Examination of QUIC-based Website Fingerprinting

Pratiksha Narasimha Nayak G
Dept. of CSE
RV College of Engineering
Bangalore, India
email:
pratikshanng.cs20@rvce.edu.in

Nimisha Dey
Dept. of CSE
RV College of Engineering
Bangalore, India
email:
nimishadey.cs20@rvce.edu.in

Neha N
Dept. of CSE
RV College of Engineering
Bangalore, India
email:
nehan.cs20@rvce.edu.in

Malavika Hariprasad
Dept. of CSE
RV College of Engineering
Bangalore, India
email:
malavikah.cs20@rvce.edu.in

Minal Moharir
Dept. of CSE
RV College of Engineering
Bangalore, India
email: minalmoharir@rvce.edu.in

Sandhya S
Dept. of CSE
RV College of Engineering
Bangalore, India
email: sandhya.sampangi@rvce.edu.in

***Abstract*— Recently, there has been a considerable amount of interest in extending QUIC for new capabilities. Since QUIC is governing most of the internet traffic, questions arise about its security. In this regard, the paper aims to understand the procedure of website fingerprinting in the case of QUIC websites as compared to TCP websites. It aims to provide a comparison of the effectiveness of the earlier used ML methods on the TCP traffic and QUIC traffic. Through this paper, a thorough comparative study of various machine learning models has been incorporated to identify the website to which a packet belongs to. In this paper, Wireshark has been used to capture packets from 50 different websites that use QUIC using a Perl script. The paper presents the results of training the data on a combination QUIC and TCP packets using the commonly used machine learning models p-FP, Var-CNN, CUMUL, Wang-KNN and k-FP. Results show that Random forest classification gives the best accuracy whereas k-FP gives the least. It can also be noted that the overall efficiencies of all the algorithms are much lesser on the combination dataset as compared to datasets with only TCP traces. This can be attributed to the fact that these models have been designed for TCP data, and are unfamiliar with QUIC traffic specifications.**

Keywords— QUIC, Website Fingerprinting, Machine learning, CNN, Random forest, k-FP, CUMUL.

I. Introduction

QUIC (Quick UDP Internet Connection) solves a number of transport layer and application layer problems experienced by modern web applications with no significant changes. QUIC is very similar to TCP, TLS and HTTP/2 integrated but implemented on top of UDP. The introduction of this novel transport protocol has allowed innovations which aren't possible with existing protocols as they are hampered by legacy clients and middleboxes. Currently Google provides server support for its properties and are gradually migrating all eligible clients to QUIC with the end goal of offering all Google services over QUIC.

For network operators and administrators, identification of traffic categories generated by different applications and protocols helps them in providing high Quality of Service (QoS) for the network users. As a network operator, there are a few things to consider:

- UDP unicast traffic should be allowed to flow unimpeded by filters or rate-limiters.
- Hash functions used for link aggregation or traffic balancing should include UDP transport headers to ensure sufficient entropy.
- NATs should be configured for UDP unicast traffic according to documented best practices, e.g., RFC 4787.

Traffic classification can be used to monitor user behaviours and predict traffic categories to assist network management. Distinguishing abnormal network traffic is also extremely important for intrusion detection and network security measurement.

Since TCP is implemented in operating system kernels and middleboxes, it is nearly impossible to make significant changes to TCP on a large scale. However, because QUIC is based on UDP and the transport mechanism is encrypted, it is not constrained in this way. Hence, with the switch to QUIC as the primary transport layer protocol may happen very soon. With this arises questions about the security of QUIC. Data has become one of the most susceptible assets shared on networks in the modern cyber world. For a new protocol, it is essential to understand its vulnerabilities and strengths. In this regard, the paper aims to provide a comparison on the various fingerprinting models like p-FP, Var-CNN, CUMUL and k-FP and present the result.

II. Literature Survey

The existing approaches to develop the framework needs extensive survey. Survey report, initial dataset, set up of packet processing tool, identification of machine learning techniques and initial set up for building the diversified dataset as well as survey of QUIC, and TCP traces for joint QUIC-TCP classification. A detailed survey of existing traffic classifiers, using the features and the process of identification of application layer protocols are studied and documented.

