox-based anonymous network web service, with more than 1 million users worldwide. The Tor network ensures client anonymity by applying TLS (Transport Layer Security) between the user (Tor browser) and each relay node with 3 relay nodes—guard, middle, and exit nodes [1].

However, an anonymous network vulnerability has been reported through website fingerprinting (WF) techniques using traffic data analysis [2]–[4]. The WF attack aims to identify the website visited by clients without analyzing or changing the packet content of the network traffic, and the adversary uses the traffic data generated when the client uses the Internet. Practical examples include government surveillance, stalkers, local area network (LAN) managers, attackers attacking Tor network entry nodes, attacks advertising ISPs, etc.

The adversary first trains a classification model by collecting data set within a period of time. The trained classifier is used to predict whether the target website is accessed. It has been discovered that time affects the classification accuracy of WF attacks. There is a problem that the performance of the trained model deteriorates over time and this reason appears in changes in the structure and content of the website [5], [6]. For example, news websites keep changing their content (e.g., images and texts) dynamically. As a result, traffic patterns change over time depending on the contents and characteristics of the websites. The classification models may have difficulty classifying updated websites.

The purpose of this study is to assess the accuracy of WF over time in the popular websites based on the Alexa categories[2]. The experiment is conducted in two cases: (1) ***Not updated***: Initially, data is collected and trained to make a model, and the WF is performed only with the model using initial data (fixed); and (2) ***Updated***: WF is performed by collecting and training the initially created data to create a model, and collecting the data over time to update the model. The contributions of this study is three-fold. First, this study proposes features for classification methods on Tor network based on network traffic and cell sequence information, and then shows selecting important features contributes to improving performance. Second, this study evaluates the performance of tree-based ensemble models for category classification. Third, this study analyzes accuracy of the models over time in two cases (*Not updated* and *Updated*).

The paper is organized as follows. Section II illustrates related works. Section III describes the method of data collection, and suggests the problems and features of website fingerprinting using tree ensemble models. Section IV evaluates the performance of the models, and analyzes the accuracy of category classification over time. Finally, Section V concludes

[1]https://www.torproject.org/

[2]https://www.alexa.com/topsites/category

this paper with future research directions.

II. Related Work

Previous research deals with contents classification for website fingerprintings under closed [2], [7]–[12] and open [2], [12]–[14] world models. The closed-world problem solves the multi-classification for websites accessed by clients through previously learned website information. On the other hand, the open-world problem performs a binary classification according to whether a client has access to a set of monitored sites.

Panchenko *et al.* collected closed-world data from 775 websites and 4,000 URLs from the web statistics service Alexa as open world dataset for the training data [2]. They removed the data that had only header information of the packet, and conducted website fingerprinting based on traffic information. By applying SVM (Support Vector Machine) algorithm [15], the two anonymous network services—JAP and Tor—showed the detection rate of 80% and 55% for closed-world. They also conducted a WF attack in an open world environment and achieved a true positive rate of 73%.

Pancheko *et al.* collected 300,000 website data and proposed CUMUL feature vectors for real-world website fingerprinting over the Tor network [7]. In the experiment, CUMUL showed higher website fingerprinting performance as the size of training data increased. The closed world scenario compared the classification performance with the Wang *et al.* data set [16], showing 91.3% performance. On the other hand, in the open world scenario, experiments were conducted while increasing the number of websites accessible to clients, reporting a classification performance of 80.0%.

Rimmer *et al.* [10] showed an adversary can automate the feature engineering process with their novel method based on deep learning on a closed-world problem. They collected data over time and used CUMUL, k-NN [15], and k-FP for classification models. Their results showed CUMUL outperformed the two other methods—k-NN and k-FP. The performance of the first collected training data showed an accuracy of 95.0%, and after 56 days, the classification performance of the training data of the same website showed an accuracy of about 66%.

The existing website fingerprinting attack assumed that the distribution of training and test data was the same. However, in realistic network traffic analysis, data distribution changes frequently [5]. Liberatore and Levine showed that the longer delay between training and test sets of traffic instances results in lower accuracy [17]. Pattern differences among training and testing instances are due to non-deterministic packet fragmentation, web page updates, various performance of Tor circuits, dynamic content, etc [5]. The content of the website may change dynamically over time, and may have dynamic content such as frequent replacement of pictures or videos [1]. It may take hundreds of hours or more to train a classifier for website fingerprinting, and an attacker may not have the latest version of the page visited by the client. Therefore, an attacker cannot keep up with the dynamic and changing content of a website.

III. Research Approach

The effectiveness of WF attacks depends heavily on the characterization of traffic features used to construct WF. Thus, the appropriate features should be selected for the learning classification. The steps for performing WF consist of data collection, data preprocessing, feature extraction, model learning, and model evaluation.

A. Threat Model

An adversary can observe the network traffic from a client to the entry Tor router (entry guard) and the traffic from the exit Tor router to a destination client to de-anonymize the connection. Examples of adversaries may be a Tor router owner, ISP (Internet Service Provider), or local network administrator. In this paper, we assume that an adversary monitors the network traffic in the broadcast domain which is between the client and the first router as in Figure 1. The adversary has abundance of training data for websites that a user is accessing.

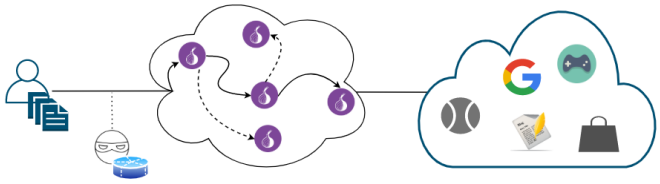


Fig. 1: The system architecture for Tor data collection

B. Data collection

Data were collected from 5 categories out of 17 ones. Data collected using Wireshark [18] consists of traffic data that is captured between a client and the entry node [5]. The content of the website is modified or updated, but the frequency of the correction or update of the content is not constant. For example, a website that represents news contents changes its contents on a daily basis. Thus, even if the information on the website changes, data were collected at regular intervals for 30 days in order to build a robust website fingerprinting model.

To collect data from network traffic similar to Tor environment, data were collected through various circuits after reconnecting. This method is used to collect various data because the Tor browser periodically (every 10 mins) reconfigures the circuit, and the data received from different circuits is different due to the influence of the middle relay node [5]. 150 instances from the top 20 websites in each category were collected and the collection time was set to 200 seconds to fully load the websites.

10 batches were iterated for data collection. At the end of each batch, the Tor browser is restarted to form new circuits. The date when the data was first collected was July 27, 2019. Additional data collection was conducted on July 21 (3 days), July 24 (7 days), July 31 (14 days) and August 16 (30 days). The experiments were conducted based on the date when the data was collected because the time required for data collection

TABLE I: The number of website data per category

Category	No. of data	Ratio	Websites
Web search	3,049	14.2%	google.com youtube.com mai.google.com
Sports	6,000	28.0%	espn.com cricbuzz.com espncricinfo.com
Game	2,000	9.3%	twitch.tv roblox.com store.steampowered.com
Article	4,392	20.5%	stackoverflow.com udemy.com nim.nih.gov
Shopping	6,000	28.0%	amazon.com netflix.com ebay.com
Total	21,441	100.0%	

TABLE II: Comparison of feature vectors

Studied	Type	Feature	No.
Time-based feature [9]	Network traffic	Forwardedd interarrival time Backwarded interarrival time Flow interarriaval time Active Idle Flow bytes per sec Flow packets per sec Duration	23
Traffic mining [13]	Network traffic	Flow direction No. of bytes & packets Packet length (PL) Interarrival time (IAT) PL-IAT statistics TCP header feature IP header feature No. of connections	81
CUMUL [7]	Cell	Cummulative packet length	104
This study	Network traffic	Packet general information Packet interarrival time	103
	Cell	Burst information	

was different for each website. Table I shows the total size, ratio of each category, and top 3 websites in each category.

C. Feature vectors

Through the preprocessing process of collected packets, a data set suitable for feature extraction is composed. Since the performance, type, and location of each website server are different, there are differences in the traffic instances generated between the client and the server. These differences can be found in packet information, packet time, the number of packet size. As such, the features that indicate differences in traffic instances are extracted.

A packet has two types of fields: header fields and a payload field (data). A packet—a datagram in the network layer—that does not have application data (payload) are called zero-payload packet. The zero-payload packet has only the header information for communication control of TCP, such as connection, terminations, and congestion control. The general packet includes header information and data. Usually, zero-payload packets are used to send acknowledements—TCP ACK packets—between sender and receiver [2]. Since the zero-payload packet is considered as noise, zero-payload packets are removed in the preprocessing process. The features are extracted from the packets containing the payload.

Various information can be obtained from packet headers, such as IP address, port number, sequence number, and ack number, and TCP flags including ACK, SYN, and FIN. However, Tor browser encrypts such information sent to the destination. Instead, Tor browser adds a new packet header for the next onion router so the packet header doesn't include the destination information. Thus, features should be obtained from various information based on a sequence of packets. The network applications use different protocols which generate different sequence of packets. The following information can be used [9]: the arrival time between two packets, the amount of time for active period, and flow bytes for a certain period of time. Because the length of features is variable for each website, statistical data is used to solve the problem of variable length and use it for classification. The statistical information used is a quartile (0.25, 0.5, 0.75, and 1) that can evaluate the maximum, minimum, standard deviation, and the range and central position of the data set. Table II compares the features of the previous study with this study. It is worth noting that CUMUL [7] and this study used almost the same number of features. As indicated in Table II, the two models—this study and CUMUL [7]—used similar numbers of features, but they would be different in performance. In Section IV-B, the differences in performance will be analyzed by experimenting with features proposed in this study and those used in CUMUL [7].

Table III shows the number of features. The detail information for each feature as follows: **Packet general** includes a packet sequence information and length of a packet. The sequence information for packets is determined by the order of packets generated during transmission process. The packet length is determined by the contents of websites (objects) and network status including routers, transport-layer parameters such as maximum segment size. The following information can be extracted from a packet sequence: the total transmission size, total transmission time, number of packets of outgoing and incoming packets, order information of packets, number of packets per second, and packet size per second. **Packet interval time** includes the time interval of the entire packets, outgoing packets, and incoming packets [3]. The time interval of the packet is affected by the node configuration of Tor network, the server's performance and protocol. If packets come through nodes located in multiple countries, the time interval of the packets may increase. **Packet length** includes the total packet size, the incoming packet size, and outgoing packet size. Each website has different information such as contents, objects, libraries, and portlets. Thus, there is a difference in the amount of data to receive such contents. **Burst** includes the number of packets that occur continuously. The webserver transmits packets according to a certain unit of chuck. If the size of chuck is larger than MTU, it is divided into a certain size in the segment of network layer. When one piece of information is divided and transmitted, continuous packets

TABLE III: Extracted Features

Feature	No. of features
Packet general	23
Packet interarrival time	42
Packet length	26
Burst	12
Total	103

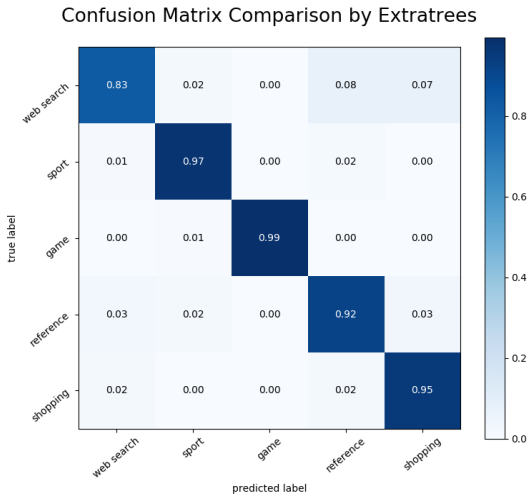


Fig. 2: Confusion matrix

are generated. When three continuous outgoing packets occur, the burst value is defined as 3. The features can be obtained from statistical information based on the number of continuous incoming and outgoing packets.

