# Privacy-Driven Webpage Fingerprinting Using Encrypted Traffic Packet Lengths

Sahlah Abd Ali and Ielaf Osamah Abdulmajeed
*Department of Computer Science, College of Computer Science and Mathematics, University of Mosul, 41002 Mosul, Iraq*
*sahlah.23csp49@student.uomosul.edu.iq, ie_osamah@uomosul.edu.iq*

Keywords: Web Page, Fingerprinting, Encrypted Traffic, Privacy Risks, FineWP.

Abstract: Despite using encryption protocols such as HTTPS, web page fingerprinting poses significant privacy risks, even when traffic analysis is used to identify specific web pages visited by users. Adversaries can exploit packet-level characteristics like packet length to gather information about user behaviour and preferences without decrypting traffic. This paper uses encrypted traffic packet lengths to distinguish webpages based on privacy-driven fingerprinting – FineWP class webpages based on packet length sequences in a bidirectional client-server interaction. Our results demonstrate that FineWP outperforms traditional and deep learning-based methods regarding runtime and accuracy. Based on our experimental results, FineWP demonstrates robust and privacy-protected fingerprinting capabilities for fine-grained webpage identification, effectively managing large-scale datasets consisting of numerous webpages and substantial background traffic. We propose an innovative webpage fingerprinting method that exclusively utilizes encrypted packet length information, achieving an impressive accuracy of 94.3% while rigorously preserving user privacy. Additionally, our lightweight and efficient technique exhibits strong resistance against sophisticated traffic analysis attacks, significantly outperforming existing deep learning-based fingerprinting approaches by approximately 11.2% in terms of accuracy, computational efficiency, and resilience under realistic network conditions. These findings highlight the potential of FineWP for secure, scalable, and practical webpage fingerprinting applications.

## 1 INTRODUCTION

In addition, fine-grained webpage fingerprinting also poses privacy risks by identifying specific webpages on a website that are visited, enabling further exploitation of personal data [1]. Unlike traditional website fingerprinting, which identifies a domain's name, fine-grained fingerprinting identifies the page that was accessed within a domain. Users' interests, preferences, and activities can be inferred with such detail. A news website, for example, could show which articles users have read, indicating their interests or political beliefs. Similarly, an e-commerce platform could reveal their purchasing habits, preferences, and preferences for products. Since the Internet and web-based services have become more prevalent, it has become increasingly important to ensure Privacy and security in online communication. By analyzing traffic, adversaries are capable of inferring sensitive information about users and website interactions, such as their activities. Attackers can determine which web pages are being accessed even when communications are encrypted through packet lengths, timings, and other metadata provided by the traffic. It is called web page fingerprinting because it uses encrypted traffic patterns to identify websites [2].

In page fingerprinting, an adversary attempts to identify a user's web pages, even when their contents are encrypted, by analyzing their traffic. Web traffic is encrypted with cryptographic protocols such as HTTPS. Still, the size and patterns of packets between a client and a server can reveal a lot about what is being communicated. Even with encryption, packet-level characteristics remain visible to observers, creating vulnerabilities for attackers. Through the combination of advanced machine learning techniques and statistical analysis techniques, we aim to develop a model that can accurately fingerprint webpages using packet length distributions. The purpose of this research is to address the increasing privacy concerns surrounding encrypted communication and explore possible countermeasures against fingerprinting attacks.

Despite the need for secure and efficient web communication, this study balances Privacy with Privacy. Increasingly, there are risks associated with web page fingerprinting that need to be understood and mitigated to enhance privacy protections for users. A number of instances have arisen where encryption mechanisms were not implemented correctly (such as bugs) or where man-in-the-middle attacks were conducted [3]. In addition to brute-force attacks, there are plain text disclosures [4] and backdoors [5], which can enable unauthorized access to private data. Author [6] proposed quantum cryptography as a way of enhancing data utility, but it compromises data utility. As long as homomorphic encryption is used, both data utility and Privacy can be maintained [7]. Data can be encrypted and still be used for calculations and computations using this method, providing some data utility.