In 2016 Muhammad Shafiq et al. proposed various techniques for analysing and classifying network traffic based on machine learning algorithms. The traditional methods of classifying network traffic are Port Based, Payload Based. Network Traffic Classification Techniques

and Comparative Analysis Using Machine Learning Algorithms [1] studies and contrasts between the different methods of traffic detection and provides an overview of the disadvantages of such traditional methods when it comes to P2P (peer- to-peer) applications which use dynamic port numbers and their ineffectiveness against networks that implement QUIC and TLS 1.3 which have very few unencrypted fields during transmission. Thus, ML based techniques are currently the most preferred method for traffic classification. The work also provides a comparison between different machine learning algorithms in the classification of packets, and finds that C4.5 classifiers obtain the highest accuracy.

Pasyuk A., Semenov E., Tyuhtyaev D in 2019 classified network traffic based on feature selection. Feature Selection in the Classification of Network Traffic Flows [2] talks about the need to classify flows and a short description on the various proposed traffic classification methods. It speaks of the concept of a "flow" and lists the 37 key features out of the 246 statistical features to be used for traffic classification. The paper also highlights the steps involved to clean the dataset captured from applications such as Wireshark, tcpdump packet analysers and other open sources. It has described the four types of sequential selection methods and shown the calculations to select a specific number of features (here 10) out of the 37 for traffic classification using KNN, Random Forest and Gradient boosting ML models. The paper concludes with sequential forward selection methods being suitable for training network flow classifiers based on the analysis obtained from line plots and bar graphs.

V. Tong, H. A. Tran, S. Souihi and A. Mellouk ,in A novel technique for traffic classification based on the convolutional neural networks [3], in 2018 proposed a novel QUIC classifier by integrating the feature extraction unit and the convolutional neural network (CNN) unit. The paper discussed some of the challenges that the current classifiers face that the QUIC-based traffic poses due to its reduced visibility. Hence, the paper proposes a more novel flow static-based method based on CNN that is used to detect different QUIC based services by incorporating flow and packet-based features to improve the performance. The proposed method was successfully able to detect some kinds of QUIC-based services such as Google Hangouts, YouTube, File transfer, etc.

Noora Al Khater; Richard E Overill in 2015 have discussed challenges and techniques for Network Traffic Classification. Classifying network traffic links network traffic with a generated application, is a vital first step for network analysis. It is the core element of network intrusion detection systems(IDS) especially for security purposes such as filtering traffic and identifying and detecting malicious activity. The application of Machine Learning (ML) algorithms in several classification techniques [4], utilising the statistical properties of the network traffic flow is described. The paper begins with a detailed note on the various proposed methods to classify the network traffic - port-based, payload-based, statistical and behavioural classification. It then discusses the challenges in classification of the traffic using ML models - such as due to higher traffic flows, more complicated ML models leading to high computational costs, trend in the encryption of data packets, etc. It highlights the previous work done in traffic classification using various ML models and algorithms like Expectation Maximisation and concludes that the use of Supervised models to train and of the Unsupervised models to detect new applications in the traffic is one of the best propositions. The paper then highlights the steps involved in traffic classification with sub-flows for faster recognition and timely detection.

In 2020 Shahbaz Rezaei, Xin Liu have proposed Multitask Learning for Network Traffic Classification [5] is proposed To mitigate the need for a large amount of labelled training samples. This paper involves the creation of a multi-task model where bandwidth requirement and duration of a flow are predicted along with the traffic class. It also illustrates that with a large amount of easily obtainable data samples for bandwidth and duration prediction tasks, and only a few data samples for the traffic classification task, one can achieve high accuracy. The input to the model contains the columns time-series, header, payload, and statistical features. The model is trained using CNN. The results show that the accuracy of the traffic class prediction, even with limited labelled samples, is considerably higher with the multi-task learning approach than the transfer learning and single-task learning.

In 2021, Yu Guoy, Gang Xiongy, Zhen Liy, Junzheng Shiy, Mingxin Cuiy, Gaopeng Gouy proposes an end-to-end framework for classifying network traffic employing Generative Adversarial Network (GAN) architecture. GANs are generally used to achieve high overall accuracy without compromising the class balance. The paper [6] proposes a novel GAN architecture that incorporates a classifier which makes the dataset generation process more stable and effective. It provides a comparison some of the oversampling methods like ADASYN, ROS, SMOTE etc.

