Analyzing Darknet Traffic: Examining how Tor Modifications Affect Onion Service Traffic Classification

K. Rama Aditya[1]; B. Sathyanarayana Murthy[2]; Dr. Chandramouli Venkatasrinivas Akana[3]
[1,2,3]Department of Computer Science and Engineering,
[1,2,3]BonamVenkataChalamayya Engineering College, Odalarevu, Andharapradesh

Abstract:- The important work of classifying network traffic for control and monitoring is examined in this study. Data protection has taken centre stage as privacy concerns have grown over the last two decades. Online privacy is possible through the Tor network, which is well-known for enabling Onion Services and offering user anonymity. But the abuse of this anonymity—especially with Onion Services—has prompted the government to work on de-anonymizing users. In this work, we address three main goals: first, we achieve over 99% accuracy in distinguishing Onion Service traffic from other Tor traffic; second, we assess how well our methods perform in the event that Tor traffic is modified to hide information leaks; and third, we detect the utmost significant article integrations for our classification task. This study tackles issues related to privacy challenges and misuse concerns in network traffic analysis.

Keywords:- *Dark Web, Traffic Analysis, Machine Learning, Network Security, Data Privacy, Feature Selection.*

I. INTRODUCTION

An obscurity network called Tor [1] uses a number of intermediate nodes to hijack traffic and conceal user identities. Additionally, Tor facilitates the deployment of anonymous services referred to as Onion Services (or not visible services).onion as the domain name at the top level. Security professionals, network defenders, and law enforcement organizations have been urged to distinguish traffic over Tor from different encrypted and unencrypted communication due to Tor's capacity to function as a forbiddance evasion tool [2], [3]. For instance, in [2], [5], and [6], the goal was to relegate the application types in Tor traffic, while in [3], [4], and [6], the goal was to relegate Traffic on Tor originating from other anonymity networks, including I2P and Web-mix traffic. In this study, however, we investigate the transit duration and discernibility of regular Tor traffic from Onion Service traffic using traffic analysis. Three research topics are mapped out to serve as the foundation for our effort. Initially, we attempt to address the query, RQ1: Is it feasible to downgrade other regular Tor traffic to Onion Service traffic? A standard Tor circuit, designed to utilize Tor to access an online service, consists of three Tor nodes. An onion service circuit consists of six Tor nodes and is the only way to access an onion service. Given that all of these circuits' communication—both the normal Tor and the Onion Service—is encrypted, we presume that we can distinguish between them using distinctive patterns found in the metadata, such as direction, timestamps, and packet size. In addition to hosting fraudulent websites, onion services have recently been utilized as botnet Command and Control (C&C) servers [7], [8]. Governments and law enforcement organizations thus seek to identify and shut down these services as well as control the flow of information on onion services [9]. Reducing access to these websites might be beneficial for businesses as well, since it can protect their systems from malicious actors and assaults.

Therefore, it can be beneficial to have methods for detecting Onion Service traffic for two major reasons: 1.These methods can serve as a foundation for Onion Service fingerprinting. 2. In delicate and private systems, they can be helpful in regulating Onion Service traffic. Secondly, we attempt to examine the same problem in non-identical environments. In particular, we aim to investigate RQ2: What is the holding power of our RQ1 results using changed Tor traffic? Tor may be configured to change its traffic patterns using certain techniques. A few of these strategies include adding padding [10], employing fake delays and bursts [11], and dividing the traffic [12]. In an effort to mask the information leaking of Tor traffic, these strategies have been expanded. Our ability to verify whether our RQ1 findings will be valid if and when these modifications are implemented to Tor traffic is the main advantage of answering RQ2. If these changes are implemented in the near future and we are able to identify traffic from Onion Service, it indicates that they are ineffective in hiding Onion Service activity. In the event that the adjustments have an impact on the Onion Service classifiability, it raises concerns regarding the validity of earlier studies, including [3] and [6,] while those modifications are in place. We contest that RQ2 is worthwhile to evaluate since its result may lead to other study directions on Tor traffic categorization.

II. WHAT IS DEEP LEARNING

Deep learning is a cornerstone of artificial intelligence (AI), and the present interest in deep learning might be partially attributed to the hype around AI. The capacity to allocate, observe, discern, and illustrate—that is, to understand—

has increased thanks to deep learning approaches [13]. Deep learning is used, for instance, to distribute photos, understand voice, identify objects, and provide content illustrations.

- *Deep Learning is now Under Danger from Several Developments:*

Advances in algorithms have validated the use of deep learning techniques.

Modern machine-learning techniques have improve model accuracy. Neural networks have developed into new classes that are ideally suited for tasks like picture classification and text translation.

We have access to additional data that we can use to construct neural networks with several deep layers, such as text from social media, medical notes, transcripts of investigations, and streaming data from the Internet of Things [14].

Thanks to advancements in graphics processing units and ubiquitous cloud computing, we now have access to ludicrous amounts of computer power. It takes this much processing power to train deep algorithms.

Human-to-machine interactions have also advanced significantly over this period. With gesture, swipe, touch, and natural language, the mouse and keyboard are reclaiming their place, heralding a resurgence of interest in artificial intelligence and deep learning [15].

- *How Deep Learning Works*

The way that you conceptualize and represent the issues that you are trying to solve using analytics is altered by deep learning. Teaching the machine how to solve a problem replaces the previous step of teaching the computer how to solve a problem.