In addition, Tor onion service websites are known to be vulnerable to fingerprinting attacks due to their limited number and sensitive nature [19]. We found that the Tor onion (hidden) service websites have a certain pattern in the size and the number of incoming and outgoing packets, and that general websites vary in size and number of packets per website. This indicates that features indicating a certain pattern can be used as fingerprinting to distinguish Tor onion service websites and regular websites, even if the contents are encrypted when transmitted.

IV. Experimental Results

The features were extracted from Tor browser according to five categories. The classification was conducted through tree ensemble models [20] such as Decision Tree (DT) [21], Adaptive Boosting (AdaBoost) [22], Random Forest (RanF) [23], Extra Trees (ExtraTrees) [24], Rotation Forest (RotF) [25], and CUMUL [7].

A. Initial Models

For the evaluation of learning model, the parameters of tree models are set by greedy algorithm. Table IV indicates the average of classification performance for multi-classification according to the feature categories. The experiments were conducted with 5-fold cross-validations. The accuracies of the ensemble models for multiclass classification are above 90.0%. The RotF has the highest classificaiton accuracy of 93.5%

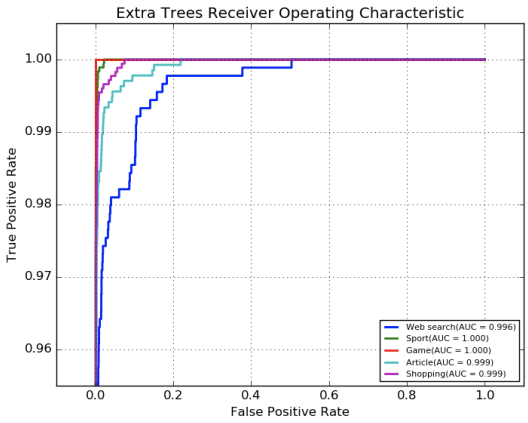


Fig. 3: ROC curve of ExtraTrees

with a relatively long learning time. The ExtraTrees have the 2nd highest classificaion accuracy of 93.4% with the shortest learning time. The accuracy of the models is listed in high order: RotF, ExtraTrees, RanF, DT, and CUMUL. CUMUL has the lowest accuracy because it uses only data transmission and size information to conduct classification.

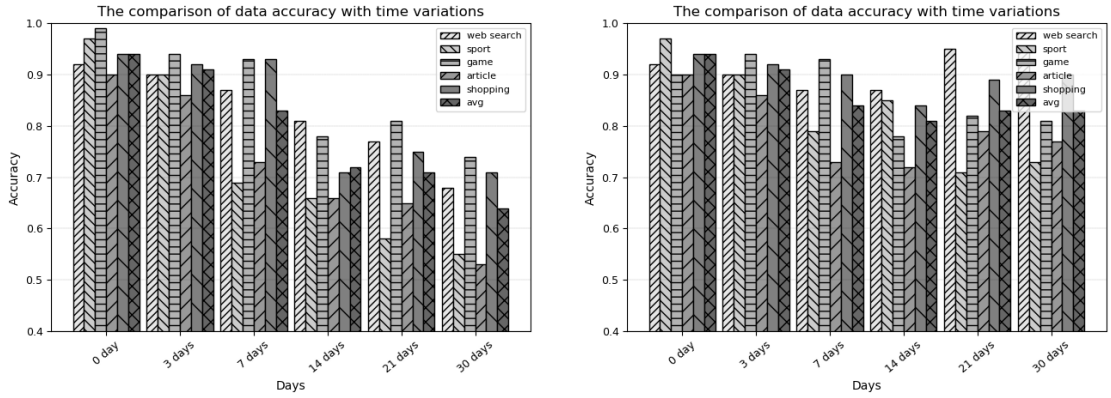
Figure 2 and 3 show confusion matrix and ROC curve for Extra tree algorithm. In multi-classification problems, the contents from shopping, games, and sports classes, which consist of large-scale objects such as images and videos, were higher than the other classes. If the websites have large contents, they have high classification accuracy because the size of the received data and the high number of packets appear as one of characteristics. On the other hand, the website categories—web search—that contain many characters, such as articles and text based contents, have low classification accuracy like web search. The reason is that even though the contents are different, there is no difference in the size and number of the packets if there is no difference in the size of the file.

B. Performance comparison

Figure 4 indicates classification accuracy for each category over time. Figure 4(a) shows accuracy for the initially collected learning data (*not updated*). Overall, the accuracies have decreased significantly over time. The root cause of poor accuracy is that the content or format of a website changes over time. After 30 days, the accuracies of sport and article classes have dropped significantly. The reason is that the update of the new content has changed a lot compared to other categories. Figure 4(b) shows accuracy of a new learning model with the addition of newly collected data to the learning data initially collected (*updated*). The accuracies have been reduced relatively slightly. Data collected over 30 days showed 83.2% accuracy when continuously updating the learning model (Figure 4(b)), while 63.2% accuracy if not updated (Figure 4(a)). In the case of the sport class where website information is modified in a short period, the largest reduction

TABLE IV: Classification accuracy of the initial models

Metric	DT	AdaBoost	RanF	ExtraTrees	RotF	CUMUL
TPR	0.902 ± 0.086	0.916 ± 0.084	0.916 ± 0.070	0.934 ± 0.065	0.935 ± 0.053	0.833 ± 0.095
FPR	0.024± 0.017	0.020 ± 0.015	0.019 ± 0.015	0.015 ± 0.011	0.016 ± 0.010	0.038 ± 0.005
F1-score	0.907±0.067	0.922 ± 0.063	0.922 ± 0.063	0.939 ± 0.050	0.936 ± 0.047	0.838 ± 0.012
Avg. Time(sec)	6.13	106.59	29.84	4.45	148.61	21.77



(a) Not updated

(b) Updated

Fig. 4: Accuracy comparison over time

is shown at 42.0% compared to the existing classification accuracy. On the other hand, the web search class with a relatively long cycle of change showed the lowest decrease of 24.0%.

Table V represents the previously studied classification accuracy (CUMUL [7]) and the accuraies of this work using five tree ensemble models. For comparison with the previously studied CUMUL, 100 traffic data of the same size were continuously trained. After 3 days, there is not much difference in accuracy between *not updated* and *updated* of all models. After 14 days, the difference in accuracy between *not updated* and *updated* began to appear in all models. After 30 days on all models, the *not updated* dropped by about (or more) 30.0% and the *updated* dropped by about 10.0%. The ExtraTrees has the highest classificaiton accuracy, while CUMUL has relatively low accuracy compared to the other models. These results indicate that the model preseneted in this paper is excellent. As in Table II, CUMUL [7] used 104 features and this model used 103 features. While the numbers of features are almost the same, each model used different features to elicit the characteristics of the websites. As such, it has led to a difference in performance.

In the WF classification problems, time and burst information are the important features [9], [26]. To verify the effectiveness of the burst feature set, the feature importances are analyzed with the result obtained from random forest model. As shown in Table VI, the top 15 feature importances are quantified according to the frequently used features after learning for URLs selected when using general browsers and Tor browsers. The concentration information (Rank 4 in general browser) in the Table VI is the feature of incoming

TABLE V: Model accuracy over time

Model	Case	3 days	14 days	30 days
DT	Not updated	0.850	0.594	0.558
	Updated	0.850	0.771	0.736
AdaBoost	Not updated	0.884	0.655	0.592
	Updated	0.884	0.761	0.744
RanF	Not updated	0.903	0.694	0.629
	Updated	0.903	0.822	0.828
ExtraTrees	Not updated	0.904	0.724	0.624
	Updated	0.904	0.812	0.833
RotF	Not updated	0.915	0.698	0.619
	Updated	0.915	0.773	0.819
CUMUL	Not updated	0.801	0.598	0.322
	Updated	0.801	0.701	0.680

and outgoing chunks in the network [27]. In general broswers, incoming/outgoing ordering and outgoing burst features appear on top, similar to the previously used features. However, in the case of Tor browsers, burst time is of high feature importance, and the burst time interval and the incoming/outgoing burst time features are are ranked around the top. These results suggest that if burst-based features are included in WF, it helps to improve performance.

V. Conclusion

This study analyzed the performance of tree ensemble models for web classification using the traffic information over Tor network. Data were collected from the popular websites based on the Alexa categories. Features were extracted from network traffic and cell information. The models were created by learning the data collected initially. Decision tree-based ensembles outperformed a single decision tree and CUMUL. To analyze the accuracy of the models over time, the experiment was conducted in two cases —*not updated* and *updated*.

TABLE VI: Feature importance

Rank	General Browser	Score	Tor Browser	Score
1	outgoing ordering	6.07	incoming ordering	17.92
2	first 30 outgoing time interval	5.90	burst time	11.11
3	incoming ordering	5.65	concentration	8.53
4	concentration	5.61	outgoing ordering	8.09
5	first 30 incoming burst	4.73	outgoing burst	5.35
6	first 30 incoming time interval	4.66	first 30 time interval	4.95
7	outgoing time interval	4.30	unique packet length	4.02
8	first 30 time interval	4.25	incoming burst	3.76
9	first 30 total burst	4.09	incoming time interval	3.21
10	outgoing burst	3.92	total packet length	3.14
11	incoming burst	3.88	total time interval	2.82
12	incoming time interval	3.62	total burst	2.82
13	total time interval	3.25	outgoing interval time	2.40
14	first 30 outgoing burst	3.00	outgoing burst time	2.37
15	total burst	2.21	incoming burst time	2.32

The accuracies decreased over time in both cases. As such, in order to improve accuracy, it is necessary to construct a training set suitable for website fingerprinting through periodic data collection. For the sustainable Tor website fingerprintings, it is necessary to study the feature selection using additional traffic in cooperation with learning models.

ACKNOWLEDGMENT

This study was conducted as a result of the study of the copyright technology development project of the Ministry of Culture, Sports and Tourism and the Copyright Commissionof Korea in 2020 (No: 2018-real_name-9500).

REFERENCES

[1] T. Wang and I. Goldberg, "Improved website fingerprinting on tor," in *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society*, 2013, pp. 201–212.

[2] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel, "Website fingerprinting in onion routing based anonymization networks," in *Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society*, ser. WPES '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 103–114. [Online]. Available: https://doi.org/10.1145/2046556.2046570

[3] T. Wang and I. Goldberg, "Comparing website fingerprinting attacks and defenses," Technical Report 2013-30, CACR, Tech. Rep., 2014.

[4] P. Winter, A. Edmundson, L. M. Roberts, A. Dutkowska-Zuk, M. Chetty, and N. Feamster, "How do tor users interact with onion services?" in *Proceedings of the 27th USENIX Conference on Security Symposium*, ser. SEC'18. USA: USENIX Association, 2018, p. 411–428.

[5] M. Juarez, S. Afroz, G. Acar, C. Diaz, and R. Greenstadt, "A critical evaluation of website fingerprinting attacks," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014, pp. 263–274.

[6] R. Attarian, L. Abdi, and S. Hashemi, "Adawfpa: Adaptive online website fingerprinting attack for tor anonymous network: A stream-wise paradigm," *Computer Communications*, vol. 148, pp. 74 – 85, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366419300763

[7] A. Panchenko, F. Lanze, J. Pennekamp, T. Engel, A. Zinnen, M. Henze, and K. Wehrle, "Website fingerprinting at internet scale." in *NDSS*, 2016.

[8] Wang, Tao, "Website fingerprinting: Attacks and defenses," 2016. [Online]. Available: http://hdl.handle.net/10012/10123

[9] A. H. Lashkari, G. Draper-Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of tor traffic using time based features." in *ICISSP*, 2017, pp. 253–262.

[10] V. Rimmer, D. Preuveneers, M. Juarez, T. Van Goethem, and W. Joosen, "Automated website fingerprinting through deep learning," *arXiv preprint arXiv:1708.06376*, 2017.

[11] Z. Zhuo, Y. Zhang, Z.-l. Zhang, X. Zhang, and J. Zhang, "Website fingerprinting attack on anonymity networks based on profile hidden markov model," *Trans. Info. For. Sec.*, vol. 13, no. 5, p. 1081–1095, May 2018. [Online]. Available: https://doi.org/10.1109/TIFS.2017.2762825

[12] S. Li, H. Guo, and N. Hopper, "Measuring information leakage in website fingerprinting attacks and defenses," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 1977–1992.

[13] A. Montieri, D. Ciuonzo, G. Aceto, and A. Pescapé, "Anonymity services tor, i2p, jondonym: Classifying in the dark (web)," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 3, pp. 662–675, 2020.

[14] S. Bhat, D. Lu, A. Kwon, and S. Devadas, "Var-cnn: A data-efficient website fingerprinting attack based on deep learning," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 4, pp. 292–310, 2019.

[15] C. Bishop and N. Nasrabadi, *Pattern recognition and machine learning*. springer New York, 2006, vol. 1.

[16] T. Wang, X. Cai, R. Nithyanand, R. Johnson, and I. Goldberg, "Effective attacks and provable defenses for website fingerprinting," in *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, 2014, pp. 143–157.

[17] M. Liberatore and B. N. Levine, "Inferring the source of encrypted http connections," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, ser. CCS '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 255–263. [Online]. Available: https://doi.org/10.1145/1180405.1180437

[18] A. Orebaugh, G. Ramirez, and J. Beale, *Wireshark & Ethereal network protocol analyzer toolkit*. Elsevier, 2006.

[19] R. Overdorf, M. Juarez, G. Acar, R. Greenstadt, and C. Diaz, "How unique is your. onion? an analysis of the fingerprintability of tor onion services," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 2021–2036.

[20] S. Euh, H. Lee, D. Kim, and D. Hwang, "Comparative analysis of low-dimensional features and tree-based ensembles for malware detection systems," *IEEE Access*, vol. 8, pp. 76 796–76 808, 2020.

[21] J. R. Quinlan, *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., 1993.

[22] R. E. Schapire and Y. Freund, *Boosting: Foundations and Algorithms*. The MIT Press, 2012.

[23] L. Breiman, "Manual on setting up, using, and understanding random forests v3. 1," *Statistics Department University of California Berkeley, CA, USA*, 2002.

[24] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, pp. 3–42, 2006.

[25] R. Blaser and P. Fryzlewicz, "Random rotation ensembles," *Journal of Machine Learning Research*, vol. 17, no. 4, pp. 1–26, 2016. [Online]. Available: http://jmlr.org/papers/v17/blaser16a.html

[26] X. Li, Y. Wang, S. Basu, K. Kumbier, and B. Yu, "A debiased mdi feature importance measure for random forests," 2019.

[27] J. Hayes and G. Danezis, "k-fingerprinting: A robust scalable website fingerprinting technique," in *25th {USENIX} Security Symposium ({USENIX} Security 16)*, 2016, pp. 1187–1203.