Maohua Guo, Jinlong Fei, Yitong Meng in 2021 proposed a deep learning model for website fingerprinting in their paper titled ‘Deep Nearest Neighbour Website Fingerprinting Attack Technology’[7]. The paper compares the deep learning model with the older triple fingerprinting model. It also highlights the importance of local fingerprint features along with other general features of website traffic. The deep fingerprinting model uses local fingerprinting features extracted using CNN or convolutional neural networks. It then feeds these features to a K-nearest neighbours or KNN model. The paper also claims that the model has better

performance and is almost immune to concept drift problems.

In 2021, Jean-Pierre Smith, Prateek Mittal, and Adrian Perrig in the Proceedings on Privacy Enhancing Technologies have discussed website fingerprinting. QUIC protocol is harder to fingerprint than TCP because QUIC uses encryption. Website Fingerprinting in the Age of QUIC [8] investigates the differences in the importance of features from a packet for the two transport layer protocols. Website fingerprinting refers to an observer trying to identify a visited website over an encrypted traffic based on the physical (side-channel) information derived from features like packet sizes and timestamps. The paper gives the idea of splitting a dataset of traffic packets from various websites into monitored and unmonitored web-pages. The authors have adapted the k-fingerprinting (k-FP), Deep fingerprinting (DF), p-FP and Var-CNN classifiers for the same. The paper finds that both QUIC and TCP traces can be fingerprinted easily. However, a model for the classification of a mixer of packets of QUIC and TCP must be developed.

Vera Rimmer, Davy Preuveneers, Marc Juarez, Tom Van Goethem, Wouter Joosen in 2017 proposed a novel method based on deep learning. The proposed classifier combines feature extraction with the training process, thus allowing it to classify the packets based on the way it is initially portrayed. Among the existing Deep Neural Networks, three different types of neural networks have been evaluated-feedforward, convolutional and recurrent i.e, Stacked Denoising Autoencoder (SDAE) which is a deep feedforward neural, Convolutional Neural Network (CNN) and Long-Short Term Memory network (LSTM) [9]. They performed semi automatic hyperparameter tuning to evaluate each hyperparameter's impact. They also discussed a concept called concept drift which involves a decrease in the accuracy of the classifier due to change in the parameters the classifier is using to predict the outcome. For a closed world with 100 websites, the achieved success rate is over 96%, and for our largest closed world with 900 classes, it is over 94%. The most effective deep learning model outperformed the most advanced attack by 2% accuracy in our open world study.

Yong-Jun Wei, Su-Juan Qin in 2016 implemented a web fingerprinting attack based on SSH anonymous communication using random decision forest classifier [10]. They paid more attention to the outgoing traffic sent from server to the client. The model is stable even when the training set is updated frequently.

Various reputed national level institutions and government labs like CAIR, DRDO, IISC, CDAC etc are working rigorously on network traffic analysis. Most of the work published at national level is about basic understanding of QUIC protocol and its performance analysis with existing protocols. There are few papers at international level which have done characterisation of encrypted traffic to extract domain name and other features. With the baseline of above surveyed we proposed an automated framework to identify QUIC traffic and fingerprinting of the website the user has visited. Further work is extended to simulate drift in data by changing input pattern and dataset characteristics. It automates the identification of variation/drift in Data and analyses the degradation in the performance of the model. The automated framework can be implemented by CNN/GAN based deep learning models to efficiently handle Drift in Data.

III. Contribution of the Paper

In light of the rapid adoption of QUIC, this paper aims to:

1. To understand the concept of web fingerprinting.
2. To analyse effectiveness of fingerprinting methods on QUIC packets captured from different sources.

IV. Methodology

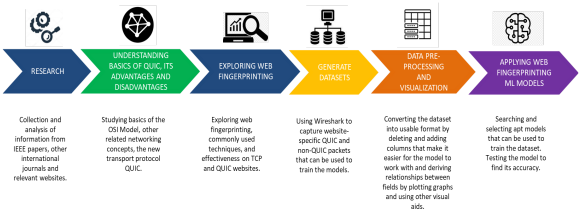

Fig 4.1: Methodology followed

The methodology of the research is depicted in Fig 4.1 and has been explained in this Section in detail.

4.1. Data generation and Repository creation for data sets

4.1.1. Data generation for various combinations and capture as pcap

Publicly available HTTPs data can be gathered by crawling top-accessed HTTPs websites on browsers like Google Chrome, Mozilla Firefox etc., which gives the raw packet capture (pcap) files. The information from the captured file can be extracted using a variety of methods so as to obtain packets of a specific protocol. The most popular options for directly extracting PCAP format files are Wireshark, TCP extract, TCP dump, Pick-Packet, and Network-miner.

The test bed used for the pcap capture includes various OS and web browsers to diversify the data captured as follows:

OS: Microsoft Windows 10, Linux

Web browsers: Google Chrome, Mozilla Firefox.

In this paper, Wireshark has been used to capture packets from 50 different websites that use QUIC using a Perl script. In the script, code was written to open the 50 websites one by one and capture packets for a fixed period of 30 seconds.

4.1.2. Repository creation for data set

The packet data can be exported as a CSV file for the dataset and stored on a remote cloud repository for further use and enhancements. The proposed setup is as shown in

figure 1.1. A python script was written to convert the pcap files to csv and extract the required fields.

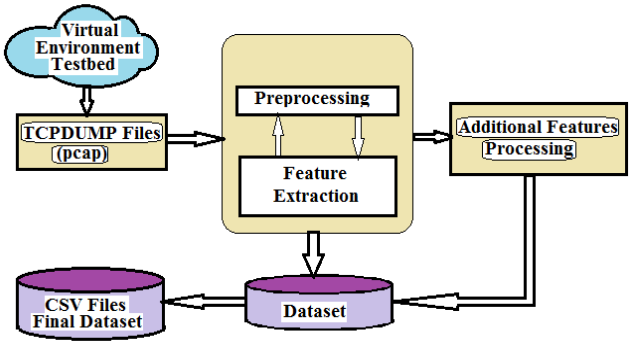

Figure 1.1 A scheme for datasets creation

4.2. Set up of traffic processing tool for processing pcaps and extracting features or payload portion

The proposed solution needs to handle diversified traffic hence a python script has been used in conjunction with a primary application which has the packet capture framework initialised already. The script converts the pcap files into csv files by extracting the required features from the captured packet data. Columns are labelled and the csv file is saved as '[1-50].csv' where 1-50 represents the corresponding website number of the site in the list used to refer to it. It will help to extract specific features which can be used for further characterisation. The statistical feature generation and pre-processing is implemented in this step.

4.3. Characterization of Encrypted QUIC traffic

4.3.1. Set up of Machine learning/Deep learning framework

The generated data in step 3 can be analysed using the ML/DL framework. As per the literature, methods used can be p-FP, k-FP, CUMUL, Random Forest Classification, Wang-KNN or Var-CNN.

The network traffic classification structure model includes step by step process as shown in Figure 1.2. The network traffic capture is explained in point 2 (above) – Data generation and Repository creation for dataset.

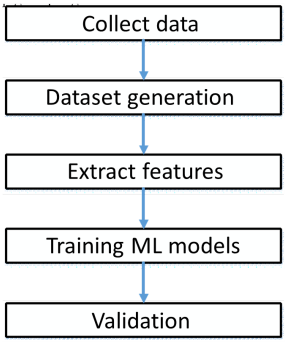

Figure 1.2: Network traffic classification model

4.3.2. Features Extraction Selection focuses on the features that are extracted from the captured data

The typical features employed in traffic classification literature for modelling traffic can be categorised into the following groups:

i. Flow Statistics: Measures of central tendencies of features such as TCP flag counts. Flow durations, number of packets, number of bytes, packet lengths, etc.
ii. Raw Bytes: The number of bytes in the header and the payload aggregated.
iii. Time-series: This refers to the data of a flow that can be recorded over consistent intervals of time. For example, the size of packets in a flow is a valid time-series feature.

The final features extracted and chosen after preprocessing of data are elaborated in the Table below:

Table 1: Features extracted to train the ML models

Column	Description
frame.encap_type frame.time_epoch frame.len frame.cap_len	Frame data
ip.version	Version of IP
ip.hdr_len	Length of the IP header
ip.flags.rb ip.flags.df ip.flags.mf ip.frag_offset	Used in the fragmentation process
ip.ttl	Time to live (TTL) is used to avoid packets moving in the network endlessly.
ip.proto	Identifies the protocol used
ip.len	Length of the IP packet
tcp.hdr_len	Length of the TCP header
Website	Name of the website

4.3.3. Training machine learning model on extracted features

i) Import the required libraries and modules - Pandas, Numpy, Keras and Sklean
ii) Read the csv files containing the captured packets' data into a dataframe.
iii) Append the website number to the corresponding csv files - Each csv file represents a website.
iv) Drop the features that are insignificant to train the model- such as the ethernet source and destination addresses, the IP source and destination addresses, IP checksum, TCP or UDP source and destination addresses - and shuffle it. The features have been scaled to increase the accuracy.
v) The results column i.e., name of the website (Y) is to be separated from the rest of the features in the dataframe (X).
vi) Using the train_test_split function in the Sklearn library, the test and the train data are obtained in an 8:2 ratio. In this stage, data sets are sampled, data are first labelled to classify unknown network applications. The standard ML tools/libraries can be implemented in Python: Sci-kit learn

library, some methods from NumPy, and DataFrame from Pandas have been used.

4.3.4. Implementation of Machine Learning Algorithms

This is the implementation step which includes applying machine learning algorithms or classifiers on the instances such as applying supervised, unsupervised and semi supervised learning algorithms. In this paper, the implementation has been done on Python using Google Colab. A brief description of the models implemented have been given below:

p-FP: p-FP is a deep learning model consisting of an input layer. One or more convolutional layers are followed by one or more fully connected layers to create a CNN. Convolution, pooling, and classification are the three series of operations that the forward propagation conducts. The convolution operation extracts features from the input layer using filters which can be slided across the width and height of the input layer to form a 2D feature map. These feature maps are used to learn about the pattern. Next, pooling is done to reduce the size of feature mas generated so that the computation required can be reduced. The features learned by the convolutional and pooling layer are fed into the model for classification. This model gave an accuracy of around 70%.

Var-CNN: Var-CNN is a deep learning based website fingerprinting attack. It uses ResNets and convolutional neural networks as its base model. Additionally, it has insights specific to network classification. It uses a combination of manually and automatically extracted features. Timing and direction related features are found to be beneficial to the model. This model was found to have a 1% higher true positive rate as compared to prior models used for the same.

Wang-KNN: Wang-KNN is a supervised learning model that uses distance metrics (Euclidean or Manhattan) to predict the website to which a packet belongs to. While training the model, the dataset is split into test and train sets and distances between the testing and all the training points is determined. The distances are then arranged in the ascending order and the top k rows are selected to determine its class. The class (or website) of the packet is obtained by majority voting or with the nearest neighbour of the testing point.

LSTM: It is a deep learning technique mainly consisting of a series of cells whose output depends on three parameters - the cell state, the hidden state and the current input to the LSTM cell. It is a feed forward network that involves the retention of selective data from the previous state to filter the information required to be fed to the next state. LSTM makes small modifications on the data by simple positional additions and multiplications. In this way, the model forgets and remembers data selectively, which makes it more efficient than Recurrent Neural Networks (RNNs).

CUMUL: CUMUL is a website fingerprinting attack based on Support Vector Machine (SVM). The kernel is the set of mathematical functions used to convert the linear data into high dimension spaces. For CUMUL, the kernel used is Radial Basis Function or RBF. It derives the cumulative length and uses this feature to train the model. It requires a large amount of computation time and resources to complete training. Hence, we only passed a small subset of the dataset to this model.

Random Forest Classifier: Random Forest is an ensemble method involving a number of decision trees. It can also evaluate the importance of the features. Even if a significant amount of data is missing, random forest was able to approximate the missing data and maintain accuracy.

k-FP: The k-fingerprinting classifier integrates two ML models - Random Forest and k-nearest neighbours. The Random Forest technique helps in determining the contribution of various features for website fingerprinting. This is useful to understand which features of the network traffic leak the most information about which website is being accessed by the user. Some of the most important features of the network traffic that helps in website fingerprinting are volume information and timing information. The k-nearest neighbours algorithm classifies the website. If the attacker's model was trained for the website, the sample website is classified into one of the 'monitored' websites. Else, in case of any ambiguity, it is classified as an 'unmonitored' website.

V. Results and Discussion

After applying the ML algorithm, validation of the result has been done using the validation or testing portion of the dataset. The table below gives the accuracy of the different models tested in the paper.

Table 2: Accuracy for models used in the training

Number of Packets	Model	Accuracy (in %)
583281	Var-CNN	45
583281	p-FP	70
583281	Wang-KNN	83.71
583281	LSTM	19.30
150000	CUMUL	11.65
583281	Random Forest	96.89

As noticed, Random forest classification gives the best accuracy whereas k-FP gives the least. This is due to the fact that only a small fraction of the original dataset is used to train k-FP so sufficient data is not obtained. The figure below also gives a graphical representation of the different models and their accuracies.

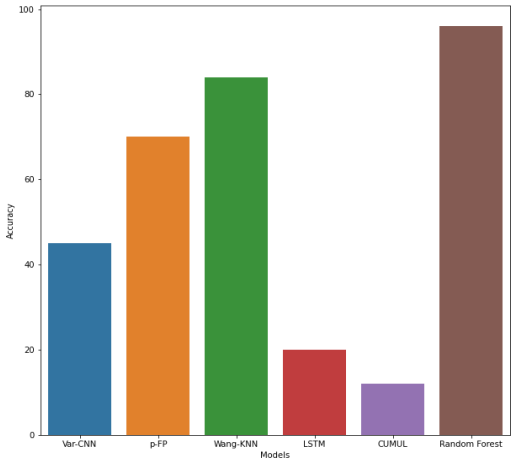

Fig 2: Graphical representation of results

VI. Conclusion and future scope

A website fingerprinting attack uses patterns of data flows, such as packet size and direction, to determine the content of encrypted and anonymous connections. This poses a huge threat to the user's privacy. Although VPNs try to reduce this type of attack, they are still vulnerable to it. In TCP networks, such attacks have high efficiency and much research has been done on both, fingerprinting as well as defence methods. However, when it comes to QUIC traffic, Website Fingerprinting is still a relatively unexplored threat. Therefore, this paper presents a thorough examination into the efficiency of commonly used fingerprinting techniques that have high efficiency on TCP traffic, on sites that use QUIC.

The dataset used contains both TCP and QUIC packets. The results of the study demonstrate that although these methods are very effective on TCP traffic alone, they give lower accuracy on TCP plus QUIC traffic. As a result, it can be inferred that using the same methods of fingerprinting is not sufficient. Attackers would have to develop new methods or train the current models extensively with QUIC data and combination data with QUIC and TCP packets to attain the same accuracy.

In the future, training with a larger amount of data and including the direction of each trace in the dataset can give a higher accuracy. Due to computational limitations, the k-FP and CUMUL models were run with a subset of the used dataset. Increase in the size of the training dataset would guarantee a better accuracy. Since QUIC is governing most of the internet traffic, attackers will quickly put more resources into improving fingerprinting methods. To resist this, research is needed to develop defence methods specific to QUIC traffic.

VII. References

[1] M. Shafiq, X. Yu, A. A. Laghari, L. Yao, N. K. Karn and F. Abdessamia, "Network Traffic Classification techniques and comparative analysis using Machine Learning algorithms," 2016 2nd IEEE International Conference on Computer and Communications (ICCC), 2016, pp. 2451-2455, doi: 10.1109/CompComm.2016.7925139.

[2] A. Pasyuk, E. Semenov and D. Tyuhtyaev, "Feature Selection in the Classification of Network Traffic Flows," 2019 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon), 2019, pp. 1-5, doi: 10.1109/FarEastCon.2019.8934169.

[3] V. Tong, H. A. Tran, S. Souihi and A. Mellouk, "A Novel QUIC Traffic Classifier Based on Convolutional Neural Networks," 2018 IEEE Global Communications Conference (GLOBECOM), 2018, pp. 1-6, doi: 10.1109/GLOCOM.2018.8647128.

[4] N. Al Khater and R. E. Overill, "Network traffic classification techniques and challenges," 2015 Tenth International Conference on Digital Information Management (ICDIM), 2015, pp. 43-48, doi: 10.1109/ICDIM.2015.7381869.

[5] S. Rezaei and X. Liu, "Multitask Learning for Network Traffic Classification," 2020 29th International Conference on Computer Communications and Networks (ICCCN), 2020, pp. 1-9, doi: 10.1109/ICCCN49398.2020.9209652.

[6] Y. Guo, G. Xiong, Z. Li, J. Shi, M. Cui and G. Gou, "TA-GAN: GAN based Traffic Augmentation for Imbalanced Network Traffic Classification," 2021 International Joint Conference on Neural Networks (IJCNN), 2021, pp. 1-8, doi: 10.1109/IJCNN52387.2021.9533942.

[7] Guo, Maohua, Jinlong Fei, and Yitong Meng. "Deep nearest neighbour website fingerprinting attack Technology." Security and Communication Networks 2021 (2021).

[8] Smith, Jean-Pierre & Mittal, Prateek & Perrig, Adrian. (2021). Website Fingerprinting in the Age of QUIC. Proceedings on Privacy Enhancing Technologies. 2021. 48-69. 10.2478/popets-2021-0017.

[9] Rimmer, Vera, et al. "Automated website fingerprinting through deep learning." arXiv preprint arXiv:1708.06376 (2017).

[10] Yong-Jun Wei, Su-Juan Qin "Website Fingerprinting Attack on SSH with Random Forests Classifier". 2nd International Conference on Electronics, Network and Computer Engineering (ICENCE 2016). Volume 67, ISSN 2352-538X, ISBN 978-94-6252-229-9.

[11] IEEE Transactions on Network and Service Management, Volume 19, Issue 2, June 2022, pp 1366–1381,https://doi.org/10.1109/TNSM.2021.3134562

[12] M. Kanagarathinam et al., "Enhanced QUIC Protocol for transferring Time-Sensitive Data," 2022 IEEE International Conference on Communications Workshops (ICC Workshops), 2022, pp. 1-6, doi: 10.1109/ICCWorkshops53468.2022.9882167.

[13] A. Ganji and M. Shahzad, "Characterising the Performance of QUIC on Android and Wear OS Devices," 2021 International Conference on Computer Communications and Networks (ICCCN), 2021, pp. 1-11, doi: 10.1109/ICCCN52240.2021.9522258.

[14] P. Biswal and O. Gnawali, "Does QUIC Make the Web Faster?," 2016 IEEE Global Communications Conference (GLOBECOM), 2016, pp. 1-6, doi: 10.1109/GLOCOM.2016.7841749.

[15] Moharir, M., Adyathimar, K.B., Shobha, G., Soni, V., Scapy scripting to automate testing of networking middleboxes, Advances in Science, Technology and

Engineering Systems Journal Vol. 5, No. 2, 293-298 (2020), ISSN: 2415-6698, pp:293-298

[16] Neha N, Pratiksha Narasimha Nayak G, Nimisha Dey, Malavika Hariprasad, Sandhya S, Minal Moharir, Muteeb Akram A detail survey on QUIC and its impact on network data transmission, 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI).https://ieeexplore.ieee.org/document/9777199

[17] Raghav Rawat, Pratheesh, Krishna Shedbalkar, Minal Moharir, N Deepamala, P Ramakanth Kumar, MGP Tanmayananda, Analysis and Detection of Malicious Activity on DoH Traffic, IEEE 2021 2nd Global Conference for Advancement in Technology (G, CAT), 1-3 Oct. 2021, pp. 1-5, doi: 10.1109/GCAT52182.2021.9587555.

[18] Akshay Koshy, Gaurav Yellur, Hima K, Isha P, Minal Moharir, N Deepamala, P Ramakanth Kumar, An Insight into Encrypted DNS protocol: DNS over TLS, 4th IEEE International Conference on Recent Developments in Control, Automation & Power Engineering (RDCAPE-2021).

[19] Rohith Raj, Rohith R, Minal Moharir, Shobha G. SCAPY – A powerful interactive packet manipulation Program IEEE International Conference on Networking, Embedded and Wireless Systems "Wireless Technology – Building a Progressive World International December 27th – 28th 2018 Organized by Department of Electronics and Communication Engineering B.M.S. College of Engineering pp: 45- 49.

[20] Bharath Rahuldev Patil, , Minal Moharir, Shobha G, Pratik, Ostinato – A powerful traffic generator, 2017 2nd IEEE Sponsored International Conference on Computational System and Information Technology for Sustainable SolutionInternational December 21-23, RV College of Engineering, Bengaluru-59, pp-210-214

[21] Khamar Ali Shaikh, Karthik Bhat, Minal Moharir, Shobha G, A survey on SSL packet structure 2nd IEEE Sponsored International Conference on Computational System and Information Technology for Sustainable Solution International December 21-23, RV College of Engineering, Bengaluru-59, pp- 268 – 273