A classic method to analytics involves first prioritizing an analytic model, then rating the constraints (or the unknowns) of that model, and then using the available data to create features to acquire new variables. Because the accuracy and completeness of a revenue prediction system rely on the model's look and quality, these strategies can improve underspecified systems [16]. For instances, when you develop a fraud model using feature engineering, you typically begin with a set of variables and finish up with a model that displays data transformations. Your model may require 30,000 variables in the end. After that, you will need to build the model and determine which of the variables are neccessary, which are not, and so on. You have to repeat the process if you want to add more data.

Reintroducing hierarchical characterizations (or layers) in the model's formulation and specification is the new deep learning strategy, which aims to extract latent characteristics from the data by using regularities in the layers [17]. The prototype shift is a gauge of the change in deep learning from feature engineering to feature representation. Deep learning has a duty to start producing prediction systems that are more energetic than those based on strict business rules and that can generalize, adapt, and improve over time as new data becomes available. You're not a model anymore. You train the task instead. Deep learning is having a significant effect on several sectors. Deep learning has applications in the biological sciences, including progressive image inquiry, drug development, illness symptom prediction, and stimulating innovations from heritable sequencing. It can help autonomous vehicles adjust to shifting conditions in the transportation industry [18]. It is also employed in the maintenance of agile response and vital infrastructure.

- *How Deep Learning Being Used*

From an external perspective, deep learning could seem like it's still in the research stage, as data scientists and computer scientists are testing its limits. But as research progresses, deep learning will find utility in a plethora of practical applications that corporations are already utilizing [19].

- *Recommendation Systems*

The idea of a recommendation system that can reasonably infer from your past behaviour what you might be interested in next has gained popularity thanks to services like Amazon and Netflix. Deep learning may be applied to strengthen suggestions in intricate contexts, such preferences for different types of music or clothes on many platforms.

Deep learning has progressed to the point that it can currently do tasks, such object recognition in photos, more accurately than humans [20].

III. LITERATURE SURVEY

A. *Classifying Tor Traffic Using Convolutional Neural Networks Based On Raw Packet Headers*

M. Kim and A. Anpalagan

Since network traffic is growing rapidly, effective resource allocation and network management depend heavily on traffic analysis and classification .But as security technologies advance,with encrypted communication like Tor, one of the most popular encryption techniques, this endeavor is getting harder. This paper proposes a convolutional neural network model and a hexadecimal raw packet header technique for categorizing Tor traffic.In line with competing machine learning methods, our approach demonstrates notable precision. To confirm this publicly, we use the UNB-CIC Tor network traffic statistics. The experiments show that our method provides 99.3% accuracy for the fractionalized Tor/non-Tor traffic classification.

B. *Improving Tor's Efficiency with Real-Time Traffic Categorization*

M. Al Sabah, K. Bauer, and I. Goldberg

Low-latency network Tor helps users protect their online privacy while preserving anonymity. Globally, hundreds of thousands of users are served daily by volunteer-run routers. A low relay-to-client ratio and congestion are two of Tor's frequent issues,which may deter users from adopting it widely and lead to a weaker global concealment for all users.

Our goal is to enhance Tor's performance by offering notable variations in service classes for its users. We note that while collaborative web surfing makes up the majority of Tor traffic, a comparatively small portion of majority downloading unfairly uses up all Tor bandwidth.

Furthermore, these traffic classes shouldn't receive the same Quality of Service (QoS) that Tor currently offers because they have different bandwidth and duration constraints.

We create and evaluate DiffTor, a machine-learning method that uses soliciting to classify Tor's encrypted circuits in real-time and then allocates different service levels to different applications.Our investigations validate that we can categorize circuits that we generate on the active Tor network with an accuracy of above 95%. We show that our real-time classification combined with QoS can greatly improve the familiarity of Tor users, as our straightforward methods yield an 86% reduction in download times at the moderate for interactive users and a 75% boost in responsiveness.

C. Online Anonymity Systems Such as Tor, I2P, And Jondonym: Identifying in the Shadows

A. Montieri, D. Ciuonzo, G. Aceto, and A. Pescapé

Categorization of traffic, which associates network traffic with the application generating it, is an essential tool for numerous applications across various domains (security, management, traffic engineering, R&D). This procedure is hampered by applications that encrypt communication material to preserve Internet users' privacy and anonymization tools that alter the communication's source, destination, and nature. In this work, we give (repeatable) classification findings using a publicly available dataset that was released in 2017. The goal is to assess the degree to which a certain anonymity technology is effective by comparing the traffic of different anonymity methods (and the traffic it conceals) can be recognized using machine learning approaches based only on statistical data. In order to do this, four classifiers—Naïve Bayes, Bayesian Network, C4.5, and Random Forest—are trained and evaluated using the dataset. The results show that Tor, I2P, and JonDonym, the three anonymity networks under examination, can be immediately recognized (with an accuracy rate of 99.99%), and that it is even possible to identify the specific application that is causing the traffic (with an accuracy rate of 98.00%).

D. Towards A Successful Defense Against Webpage Fingerprinting

M. Juarez, M. Imani, M. Perry, C. Diaz, and M. Wright

Website fingerprinting attacks allow a passive eavesdropper to enhance a user's otherwise anonymous web browsing by comparing the traffic that is observed to web traffic templates that have already been recorded. The defences that have been suggested to thwart these assaults are too costly in terms of additional latency and bandwidth consumption to be deployed in real-world systems. Furthermore, these defenses were designed to thwart assaults that were rejected for assuming unrealistic attack circumstances in the assessment environment, even if they had a high success rate. In this work, we aim to provide a distinct,lightweight defense against website fingerprinting, mostly in actual assessment situations, based on adaptive padding. This defense lowers the accuracy of the most advanced assault from 91% to 20% in a closed-world setting, all while adding zero latency overhead and less than 60% bandwidth overhead. The assault precision in an open environment is just 1% and decreases further with an increase in the number of locations.

IV. EXISTING SYSTEM

The mismanagement of online anonymity in the Tor network's Onion Services, which pretences privacy and security risks. This misappropriation has led to concerns, prompting governments and law enforcement agencies. Need to cultivate methods to accurately categorize Onion Service traffic and gauge the impact of potential obfuscation techniques on the distinguishability of such traffic, addressing concerns surrounding anonymity misuse and the need for effective traffic classification.

➢ *Disadvantages of the Existing System*

- Privacy risks due to misuse.
- Challenges in effective regulation.
- Potential for traffic misclassification.
- Continuous anonymity misuse concerns.

V. PROPOSED SYSTEM

A computer network entails of many protocols for data transmission and these protocols include HTTP, Voice Protocol, Email Protocol, and many more. These protocols are often vulnerable to attacks for data steal and to overcome this issue TOR protocol was announced which affords anonymity or security to user data. Tor which affords anonymity is called ONION Services. Often Onion services were used by illegitimate websites to avoid detection and perform malicious activities. Therefore in the recommended paper author applies machine learning algorithms to classify Tor Services as Onion Services or normal Tor services.

The author of the proposed study evaluated the effectiveness of a number of machine learning algorithms, including SVM, KNN, and Random Forest. The enactment of each method is estimated in expressions of FSCORE, accuracy, precision, and recall.

➢ *Advantages of Proposed System*

- Enhanced data security through Tor
- Protection from potential data theft
- Anonymity for user data transmission
- Machine learning aids classification

VI. FEASIBILITY STUDY

The viability of the project is assessed in this phase and business proposal is put up with a highly During system analysis the feasibility assessment of the proposed system is to be carried out. This is to make sure that the project's suggested overall plan and some cost projections are accurate. system does not burden the business. A basic understanding of the system's primary requirements is necessary for feasibility study.

VII. IMPLEMENTATION: SYSTEM DESIGN

A. Methodology

➢ *Importing Required Packages*

- Utilize Python classes and packages for data analysis, such as pandas for data handling and matplotlib for visualizations.

➢ *Loading and Displaying Datasets*

- Load the original Tor dataset and visualize its content.
- Load class labels for the Tor dataset and then load and display the WTFPAD dataset alongside these labels.

➢ *Feature Selection*

- Use Information Gain algorithm to select 50 relevant features from the dataset.
- Display a graph depicting the importance scores of these selected features.

➢ *Feaftures Importance Calculation*

- Compute feature importance using Correlation Coefficient and Fisher Score methods.

➢ *Dataset Processing*

- Perform necessary data processing steps like shuffling and splitting the dataset into training and testing subsets.

➢ *Metrics Function Definition*

- Define a function to calculate accuracy and other evaluation metrics for the classification models.

➢ *Model Training and Optimization*

- Train models like KNN, Random Forest, and SVM on the original No Defence dataset.
- Optimize the KNN model using hyperparameters to improve accuracy.

➢ *Model Evaluation on WTFPAD Dataset*

- Assess the performance of trained models (KNN, Random Forest, SVM) on the modified WTFPAD dataset and observe the impact on accuracy compared to the original dataset.

➢ *Merging Datasets and TOR/Onion Classification*

- Merge the WTFPAD and Onion Services datasets for TOR classification.
- Train KNN, SVM, and potentially other models on the merged dataset and evaluate their accuracy using confusion matrices and other visualizations.

➢ *Algorithm Performance Evaluation*

- Display a table summarizing the performance of all algorithms in classifying Tor and Onion Services, showcasing high accuracy levels.

➢ *Test Data Classification*

- Read test network data and employ the ADABOOST algorithm to classify network traffic as Tor or Onion services, showcasing predicted service types.

B. System Architecture

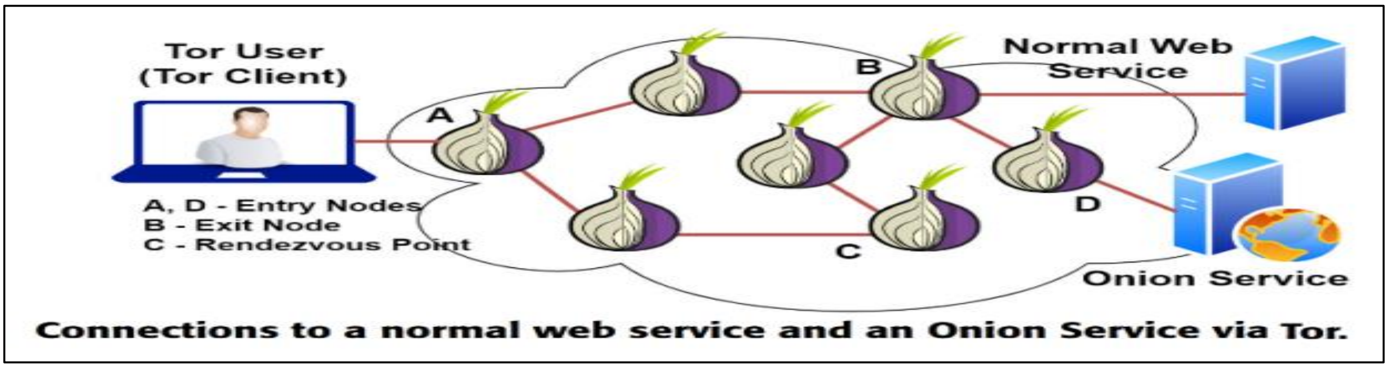

Fig 1: System Architecture

VIII. RESULT

➢ *Output*

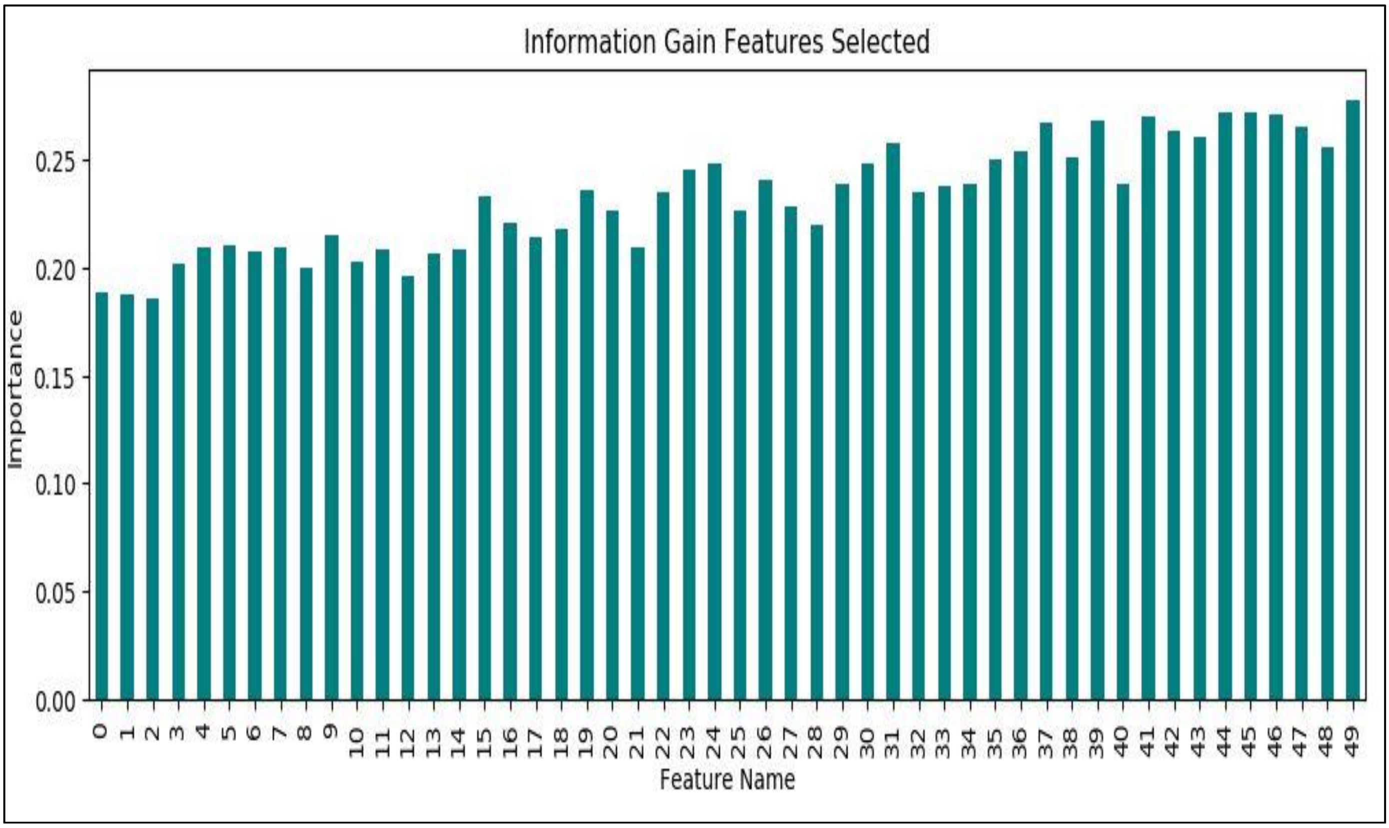

Fig 2: The x-Axis in the Graph above Denotes the Names of the Features, while the y-Axis Denotes their Significance or Pertinent Score Value.

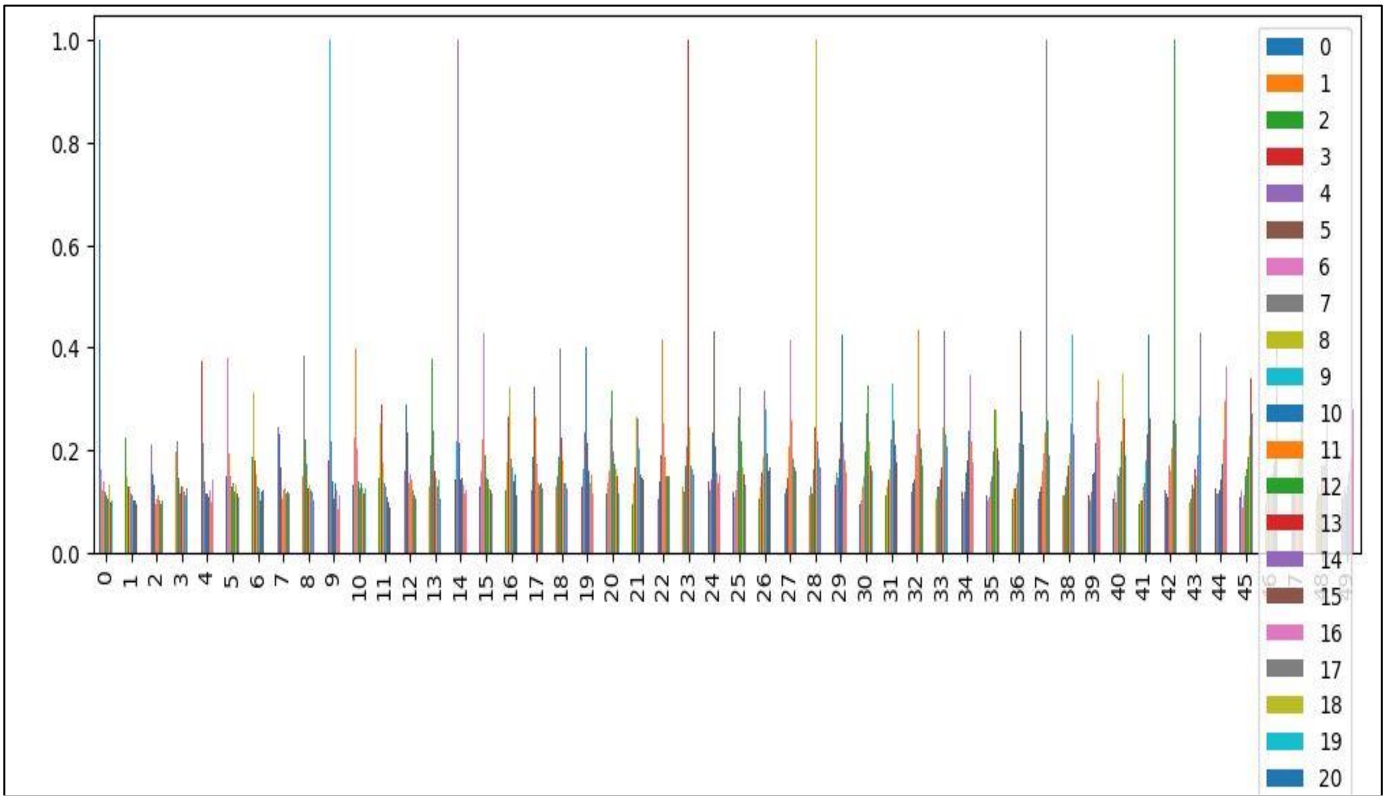

Fig 3: In the above Figure Calculating Features' Importance using the Correlation Coefficient

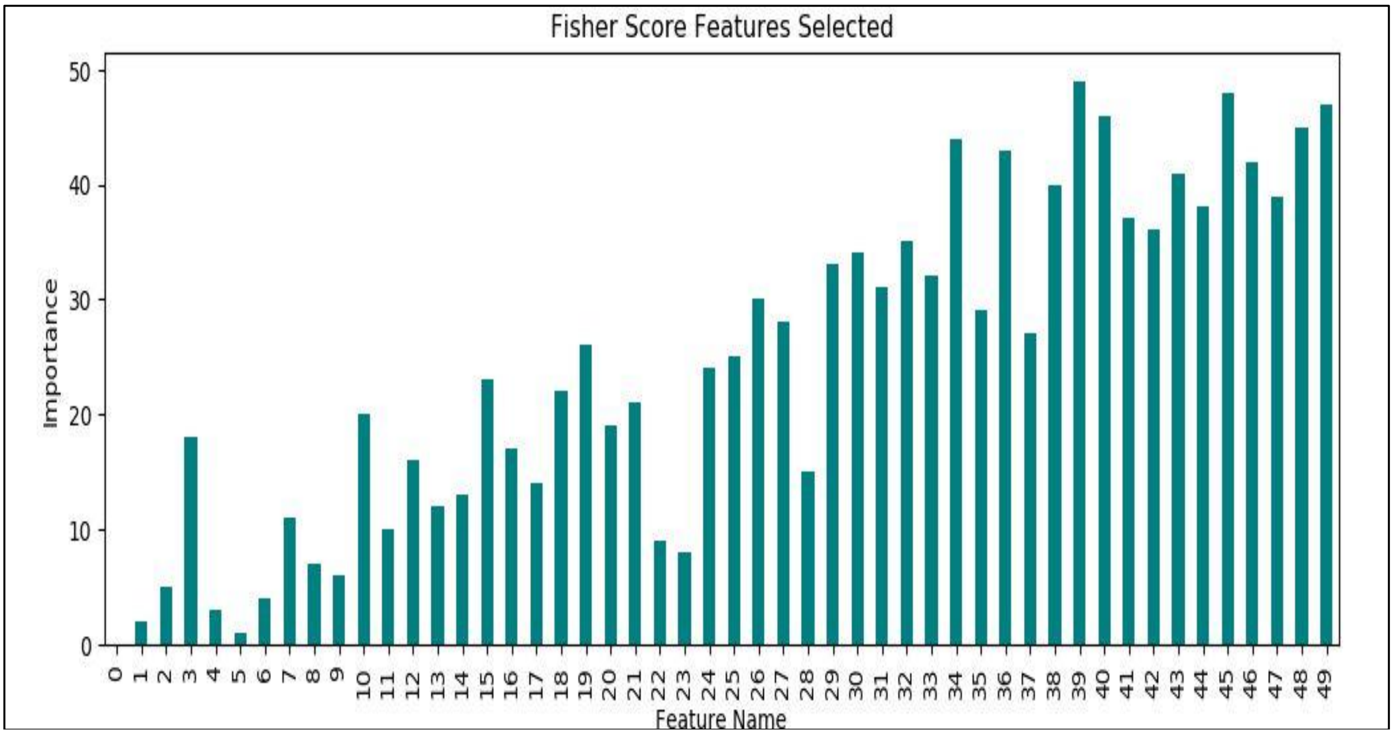

Fig 4: In the above Figure Calculating Features' Importance using the Fisher Score

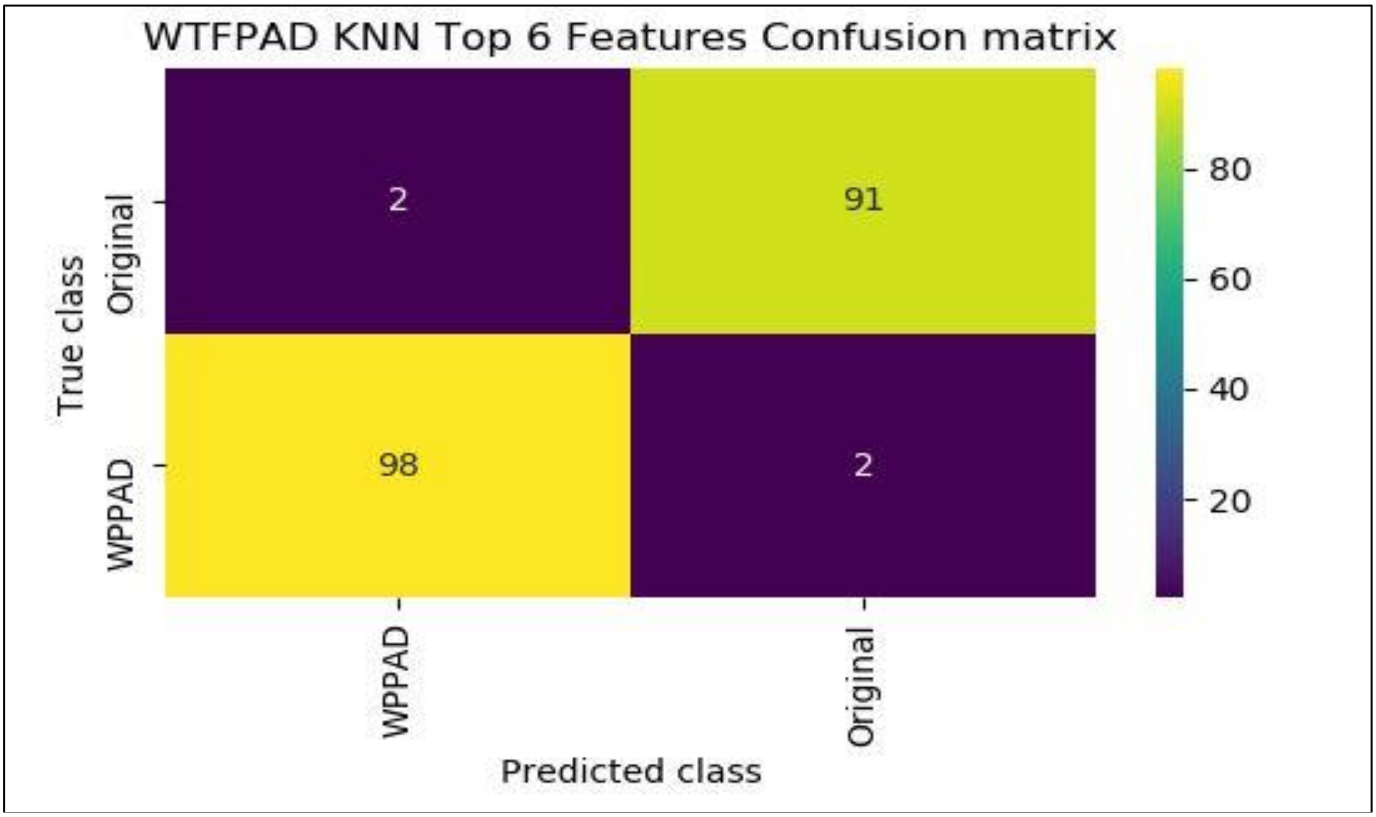

Fig 5: In the above Figure Training KNN on Merged TOR and OS TOP 6 Features Selected Dataset and then after Training KNN got 97.92% Accuracy

In the confusion matrix graph, the expected Original OS service label and WTFPAD service label are shown on the x-axis and the y-axis signifies true labels. Yellow and light green boxes contain correct prediction count and blue boxes contain incorrect prediction.

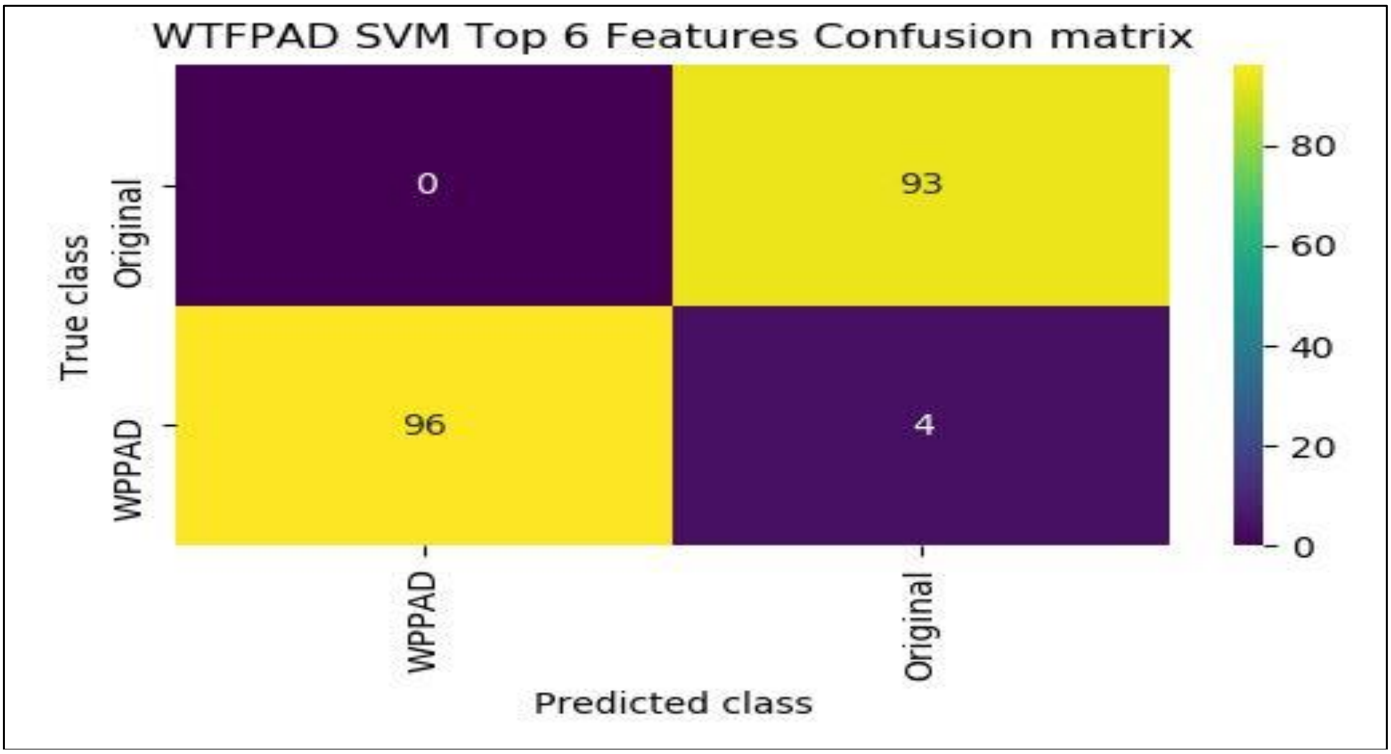

Fig 6: In the above Figure SVM got 97.92% Accuracy

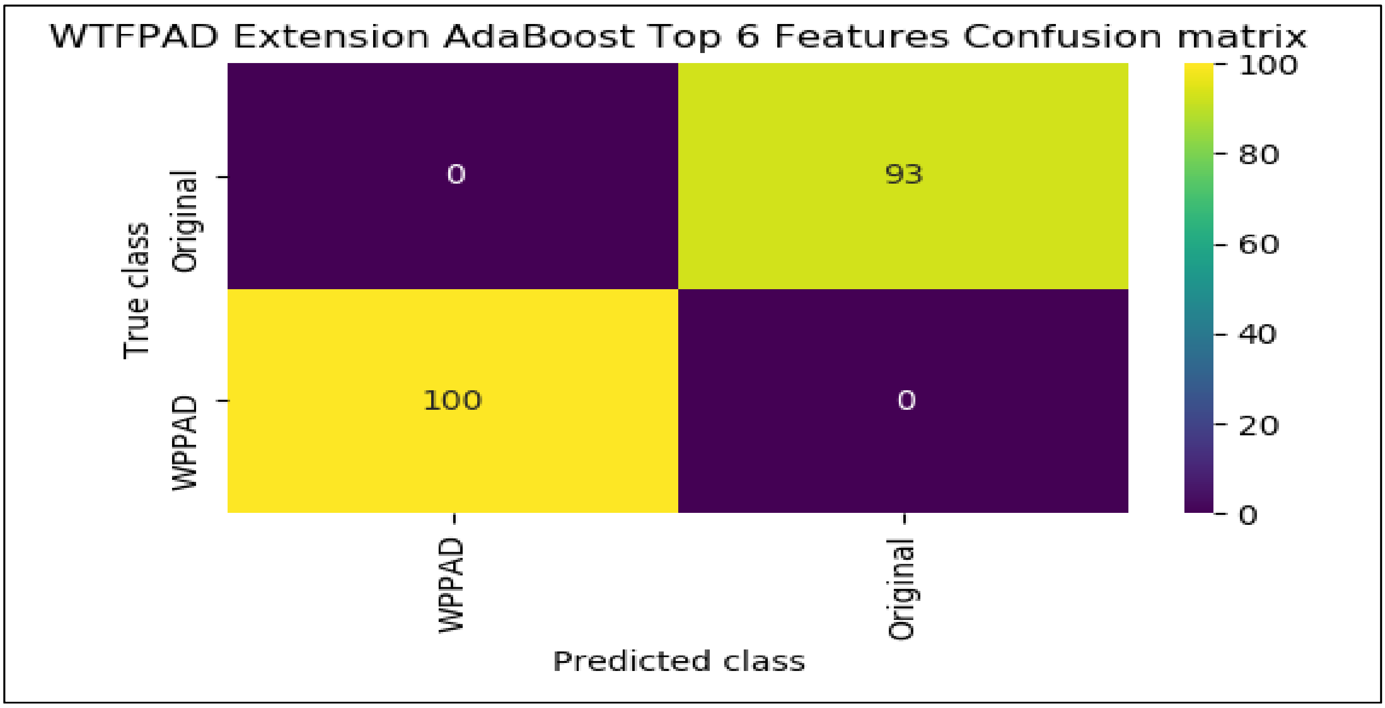

Fig 7: In the above Figure Training Extension Adaboost Algorithm and then after the Training Extension got 100% Accuracy

IX. CONCLUSION

We calculated how well supervised machine learning models would work for sorting Tor traffic from Onion Service traffic. Each traffic trace had fifty characteristics that we mined, and we sent that feature set into the machine learning classifiers. Our results shown that 100% accuracy can be achieved in differentiating between Tor and Onion Service traffic using KNN, RF, SVM, and AdaBoost classifiers. Next, we attempted to classify whether or not modern Website Fingerprinting defenses affect Tor traffic's capacity to be classified. We evaluated how these defenses affect the Onion Service traffic categorization by introducing various modifications in an attempt to impede information leakage from traffic. Our tests showed that the performance of Onion Service traffic categorization is reduced when the aforementioned classifiers are combined with our feature collection.

REFERENCES

[1]. R. Dingledine, N. Mathewson, and P. Syverson, ‘‘Tor: The second-generation onion router,’’ in Proc. 13th USENIX Secur. Symp. (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. Comput. Com-mun. Secur. (CCS), New York, NY, USA, Oct. 2012, pp. 73–84.

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

[4]. M. Kim and A. Anpalagan, ‘‘Tor traffic classification from raw packet header using convolutional neural network,’’ in Proc. 1st IEEE Int. Conf. Knowl. Innov. Invention (ICKII), Jeju 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. Montieri, D. Ciuonzo, G. Aceto, and A. Pescapé, ‘‘Anonymity services tor, I2P, JonDonym: Classifying in the dark (web),’’ IEEE Trans. Depend-able Secure Comput., vol. 17, no. 3, pp. 662–675, May 2020.

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

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

[10]. M. Juarez, M. Imani, M. Perry, C. Diaz, and M. Wright, ‘‘Toward an efficient website fingerprinting defense,’’ in Proc. 21st Eur. Symp. Res. Comput. Secur. (ESORICS), Heraklion, Greece, Sep. 2016, pp. 27–46.

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

[12]. W. De la Cadena, A. Mitseva, J. Hiller, J. Pennekamp, S. Reuter, J. Filter, T. Engel, K. Wehrle, and A. Panchenko, ‘‘TrafficSliver: Fighting web-site fingerprinting attacks with traffic splitting,’’ in Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS), New York, NY, USA, Nov. 2020, pp. 1971–1985.

[13]. J. Hayes and G. Danezis, ‘‘k-fingerprinting: A robust scalable website fin-gerprinting technique,’’ in Proc. 25th USENIX Conf. Secur. Symp. (SEC), Austin, TX, USA, Aug. 2016, pp. 1187–1203.

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

[15]. O. Berthold, H. Federrath, and S. Köpsell, ‘‘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. Federrath, Ed., Berkeley, CA, USA, Jul. 2000, pp. 115–129.

[16]. B. Zantout and R. Haraty, ‘‘I2P data communication system,’’ in Proc. 10th Int. Conf. Netw. (ICN), Sint Maarten, The Netherlands, Jan. 2011, pp. 401–409.

[17]. P. Sirinam, M. Imani, M. Juarez, and M. Wright, ‘‘Deep fingerprint-ing: Undermining website fingerprinting defenses with deep learning,’’ in Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS), Toronto, ON, Canada, Oct. 2018, pp. 1928–1943.

[18]. R. Overdorf, M. Juárez, G. Acar, R. Greenstadt, and C. Díaz, ‘‘How unique is your.onion?: An analysis of the fingerprintability of Tor onion services,’’ in Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS), Dallas, TX, USA, Oct. 2017, pp. 2021–2036.

[19]. I. H. Witten, E. Frank, and M. A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann, 2011.

[20]. X. He, D. Cai, and P. Niyogi, ‘‘Laplacian score for feature selection,’’ in Proc. Adv. Neural Inf. Process. Syst. (NIPS), Vancouver, BC, Canada, Dec. 2005, pp. 507–514