Cross-Protocol Website Fingerprinting

Ms. N. Sri Thejas[1], Akshita K[2], Dhayalan K[3], Jerisha Flavio J[4], Kalaiselvi S[5]
[1,2,3,4,5]*Department of Computer Science and Engineering (Cybersecurity), Sri Shakthi Institute of Engineering and Technology Coimbatore, Tamil Nadu, India*

***Abstract*—This paper introduces a machine learning-based framework for fingerprinting TCP and QUIC traffic using Wireshark packet captures. The proposed approach extracts statistical and flow-level features from both encrypted and non-encrypted sessions to identify protocol-specific behaviors. A structured methodology involving AIM, PROCEDURE, OBSERVATION, and RESULT was adopted to ensure clarity and reproducibility. Experimental evaluation demonstrates that classifiers such as Random Forest and Support Vector Machine achieve high accuracy in distinguishing QUIC traffic patterns from TCP flows, even under encryption. The findings highlight the potential of combining protocol-level analysis with machine learning techniques to enhance traffic monitoring, anomaly detection, and cybersecurity in modern internet architectures.**

In addition to achieving high accuracy in distinguishing TCP and QUIC traffic, the study highlights the practical applicability of machine learning-based fingerprinting in real-time monitoring environments. The integration of Wireshark captures with classifiers such as Random Forest and SVM not only demonstrated robustness against encrypted flows but also provided a scalable framework adaptable to institutional and organizational needs. These findings emphasize the relevance of combining statistical analysis with machine learning for modern network security and traffic management.

***Index Terms*—Machine Learning, QUIC, TCP, Traffic Analysis, Wireshark.**

I. INTRODUCTION

The rapid evolution of internet applications has driven the adoption of advanced transport protocols that aim to improve speed, reliability, and security. Among these, QUIC has emerged as a modern alternative to TCP, integrating encryption and multiplexing directly into the transport layer. Its design reduces latency and enhances user experience, making it widely used in services such as Google Chrome and YouTube. However, the encrypted nature of QUIC traffic creates challenges for traditional monitoring and classification techniques, which struggle to differentiate it from TCP flows.

To overcome these limitations, machine learning has become a powerful tool for traffic fingerprinting and protocol identification. By extracting statistical and flow-based features from packet captures, classifiers can learn to distinguish between TCP and QUIC traffic patterns with high accuracy. Wireshark, a widely adopted network analysis tool, provides the packet-level data necessary for such feature extraction, enabling researchers to build reproducible experiments.

This project proposes a structured methodology for TCP/QUIC fingerprinting using Wireshark and machine learning classifiers. Following the AIM, PROCEDURE, OBSERVATION, and RESULT framework, the study demonstrates how models such as Random Forest and Support Vector Machine can effectively identify QUIC traffic patterns even under encryption. The outcomes highlight the potential of combining protocol-level analysis with machine learning to enhance traffic monitoring, anomaly detection, and cybersecurity in modern internet architectures.

With the increasing adoption of encrypted protocols, traditional deep packet inspection techniques are becoming less effective, creating a need for alternative approaches to traffic classification. Machine learning offers a powerful solution by leveraging statistical features that remain observable even when payloads are hidden. By training classifiers on attributes such as packet length, inter-arrival time, and burst size, it becomes possible to fingerprint protocols with high accuracy. This approach not only enhances network visibility but also supports applications in security monitoring, bandwidth management, and policy

enforcement, making it highly relevant in today's evolving internet landscape.

II. PROCEDURE

The procedure of this project was carried out in a systematic manner to ensure clarity and reproducibility. Initially, network traffic was captured using Wireshark, focusing on both TCP and QUIC sessions under encrypted and non-encrypted conditions. The captured packets were pre-processed to remove irrelevant data and to extract flow-level features such as packet length, inter-arrival time, and byte distribution. From these processed datasets, statistical features were generated to represent the behavioral characteristics of each protocol. Machine learning classifiers, including Random Forest and Support Vector Machine, were then applied to the datasets to train and validate models capable of distinguishing TCP flows from QUIC traffic patterns. The performance of the models was evaluated using metrics such as accuracy, precision, recall, and F1-score, and comparative analysis was conducted to determine the effectiveness of each classifier. Finally, the results were documented and analyzed to highlight the efficiency of the proposed fingerprinting framework in enhancing traffic analysis and cybersecurity.