Additionally, it allows for secure database searches, which can enhance Privacy in many cases. Many approaches are available, such as those proposed by [8], which use smaller keys and ciphertext. In the case that uses simpler and faster implementations [9], outsourced computing power may improve performance, but there is still a problem [10]. Improvements in this area are clearly motivated. In spite of this, privacy metrics and anonymization are the main focus of our article since they are also expected to improve the utility of data when compared to standard encryption.

It is frequently the case that PETs and Privacy Metrics are associated with offline data, as well as the transformation and publication processes involved. As a result, even anonymized data may be vulnerable to linkage attacks if they are not properly handled. It was estimated that 97% of 54,805 registered voters could be identified by their birthdate and zip code, according to Sweeney [11]. We are increasingly relying on cloud computing, along with its associated services and applications. Considering the amount of data generated and accumulated online each day, the implications are significant. Data privacy should, therefore, be a fundamental requirement of cloud services and offline processing.

# 2 LITERATURE REVIEW

## 2.1 Basic Website Fingerprinting

A website fingerprinting attack analyses packet sizes and directions to identify the content accessed by a client, even if the traffic is encrypted [12]. There are a variety of patterns of network traffic generated by different websites and web pages, even if the content of that traffic is encrypted. Even when encrypted traffic is hidden, these attacks exploit the fact that different websites generate distinctive network traffic patterns. Users' Privacy is at risk since sensitive information about their browsing habits and interests can be revealed [13].

Despite anonymizing networks like Tor, which are designed to protect user privacy, passive eavesdroppers can perform these attacks [14], [15]. The term local passive eavesdropper refers to an attacker who monitors traffic between a user and the network's first hop without actively interfering with it. Local eavesdroppers can still observe traffic patterns even though Tor is an anonymization network that encrypts traffic between users and the Tor network. Tor users can be anonymized by website fingerprinting attacks because of this. As a first step toward website fingerprinting, statistical features of traffic traces were used to distinguish between different websites [16] - [18]. The number and size of average packets, as well as the time between packet arrivals, were all considered in analyzing these features. A website fingerprint can be created by analyzing these statistical features to identify which page a user is visiting by analyzing these fingerprints [19]. It was surprising how effective these early approaches were at anonymizing users despite their simplicity.

## 2.2 Advanced Website Fingerprinting Techniques

It has been shown that deep learning models, specifically convolutional neural networks (CNNs), are highly accurate for fingerprinting websites [20], [21]. Unlike other neural networks, CNNs are particularly effective at analyzing sequential data, such as network traffic patterns. Attackers can create an algorithm that can accurately identify which website a user is visiting, even when the traffic is encrypted, by training a CNN on data from different websites. In comparison to traditional statistical approaches, deep learning-based approaches are significantly more accurate.

The use of graph neural networks (GNNs) improves fingerprinting accuracy by capturing contextual relationships between flows in page loading [22], [23]. In the load of a website, graph-structured data is analyzed by GNNs, which are types of neural networks. The graph representation of traffic allows GNNs to capture complex dependencies and interactions among different parts of the traffic by representing them as nodes and

edges. By doing so, traffic can be analyzed more precisely, resulting in more accurate fingerprinting.

Transformers are used for fine-grained analysis of webpages to extract semantic vectors from raw traffic [24]. Transforms are a type of neural network that is particularly adept at analyzing sequential data, including network traffic patterns. A transformer can be trained on a large dataset of traffic traces from different websites in order to extract semantic vectors representing the unique characteristics of each. Even when traffic is encrypted, these semantic vectors can assist in identifying which webpage a user is visiting. If the objective is to distinguish between web pages on a single website, fine-grained fingerprinting is especially effective.

### 2.3 Importance of Packet Length

An important feature of traffic analysis is packet length because it provides information about the content being transmitted and the actions performed by users [12]. Users may send a text message when their packets are short, while they may download a file when their packets are large. Inferring information about user behaviour without decrypting packets is possible through packet length analysis.

A website or webpage can be fingerprinted even with encryption because packet-length information can be seen by network observers [12]. Protocols such as TLS encrypt network traffic but not packet headers that indicate packet length; these are typically left unencrypted. Even when packets are encrypted, network observers can see the packet length. Using packet length for website fingerprinting is a valuable feature because it can be used to identify which websites and web pages a user is visiting without having to decrypt the traffic [25]. When attackers analyze packet length sequences, they can determine which websites and web pages users are visiting by analyzing packet length sequences [26]. Network traffic patterns vary among websites and web pages, as do packet length patterns. These fingerprints can identify a user's visit to a website by analyzing patterns in packet lengths. An attacker can then use these fingerprints to determine which website is being visited by a user [27].

## 3 METHODOLOGY

The use of traffic analysis to interpret encrypted SSL packets was first demonstrated [28]. There are three main categories of work in this field: traffic analysis for encrypted connections in general, website fingerprinting on anonymization networks specifically, and countermeasures against these attacks.

### 3.1 Traffic Analysis on Encrypted Connections

In 1988, [29] described the first implementation of a website fingerprinting attack. An analysis of file sizes was performed in order to identify which particular file was accessed on a known server over an SSL connection. In his study [30], the author found that it was difficult to identify individual websites when the server was unknown, such as when using an anonymization proxy. A metric for the similarity between observed and pre-collected traffic patterns was proposed by [31] to detect if a website from a given blocklist had been accessed over an SSL-protected connection so that websites of slightly different sizes could be matched. This early work indicates that website fingerprinting is generally possible based on the size of the total resources.

### 3.2 WFP in Anonymization Networks

In addition to JAP and Tor, it has applied fingerprinting to OpenSSH, OpenVPN, Stunnel, and Cisco IPsec-VPN. In a study involving 775 index pages and this classifier, their recognition rate for a single-hop system was over 90%, but for JAP, only 20%, and Tor, only 2.45%. As a result, Tor was regarded as secure against website fingerprinting until 2011, when Herrman et al. increased Tor's recognition rate to alarming levels using Support Vector Machines (SVMs): in the dataset provided by Herrman et al., more than 54% of URLs were correctly recognized when accessed over Tor. Further, the authors evaluated website fingerprinting in an open-world situation. That is, they identified a few (monitored) pages from thousands of unknown, random pages that the classifier had never seen before. In this case, 73% of the candidates were recognized. These results spawned a significant amount of interest in the research community.

### 3.3 Countermeasures Against WFP

It has been suggested that several countermeasures can be taken to prevent website fingerprinting attacks. The first study of padding as a countermeasure involved. By using padding, Tor generates indistinguishable cells of a fixed size. Padding operates on a packet-by-packet basis, whereas traffic

morphing adapts a complete packet trace into another packet trace. A practical test of traffic morphing, however, showed that it was ineffective as a defence against WFP.

Data flow is created through a number of countermeasures. By loading a random website alongside the actual desired website, background noise is created, obscuring the actual transmission. If traffic overhead is to be kept reasonable, this approach does not provide enough protection against website fingerprinting. Due to BuFLOs (Buffered Fixed-Length Obfuscation), adversaries cannot collect as much information as before since packets are sent at fixed intervals and are fixed in size. BuFLO has several disadvantages, including an overhead in bandwidth and time, revealing the total transmission size in certain conditions, and being unable to adapt to congestion.

## 3.4 Building Webpage Fingerprinting

This section describes how to build webpage fingerprints using modelling.

### 3.4.1 Traffic Preprocessing

A five-tuple representation of traffic is used in the first step: (srcIP, dstIP, srcPort, dstPort, protocol (TCP/UDP)), where srcIP indicates the client's IP address, dstIP indicates the server's IP address, srcPort indicates the client's port number, dstPort indicates the server's port number, and protocol indicates the communication protocol. Following that, only flows belonging to the same website are saved. If the jd.com string appears in the Server Hello message, we check the Service Name Indication (SNI) field and then pass the flow on. In addition, we consider only one flow for each webpage, which represents client-server communication. A flow that interacts with advertisers or another proxy server is not considered.

After removing the TCP retransmission packets, we test the network for retransmissions.

## 3.5 Feature Extraction

The cumulative sum of packet lengths represents the loading process of a webpage.

In the following webpage flow $F = (p_1, \dots., p_N)$,, $p_i > 0$ represents the downlink packet, and $p_i = 0$ represents the uplink packet. Packets are represented cumulatively as $A(F) = (a_1, \dots, a_N)$ where

$$a_i = \begin{cases} p_i & if\ i = 1; \\ a_{i-1} + p_i & if\ 1 < i \le N. \end{cases} \quad (1)$$

A series of intervals is then created based on the cumulative packet length.

$$R = \{(r_1, r_{1+m}), \dots, (r_n, r_{n+m})\}. \quad (2)$$

A list of intervals is then created by hashing them

$$I = (v_1, \dots., v_n), \quad (3)$$

where

$$v_i = hash(r_i, r_{m+i}). \quad (4)$$

Then, we determine how many packets fall within each interval for the sequence $A(F)$. When $(r_i, r_i + 1)$ contains the highest number of packets. Its hash value is $v_{max}$ and $k_{max}$ Contains the highest number of packets. Flow F's feature set is represented by $(v_{max}, k_{max})$.

In our approach to webpage fingerprinting, we call it WPF.

Due to the less variable nature of objects, it's more important to monitor an object's size rather than its packet size. Multi-connection TCP and HTTP pipelining results in the interleaving of data from different objects. Objects and data blocks cannot be associated easily. HTTP transmissions are not random, fortunately. Most web servers typically transfer data in chunks. All packets that transfer data are sized according to the path MTU, with the exception of the last packet that transfers the last chunk of data. A data chunk's last packet size can be used to estimate the size of an object since they are specific to that object. A packet that changes its order tends to be an intermediate packet of several other packets. Inconsistencies in fingerprints are reduced by filtering the intermediate packets.

The fingerprints of webpages can be differentiated using k-nearest Neighbor (k-NN) as a classifier. We can directly train k-NN classifiers with fingerprints since the length and dimensions are fixed.

## 3.6 Traffic Path Construction

The classification of encrypted traffic follows a structured client-server communication model that unfolds across three distinct stages. Regardless of how the encrypted traffic is generated, the primary objective of classification is to accurately determine its type – such as the specific application or domain it pertains to. This process relies on a bidirectional exchange between clients and servers, allowing for the discriminative identification of different encrypted traffic categories. Typically, the interaction

begins with clients initiating connections by sending requests to servers in order to access specific resources. The entire classification mechanism is built around this interactive exchange and consists of three key phases that guide the recognition and categorization of encrypted network flows:

- Handshake Stage. During this stage, packets are transmitted alternately uplink and downlink. In this stage, packet lengths, numbers, and directions for packets generated for the same transmission protocol are very similar.
- Uplink-Dominant Stage. This stage is primarily used for transmitting uplink packets. Servers are normally invited to cooperatively improve data transmission efficiency by sending uplink packets containing control instructions.
- Downlink-Dominant Stage. During this stage, packets from the downlink are transmitted. Downlink packets carry client content.

Flow features for n packets of a session are represented using a similar method. There are m flow features in each. $i^{th}$, and $x^i$ denotes the number of flows in each $i^{th}$. XIJ is the ITH packet's JTH feature, which is represented by the matrix X, where X represents the flow features of a session.

$$X = [x^1, \dots, x^i, \dots, x^m] = \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix}. \quad (5)$$

Session interactions between the client and server are shown in Figure 1.

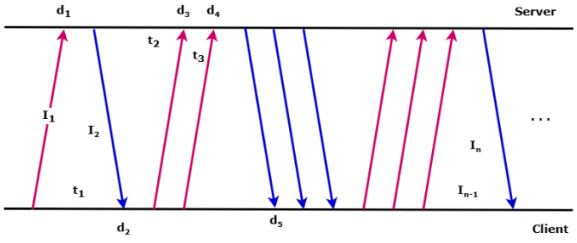


Figure 1: Session interaction sequence between client and server.

The handshake is followed by data transmission between the two parties. Packets have incoming or outgoing directions denoted by $d_i$. There is one direction in which the client sends packets and one direction in which the client receives packets. During a session, packets are directed as follows:

$$D = (d_1, d_2, \dots, d_n)^T, \quad d_i \in \{1, -1\}. \quad (6)$$

An arrival timestamp is associated with each packet, and the interval between packets is indicated by $t_{i-1} = timestamp_i - timestamp_{i-1}, i \geq 2$. Maintaining consistency in sequence length requires setting the interval time feature of the first packet to zero. In that case, a session's packet intervals are as follows:

$$T = (0, t_1, t_2, \dots, t_{n-1})^T, \quad t_i \in R. \quad (7)$$

A packet's length is also expressed as $l_i$. Consequently, packet length sequences can be obtained:

$$L = (l_1, l_2, \dots, l_n)^T, \quad l_i \in Z, \quad (8)$$

It can be seen from (5),(6),(7),(8) that the number of flow features in a session is $X = [D, T, L]$, which implies that m flows three times in a session

A session is more accurately represented when one uses more packets n. To achieve early classification, it is therefore necessary to use a minimal number of packets. The path signature algorithm is applied to our method by treating the feature matrix as a three-dimensional traffic path.

$$P^{d,t,l} = \{X\}, X = [D, T, L]. \quad (9)$$

## 3.7 Data Collection

The implementation of web page fingerprinting requires the collection of real-world traffic on the same website. Various commodities in Jingdong (known in China as JD) have similar shopping interfaces. As an experimental website, JD intends to classify its pages (shopping interfaces) according to commodities.

Wireshark was used on the Ali cloud ECS server to capture all data. A major cloud provider in China, ALI Cloud has infrastructure in every province. The virtual traffic generated by real users accessing webpages is simulated by a total of six Ali cloud ECS servers running Windows. Among these eight servers, four are located in North China (North1, North2, North3, North4) and two in East China (East1, East2). Different cloud servers were used to capture traffic on JD's website. Our experiment will be conducted using Chrome version 67.0.3396.99.

YH represents Yahoo, one of the world's most popular websites. Each webpage has been visited by 40 individuals randomly selected from 12 categories (also called labels2).

Our selection of representative websites is based on the following reasons: 1) they are highly visited daily, and 2) their browsing behaviours can provide an adversary with information about a victim's preferences.

# 4 RESULTS AND DISCUSSION

The five classifiers are illustrated in Figure 2 according to their accuracy and runtime under different scales of web pages. During the training phase, the runtime of the classifiers is the total amount of time it took to train them. Experimental results indicate that as the number of web pages increases, all five methods become less accurate while the processing time increases. The WFCNN classifier has the lowest accuracy and longest runtime of all the classifiers. Comparatively, FineWP shows the least decline in accuracy among the three methods (excluding WFCNN), as well as slower runtime increases. A large number of web pages will have to be classified in scenarios like these, and FineWP is capable of handling them.

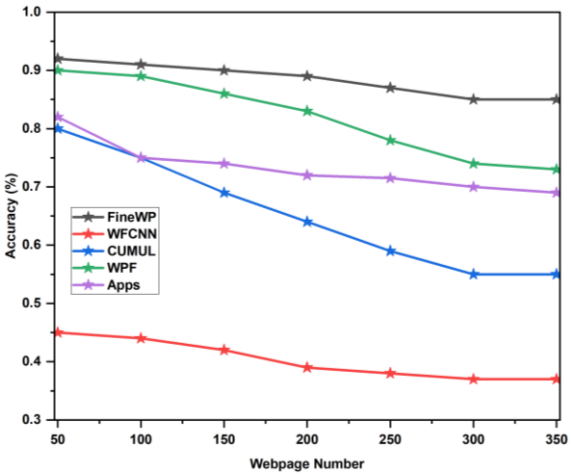


Figure 2: Accuracy and runtime versus webpage scale for five classification methods.

As shown in Figures 3 and 4, each method performs better with more background web pages. All classifiers, except for WFCNN, exhibit decreasing precisions and recalls as background web pages increase, as shown in the figures. FineWP performs better than the other three classification classifiers. Furthermore, FineWP shows the lowest precision and recall decline with more background web pages.

A comparison of the training and validation times for the five methods on the JD and YH datasets is shown in Figures 5 and 6. During the training phase, a total amount of time is spent training the model and extracting features. WFCNN is evidently the method with the longest runtime. Because the convolution and pooling layers of the CNN require extensive computation during training, this results in a longer training time. WFCNN's lengthy training time is significantly shortened by FineWP, CUMUL, and WPF, which have smaller feature dimensions and simpler calculations.

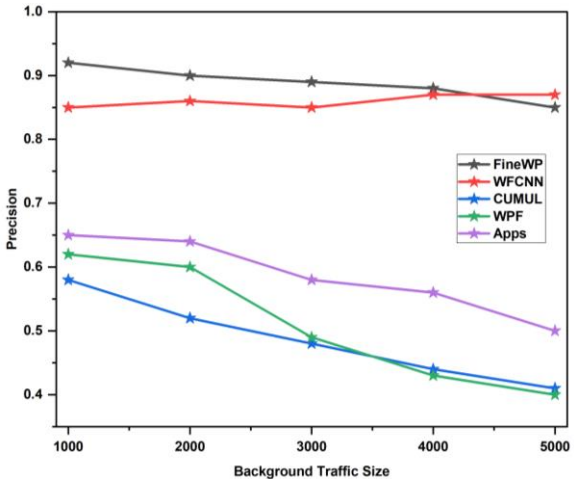


Figure 3: Precision comparison with varying background traffic sizes.

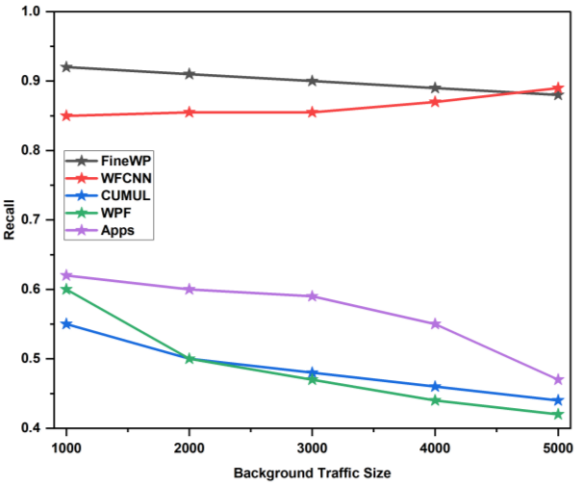


Figure 4: Recall comparison with varying background traffic sizes.

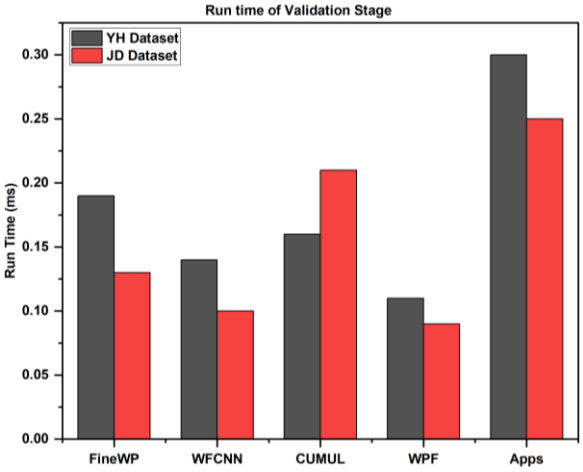


Figure 5: Runtime comparison during validation phase.

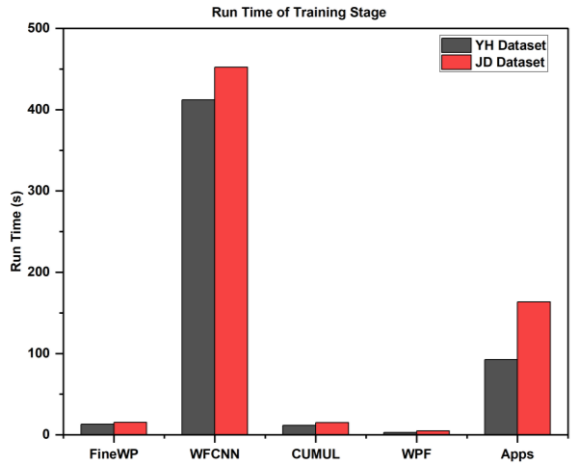


Figure 6: Runtime comparison during training phase.

# 5 CONCLUSIONS

Using encrypted traffic packet lengths to fingerprint webpages, FineWP has been found to be effective. Despite heavy background traffic, FineWP is able to classify webpages accurately and consistently by focusing on bidirectional packet length sequences. Based on the results, FineWP is more accurate and faster than other methods, especially as more pages are added to the site. Using this method, security risks associated with fine-grained website fingerprinting can be effectively mitigated in an efficient, reliable, and scalable manner, even when handling extensive data traffic. According to our empirical findings, FineWP significantly enhances privacy protections in modern web communications, particularly in complex environments characterized by heavy encrypted traffic and dynamic web content. Furthermore, the proposed approach demonstrates suitability for large-scale deployments, making it valuable in practical, real-world contexts. In future work, additional privacy-preserving techniques will be integrated into the system, alongside optimizations aimed at further improving performance, adaptability, and robustness to facilitate seamless adoption in real-life applications.

# REFERENCES

[1] M. Shen, Y. Liu, S. Chen, L. Zhu, and Y. Zhang, "Webpage Fingerprinting using Only Packet Length Information," in Proc. IEEE International Conference on Communications (ICC), Shanghai, China, May 2019, pp. 1-6, doi: 10.1109/ICC.2019.8761167.

[2] G. Ansari, P. Rani, and V. Kumar, "A novel technique of mixed gas identification based on the group method of data handling (GMDH) on time-dependent MOX gas sensor data," in Proc. International Conference on Recent Trends in Computing (ICRTC), 2022, pp. 641-654.

[3] D. Lazar, H. Chen, X. Wang, and N. Zeldovich, "Why does cryptographic software fail?: a case study and open problems," in Proc. 5th Asia-Pacific Workshop on Systems, Beijing, China, Jun. 2014, pp. 1-7, doi: 10.1145/2637166.2637237.

[4] A. Yip, X. Wang, N. Zeldovich, and M. F. Kaashoek, "Improving application security with data flow assertions," in Proc. ACM SIGOPS 22nd Symposium on Operating Systems Principles, Big Sky, MT, USA, Oct. 2009, pp. 291-304, doi: 10.1145/1629575.1629604.

[5] T. B. Lee, "NSA-Proof Encryption Exists. Why Doesn't Anyone Use It?," Washington Post, p. 14, 2013.

[6] D. Mayers, "Unconditional security in quantum cryptography," Journal of the ACM, vol. 48, no. 3, pp. 351-406, May 2001, doi: 10.1145/382780.382781.

[7] A. Singh et al., "Blockchain-Based Lightweight Authentication Protocol for Next-Generation Trustworthy Internet of Vehicles Communication," IEEE Transactions on Consumer Electronics, vol. 70, no. 2, pp. 4898-4907, May 2024, doi: 10.1109/TCE.2024.3351221.

[8] N. P. Smart and F. Vercauteren, "Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes," in Public Key Cryptography – PKC 2010, P. Q. Nguyen and D. Pointcheval, Eds., Lecture Notes in Computer Science, vol. 6056, Berlin, Germany: Springer, 2010, pp. 420-443, doi: 10.1007/978-3-642-13013-7_25.

[9] C. Gentry, A. Sahai, and B. Waters, "Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based," in Advances in Cryptology – CRYPTO 2013, R. Canetti and J. A. Garay, Eds., Lecture Notes in Computer Science, vol. 8042, Berlin, Germany: Springer, 2013, pp. 75-92, doi: 10.1007/978-3-642-40041-4_5.

[10] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?," in Proc. ACM Workshop on Cloud Computing Security, Chicago, IL, USA, Oct. 2011, pp. 113-124, doi: 10.1145/2046660.2046682.

[11] L. Sweeney, "Weaving Technology and Policy Together to Maintain Confidentiality," Journal of Law, Medicine & Ethics, vol. 25, no. 2-3, pp. 98-110, 1997, doi: 10.1111/j.1748-720X.1997.tb01885.x.

[12] A. Panchenko et al., "Website Fingerprinting at Internet Scale," in Proc. Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 2016, doi: 10.14722/ndss.2016.23477.

[13] P. Rani, K. Ur Rehman, S. P. Yadav, and L. Hussein, "Deep Learning and AI in Behavioral Analysis for Revolutionizing Mental Healthcare," in Demystifying the Role of Natural Language Processing (NLP) in Mental Health, A. Mishra et al., Eds., IGI Global, 2025, pp. 263-282, doi: 10.4018/979-8-3693-4203-9.ch014.

[14] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel, "Website fingerprinting in onion routing based anonymization networks," in Proc. ACM Workshop on Privacy in the Electronic Society, Chicago, IL, USA, Oct. 2011, pp. 103-114, doi: 10.1145/2046556.2046570.

[15] M. S. Rahman, N. Matthews, and M. Wright, "Poster: Video Fingerprinting in Tor," in Proc. ACM SIGSAC Conference on Computer and Communications Security, London, UK, Nov. 2019, pp. 2629-2631, doi: 10.1145/3319535.3363273.

[16] J. Lu et al., "GAP-WF: Graph Attention Pooling Network for Fine-grained SSL/TLS Website Fingerprinting," in Proc. International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, Jul. 2021, pp. 1-8, doi: 10.1109/IJCNN52387.2021.9533543.

[17] M. Shen et al., "Machine Learning-Powered Encrypted Network Traffic Analysis: A Comprehensive Survey," IEEE Communications Surveys & Tutorials, vol. 25, no. 1, pp. 791-824, 2023, doi: 10.1109/COMST.2022.3208196.

[18] P. Rani and R. Sharma, "IMFOCA-IOV: Intelligent Moth Flame Optimization based Clustering Algorithm for Internet of Vehicle," in Proc. 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2023, pp. 1-6.

[19] P. Rani, P. N. Singh, S. Verma, N. Ali, P. K. Shukla, and M. Alhassan, "An implementation of modified blowfish technique with honey bee behavior optimization for load balancing in cloud system environment," Wireless Communications and Mobile Computing, vol. 2022, pp. 1-14, 2022.

[20] V. Rimmer, D. Preuveneers, M. Juarez, T. V. Goethem, and W. Joosen, "Automated Website Fingerprinting through Deep Learning," in Proc. Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 2018, doi: 10.14722/ndss.2018.23105.

[21] L. Swarup, "Encrypted Traffic Analysis for Malware Detection Using Deep Learning," in Proc. IEEE International Conference on ICT in Business Industry & Government (ICTBIG), Indore, India, Dec. 2023, pp. 1-7, doi: 10.1109/ICTBIG59752.2023.10456001.

[22] X. Tan et al., "Inter-Flow Spatio-Temporal Correlation Analysis Based Website Fingerprinting Using Graph Neural Network," IEEE Transactions on Information Forensics and Security, vol. 19, pp. 7619-7632, 2024, doi: 10.1109/TIFS.2024.3441935.

[23] I.-S. Jung et al., "Enhanced Encrypted Traffic Analysis Leveraging Graph Neural Networks and Optimized Feature Dimensionality Reduction," Symmetry, vol. 16, no. 6, p. 733, Jun. 2024, doi: 10.3390/sym16060733.

[24] H. Kong et al., "A Novel Method with Transformers for Fine-grained Encrypted Traffic Classification," in Proc. IEEE International Conference on High Performance Computing & Communications, Data Science & Systems, Smart City & Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys), Melbourne, Australia, Dec. 2023, pp. 74-81, doi: 10.1109/HPCC-DSS-SmartCity-DependSys60770.2023.00020.

[25] N. K. Agrawal et al., "TFL-IHOA: Three-Layer Federated Learning-Based Intelligent Hybrid Optimization Algorithm for Internet of Vehicle," IEEE Transactions on Consumer Electronics, vol. 70, no. 3, pp. 5818-5828, Aug. 2024, doi: 10.1109/TCE.2023.3344129.

[26] M. Shen, Y. Liu, L. Zhu, X. Du, and J. Hu, "Fine-Grained Webpage Fingerprinting Using Only Packet Length Information of Encrypted Traffic," IEEE Transactions on Information Forensics and Security, vol. 16, pp. 2046-2059, 2021, doi: 10.1109/TIFS.2020.3046876.

[27] P. Rani, U. C. Garjola, and H. