Vol.15 No.3(2024),184-192
DOI: https://doi.org/10.61841/turcomat.v15i3.14789

Darknet Traffic Analysis: Examining How the ADABOOST Algorithm Affects the Classification of Onion Service Traffic Given Modified Tor Traffic

Mr. Francis Vijay Kumar Anna Reddy [1], K. Yamini [2] , K. Kruthi [3] , K. Akshitha Sree [4]

[1] Assistant Professor, Department of CSE, Malla Reddy Engineering College for Women, Autonomous, Hyderabad,

[2],[3],[4]Student, Department of CSE, Malla Reddy Engineering College for Women, Autonomous, Hyderabad

ABSTRACT:

In order to shape and monitor traffic, it is necessary to classify network traffic. The significance of privacy-preserving technology has increased in the last twenty years due to the growth of privacy concerns. One common method of remaining anonymous while surfing the web is to join the Tor network. This will allow you to remain anonymous while also supporting anonymous services called Onion Services. The problem is that government and law enforcement organizations often take advantage of this anonymity, particularly with Onion Services, and end up de-anonym zing its users. This paper's emphasis is on three primary contributions in an effort to discover the capability to categorize Onion Service traffic. Separating Onion Service communication from regular Tor traffic is our first objective. With over 99% accuracy, our methods can detect Onion Service traffic. On the other hand, Tor traffic may have its information leaking concealed by making a

Few adjustments. We assess the efficacy of our methods in light of these changes to Tor traffic in our second contribution. According to our experiments, under these circumstances, the Onion Services traffic becomes less distinct, with an accuracy decrease of over 15% seen in some instances. We conclude by determining and assessing the effect of the most important feature combinations on our classification task.

INTRODUCTION

Vol.15 No.3(2024),184-192
DOI: https://doi.org/10.61841/turcomat.v15i3.14789

Through the use of a network of intermediate nodes, Tor is able to mask its users' online activity and prevent them from being identified. Onion Services, often called hidden services, are another anonymous service that Tor facilitates; they use the. Onion top-level domain. Due to Tor's censorship-evasion capabilities, security professionals, network opponents, and law enforcement organizations are learning to distinguish Tor traffic from both encrypted and non-encrypted forms of data transmission. For instance, we attempted to distinguish traffic from Tor from non-Tor traffic, differentiating Tor traffic based on application categories, and distinguishing Tor traffic from different anonymity network traffic like I2P and Web-mix. But our goal here is to do traffic analysis to see if Onion Service traffic can be distinguished from regular Tor traffic. To provide the groundwork for our study, we create three research questions. Answering the question is our first priority. Three Tor nodes make up a typical Tor circuit used to access an online service via the Tor network. Six Tor nodes make up an Onion Services circuit, the only means of accessing an Onion Service. Given that both the normal Dar and Onion Service circuits encrypt their traffic, we may presume that we can differentiate between them using the metadata that has leaked (e.g., packet size, direction, timestamps). In the past, Onion Services have hosted illicit websites; recently, however, they have served as botnet C&C servers. Consequently, government and law enforcement organizations want to monitor and control the traffic on the Onion Service in order to shut down and identify such services. To safeguard their systems against possible malicious actors (like hackers) and assaults, even corporations could benefit from limiting access to such websites. Therefore, there are two primary uses for methods that may identify Onion Service traffic: 1. these methods can be used as a foundation for Onion Service fingerprinting. 2. They may be helpful for limiting traffic from the Onion Service in systems that are sensitive or secret. Secondly, we make an effort to examine the same issue in several contexts. One thing we do is try. To modify Tor's traffic patterns, one may use certain strategies. Several examples of such methods include the use of fake bursts and delays, traffic splitting, and the use of padding. Concealing the information leak of Tor traffic is the goal of these techniques1. In order to verify whether our results from RQ1 will be valid after these changes are applied to the Tor traffic, it is crucial to answer RQ2. Assuming these changes are implemented in the future and still distinguish Onion Services traffic, it means they are ineffective in masking that traffic. Questions about the legitimacy of previous efforts, such

Vol.15 No.3(2024),184-192
DOI: https://doi.org/10.61841/turcomat.v15i3.14789

as an in a setup with those changes applied, arise if the alterations actually impact the Onion service classifiably. We propose that RQ2 warrants evaluation since its results may pave the way for other studies on Tor traffic categorization.

RELATED WORK

"Tor: Upcoming Onion Router Generation Two,"

Tor is a low-latency anonymous communication platform that we provide. It is built on circuits. By including features such as directory servers, adjustable exit rules, complete forward secrecy, congestion management, and a workable architecture for location-hidden services through rendezvous points, this the second generation Onion Routing network overcomes drawbacks in the original concept. Tor offers a fair compromise across anonymity, usability, and efficiency; it operates on the actual Internet; it does not need special privileges or kernel changes; and it requires minimal synchronization or cooperation between nodes. In this short, we will go over our experiences using a global network that has over 30 nodes. At the end, we enumerate all the unresolved issues related to anonymous messaging.

"Classifying Tor traffic using a convolutional neural network trained on raw packet headers,"

Analysing and classifying traffic is becoming more important for effective network management and resource allocation due to the exponential growth of network traffic. But new security measures are making this job harder by enabling encrypted communication methods like Tor, which is among the most widely used encryption protocols. In this study, we provide a method for Tor traffic classification that makes use of a convolutional neural network framework and hexadecimal raw packet header. When compared to other machine learning methods, our method demonstrates exceptional precision. In order to publicly verify our strategy, we use the UNB-CIC Tor internet traffic dataset. Our method achieves a fractionized Tor/non-Tor data classification precision of 99.3% according to the trials.

"Using Tor-encrypted data to deduce application types,”

To protect one's privacy while using the internet, many turn to Tor, a well-known anonymous communication technology. It works with TCP programs and encrypts data in equal-sized cells to conceal user information, such the kind of application running (Web, P2P, FTP, Etc). A decrease in the privacy set and the facilitation of additional assaults are two reasons why the recognized application kinds are hazardous. Unfortunately, some application behaviours cannot be concealed by the present Tor architecture. As an example, Tor traffic retains the characteristic of concurrent file uploads and downloads, as is common in P2P applications. We look at a new kind of attack against Tor—the traffic classification attack—that can identify application kinds from Tor traffic because of this finding. An adversary uses an effective machine learning method to simulate various application types after meticulously selecting flow parameters, such as burst volumes and directions, to describe application behaviours. Then, we may utilize these pre-existing models to determine the application type of a target's traffic through Tor and categorize it accordingly. We have shown the viability and efficacy of the traffic categorization attack by implementing it on Tor and conducting trials.

"Mononym: Classifying in the shadows (web)," "Anonymity services tor," and "I2P."

Assigning applications to network traffic—a process known as traffic classification—is a useful tool for a variety of purposes in many domains, including management, research and development, security, and traffic engineering. Anonymity technologies further obfuscate the sender, receiver, and nature of the communication, whereas programs that encrypt communication material to protect users' privacy pose a threat to this process. Using a publicly available dataset from 2017, we present repeatable classification results in this paper. We want to find out how well machine learning methods based on statistical features can identify a particular anonymity tool and the traffic it conceals, in comparison to other tools in the same category. Four classifiers—Naïve Bayes, Bayesian Network, C4.5, and Random Forest—are trained and evaluated on the dataset in order to achieve this goal. It is possible to discriminate between the three anonymity network (Tor, I2P, and Jon Deonym) with a 99.99% success rate, and the results reveal that the particular application causing the traffic may be identified with a 98.00% success rate.

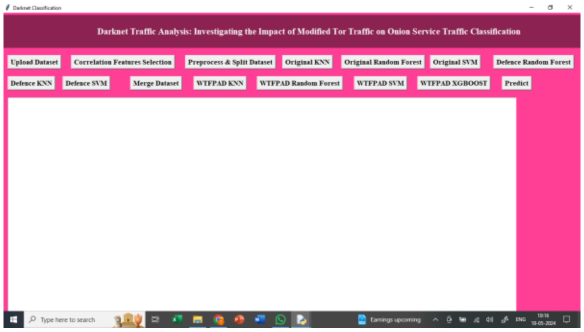
METHODOLOGY

Vol.15 No.3(2024),184-192
DOI: https://doi.org/10.61841/turcomat.v15i3.14789

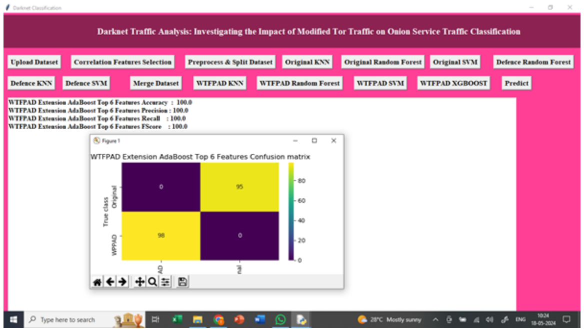
1) Transfer Dataset: The NO DEFENCE initial TOR dataset & the WTFPAD dataset are loaded and shown in this module.
2) How to Choose Correlation Features: Here we load the WTFPAD class labels, define the code to extract 50 features from the dataset using the Information Gain technique, and finally, we see the significance graph for those features.
3) Preparing and Dividing the Dataset: This section divides the data into two parts: training and testing.
4) The first KNN model: Instructions for training the KNN algorithm using the original dataset without defences.
5) Random Forest from the Start: Use the original dataset without defences to train the Random Forest algorithm.
6) A classic SVM: Introduce the SVM algorithm to the original dataset without defenses.
7) Protective KNN: Incorporate defines dataset into KNN algorithm training.
8) Random Forest for Defense: Develop a Random Forest algorithm using a dataset related to defense.
9) Protective SVM: Teach a support vector machine algorithm using a dataset related to defense.
10) Join Datasets: Learn the KNN method using the WTFPAD dataset that has been adjusted.
11) Random Forest using WTFPAD: Create a Random Forest model using the WTFPAD dataset that has been changed.
12) Classification with WTFPAD SVM: Introduce the SVM algorithm to the original dataset without defenses.
13) ADABOOST WTFPAD: Use the WTFPAD dataset to train the ADABOOST algorithm.
14) Predict: Reading test networking data and subsequently categorizing network activity as Tor or Onion services using the extension ADABOOST algorithm is shown in the following screen capture. The test data is shown in square brackets, and the projected service type is shown following the arrow sign.

RESULT AND DISCUSSION

Vol.15 No.3(2024),184-192
DOI: https://doi.org/10.61841/turcomat.v15i3.14789

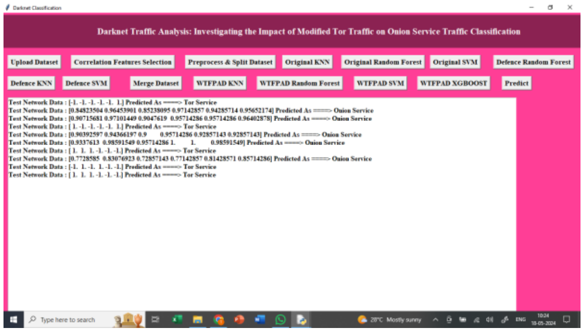


Above, you can see the NO DEFENCE full TOR dataset loaded and displayed.



```
WTFPAD Extension AdaBoost Top 6 Features Accuracy  :  100.0
WTFPAD Extension AdaBoost Top 6 Features Precision : 100.0
WTFPAD Extension AdaBoost Top 6 Features Recall    : 100.0
WTFPAD Extension AdaBoost Top 6 Features FScore    : 100.0
```

After training, the ADABOOST algorithm in the aforementioned screen extension achieved a perfect score of 100%.



```
Test Network Data : [-1. -1. -1. -1. -1.  1.] Predicted As ====> Tor Service
Test Network Data : [0.84823504 0.96453901 0.85238095 0.97142857 0.94285714 0.95652174] Predicted As ====> Onion Service
Test Network Data : [0.90715681 0.97101449 0.9047619  0.95714286 0.95714286 0.96402878] Predicted As ====> Onion Service
Test Network Data : [ 1. -1. -1. -1. -1.  1.] Predicted As ====> Tor Service
Test Network Data : [0.90392597 0.94366197 0.9        0.95714286 0.92857143 0.92857143] Predicted As ====> Onion Service
Test Network Data : [0.9337613  0.98591549 0.95714286 1.         1.         0.98591549] Predicted As ====> Onion Service
Test Network Data : [ 1.  1.  1. -1. -1. -1.] Predicted As ====> Tor Service
Test Network Data : [0.7728585  0.83076923 0.72857143 0.77142857 0.81428571 0.85714286] Predicted As ====> Onion Service
Test Network Data : [-1.  1. -1.  1. -1. -1.] Predicted As ====> Tor Service
Test Network Data : [ 1.  1.  1. -1. -1. -1.] Predicted As ====> Tor Service
```

Analysing test network data and subsequently categorizing network traffic as Tor as well Onion services using the extension ADABOOST algorithm is shown in the following screen capture. Below the arrow sign, we can see the projected service type, and in the square brackets, we can see the test data.

CONCLUSION:

Vol.15 No.3(2024),184-192
DOI: https://doi.org/10.61841/turcomat.v15i3.14789

This study addressed three inquiries about the categorization of Onion Service traffic. We used models trained with supervised machine learning to see whether they could distinguish between Tor traffic and Onion Service traffic. In order to train the machine learning classifiers, we first retrieved fifty characteristics from every traffic trace. Among the classifiers tested, KNN, RF, and SVM had a 99% success rate in differentiating between Tor and Onion Service traffic. We next set out to determine whether the classifiability of Tor traffic is impacted by state-of-the-art Website Fingerprinting defenses. We assessed the impact on the Onion Service traffic categorization of the various modifications introduced by these defences in an effort to conceal information leakage from traffic. When used in conjunction with our feature set, the aforementioned classifiers degrade Onion Service traffic categorization performance, according to our studies. Having said that, we did see that the altered Tor traffic can still be identified. Information gain, Pearson's correlation, and Fischer Score were the three selection of features metrics we utilized to determine which characteristics were most relevant to this assignment. When it came to distinguishing between Onion Service and Tor traffic, those top characteristics had a success rate of over 98%. Nevertheless, after modified network traffic traces were used, they failed to provide the expected findings.

REFERENCES:

[1] R. Dingledine, N. Mathewson, and P. Sigerson, ‘‘Tor: The second generation onion router,’’ in Proc. 13th USENIX Secure. Sump. (SSYM), San Diego, CA, USA, Aug. 2004, pp. 303–320.

[2] M. Al Sabah, K. Bauer, and I. Goldberg, ‘‘Enhancing Tor’s performance using real-time traffic classification,’’ in Proc. ACM Conf. Compute. Common. Secure. (CCS), New York, NY, USA, Oct. 2012, pp. 73–84.

[3] A. H. Lashkar, G. D. Gil, M. S. I. Mamun, and A. A. Ghobadi, ‘‘Characterization of Tor traffic using time based features,’’ in Proc. 3rd Int. Conf. Inf. Syst. Secure. Privacy (ICISSP), Porto, Portugal, Feb. 2017, pp. 253–262.

Vol.15 No.3(2024),184-192
DOI: https://doi.org/10.61841/turcomat.v15i3.14789

[4] M. Kim and A. Analgen, ‘‘Tor traffic classification from raw packet header using convolutional neural network,’’ in Proc. 1st IEEE Int. Conf. Know. Innova. Invention (ICKII), Juju Island, South Korea, Jul. 2018, pp. 187–190.

[5] G. He, M. Yang, J. Luo, and X. GU, ‘‘Inferring application type information from Tor encrypted traffic,’’ in Proc. 2nd Int. Conf. Adv. Cloud Big Data (CBD), Washington, DC, USA, Nov. 2014, pp. 220–227.

[6] A. Monteiro, D. Cuonzo, G. Ace to, and A. Escape, ‘‘Anonymity services tor, I2P, Mononym: Classifying in the dark (web),’’ IEEE Trans. Dependable Secure Compute., vol. 17, no. 3, pp. 662–675, May 2020.

[7] (May 2017). Wry Ransomware Analysis. Accessed: Apr. 26, 2023. [Online]. Available: https://www.secureworks.com/research/wcryransomware-analysis

[8] (Jul. 2019). Keeping a Hidden Identity: Miraa C&Cs in Tor Network. Accessed: Apr. 26, 2023. [Online]. Available: https://blog.trendmicro. Com/trendlabs-security-intelligence/keeping-a-hidden-identity-mirai-ccsin-tor-network/

[9] (Nov. 2014). Global Action against Dark Markets on Tor Network. Accessed: Aug. 4, 2020. [Online]. Available: https://www.europol. europa.eu/newsroom/news/global-action-against-dark-markets-tornetwork

[10] M. Juarez, M. Imani, M. Perry, C. Diaz, and M. Wright, ‘‘toward an efficient website fingerprinting defines,’’ in Proc. 21st Eur. Sump. Res. Compute. Secure. (ESORICS), Heraklion, Greece, Sep. 2016, pp. 27–46.

[11] T. Wang and I. Goldberg, ‘‘Walkie-talkie: An efficient defines against passive website fingerprinting attacks,’’ in Proc. 26th USENIX Secure. Sump. (SEC), Vancouver, BC, Canada, Aug. 2017, pp. 1375–1390.

[12] W. De la Cadena, A. Mitzva, J. Hiller, J. Penkyamp, S. Reuter, J. Filter, T. Engel, K. Where, and A. Pachinko, ‘‘Traffic Sliver: Fighting website fingerprinting attacks with traffic splitting,’’ in Proc. ACM SIGSAC Conf. Compute. Common. Secure. (CCS), New York, NY, USA, Nov. 2020, pp. 1971–1985.

Vol.15 No.3(2024),184-192
DOI: https://doi.org/10.61841/turcomat.v15i3.14789

[13] J. Hayes and G. Danzi's, ''k-fingerprinting: A robust scalable website fingerprinting technique,'' in Proc. 25th USENIX Conf. Secure. Sump. (SEC), Austin, TX, USA, Aug. 2016, pp. 1187–1203.

[14] X. Bai, Y. Zhang, and X. Nau, ''Traffic identification of Tor and web mix,'' in Proc. 8th Int. Conf. Intel. Syst. Design Appl. (ISDA), Kaohsiung, Taiwan, vol. 1, Nov. 2008, pp. 548–551.

[15] O. Berthold, H. Federate, and S. Koppel, ''Web Mixes: A system for anonymous and unobservable Internet access,'' in Proc. Int. Workshop Design Issues Anonymity Unobservability, in Lecture Notes in Computer Science, vol. 2009, H. Federate, Ed., Berkeley, CA, USA, Jul. 2000, pp. 115–129.