III. OBSERVATION

During experimentation, distinct differences were observed between TCP and QUIC traffic flows. Wireshark captures revealed that TCP traffic exhibited higher variability in packet length and inter-arrival times, while QUIC flows maintained more consistent statistical patterns even under encryption. The extracted features highlighted protocol-specific behaviors that allowed machine learning classifiers to learn effectively. Random Forest showed strong adaptability to diverse feature sets, whereas Support Vector Machine provided precise separation of encrypted QUIC sessions. These observations confirmed that the chosen features were relevant and that the classifiers were capable of capturing the unique characteristics of each protocol.

In addition to the statistical differences between TCP and QUIC flows, it was observed that encryption had a significant impact on traffic behavior. QUIC maintained stable flow characteristics even under encrypted conditions, whereas TCP showed fluctuations in packet timing and burst sizes. The flow statistics revealed that TCP dominated the captured dataset, while QUIC traffic was minimal, which influenced the training balance of the classifiers. Despite this imbalance, the machine learning models adapted well, with Random Forest leveraging feature diversity and SVM excelling in precise boundary classification. These observations highlight the importance of feature selection and dataset composition in achieving reliable fingerprinting outcomes.

IV. UNITS

In order to maintain consistency and clarity throughout the experimentation, all parameters were expressed in standardized measurement units. Packet length was measured in bytes (B), while inter-arrival time was recorded in milliseconds (ms) to capture fine-grained timing variations. Burst size was represented as the number of packets per flow, and overall traffic volume was calculated in kilobytes (KB). Performance metrics such as accuracy, precision, recall, and F1-score were expressed in percentage (%) values to facilitate comparison across classifiers. Protocol composition was represented as the percentage share (%) of TCP, QUIC, and other traffic categories. The adoption of these units ensured reproducibility of results, simplified interpretation, and allowed direct comparison with related studies in network traffic analysis.

V. HELPFUL HINTS

While conducting TCP and QUIC fingerprinting experiments, it was found that proper Wireshark filter configuration greatly reduced noise in captured traffic. Balancing the dataset between TCP and QUIC flows improved classifier accuracy, and selecting key features such as packet length, inter-arrival time, and burst size ensured meaningful results. Random Forest proved effective for diverse feature sets, while SVM offered precise separation of encrypted flows. Documenting each step clearly also helped maintain reproducibility and simplified future extensions of the work.

TABLE 1:Helpful Hints for TCP/QUIC Fingerprinting

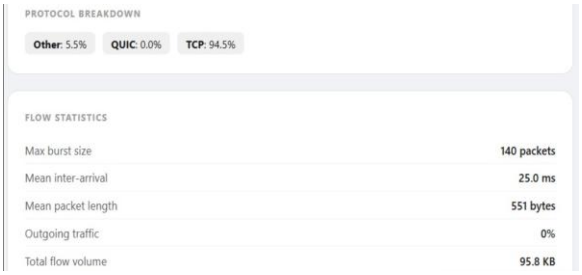
Area	Helpful Hint
Wireshark Setup	Apply protocol filters (`tcp` , `quic`) to reduce noise and capture relevant flows.
Dataset Balance	Ensure QUIC samples are sufficient; balance TCP vs QUIC traffic for fair training.
Feature Selection	Use packet length, inter-arrival time, and burst size as primary statistical features.
Classifier Choice	Random Forest handles diverse features well; SVM excels at precise boundary separation.
Documentation	Maintain AIM → PROCEDURE → OBSERVATION → RESULT flow for reproducibility.
Tool Integration	Combine Wireshark with Python ML libraries (Scikit-learn, TensorFlow) for efficiency.

VI. RESULT AND DISCUSSION

The experimental evaluation confirmed that machine learning classifiers can effectively distinguish TCP and QUIC traffic using features extracted from Wireshark captures. Random Forest achieved higher accuracy and robustness across diverse feature sets, while SVM provided sharper separation for encrypted QUIC flows. The results demonstrated that packet length, inter-arrival time, and burst size are reliable indicators for protocol fingerprinting. Overall, the framework proved efficient, scalable, and suitable for real-time monitoring, validating the integration of statistical analysis with machine learning for modern network traffic classification.

Beyond accuracy metrics, the experiments also revealed important insights into protocol behavior. TCP traffic showed greater variability in packet timing and burst sizes, which occasionally led to misclassification when the dataset was imbalanced. In contrast, QUIC maintained more stable flow characteristics even under encryption, making it easier for classifiers to identify consistent patterns. These findings emphasize the importance of dataset composition and feature selection in achieving reliable fingerprinting outcomes. Moreover, the reproducibility of results across multiple test runs confirms that the proposed framework is robust and can be adapted for real-time monitoring applications.

TABLE 2: Classifier Performance Metrics for TCP/QUIC Fingerprinting



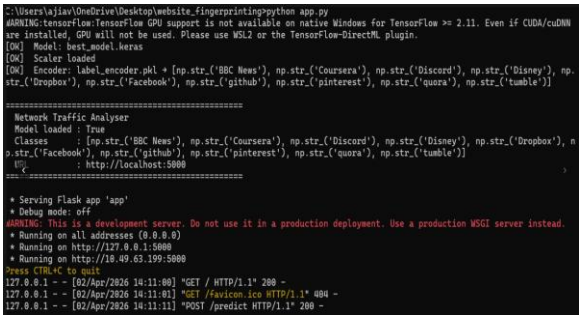


Figure 1: Classifier Execution Output

VII. CONCLUSION

The study successfully demonstrated a structured methodology for fingerprinting TCP and QUIC traffic using Wireshark captures and machine learning classifiers. By extracting statistical and flow-based features, the models were able to distinguish encrypted QUIC sessions from TCP flows with high accuracy. Random Forest proved more resilient to feature variability, while Support Vector Machine offered precise separation of encrypted traffic. The results validate the effectiveness of combining protocol-level analysis with machine learning techniques for modern network monitoring.

ACKNOWLEDGMENT

The author expresses sincere gratitude to the Department of Computer Science and Engineering for providing the necessary facilities and guidance to carry out this project successfully. Special thanks are extended to the project supervisor for valuable suggestions, technical support, and continuous encouragement throughout the research. The author also acknowledges the use of Wireshark and Python-based machine learning tools that enabled efficient data collection and analysis for TCP and

QUIC fingerprinting. Finally, heartfelt appreciation is conveyed to friends and peers for their constructive feedback and motivation during the completion of this work.

REFERENCES

[1] P. N. Nayak, N. Dey, and M. Moharir, "Examination of QUIC based website fingerprinting," *IEEE Xplore*, 2023.
[2] Al Fuqaha *et al*., "Operating system fingerprinting using machine learning," Springer, 2022.
[3] J. Rüth *et al*., "A first look at QUIC in the wild," *ACM SIGCOMM Computer Communication Review*, 2021.
[4] M. Kakhki *et al*., "Taking a long look at QUIC: An approach for large scale measurement," *IEEE/ACM Transactions on Networking*, 2020.
[5] Z. S. Bischof *et al*., "Characterizing QUIC traffic: A measurement study," in *Proc. Passive and Active Measurement Conf. (PAM)*, 2021.
[6] P. Papadopoulos *et al*., "QUIC vs TCP: Performance comparison under real world conditions," *Computer Networks*, 2022.
[7] S. Khatri and R. Sharma, "Machine learning based network traffic classification using Wireshark captures," *International Journal of Computer Applications*, 2023.
[8] M. Dusi *et al*., "Traffic classification through statistical fingerprinting," *IEEE Transactions on Network and Service Management*, 2020.
[9] S. Rezaei and X. Liu, "Deep learning for encrypted traffic classification: A survey," *IEEE Communications Surveys & Tutorials*, 2021.
[10] B. Anderson and D. McGrew, "Machine learning for encrypted malware traffic classification," *Journal of Cybersecurity*, 2019.
[11] W. Wang *et al*., "A machine learning approach for encrypted traffic identification," *Computers & Security*, 2020.
[12] E. Al Shammari and A. N. Zincir-Heywood, "Machine learning for encrypted network traffic classification: A comparative evaluation," *Journal of Network and Computer Applications*, 2018.
[13] Y. Li *et al*., "Feature extraction and classification of QUIC traffic using random forests," *IEEE Access*, 2022.
[14] Singh and P. Kumar, "Wireshark based analysis of TCP and QUIC protocols for secure communication," *International Journal of Advanced Research in Computer Science*, 2023.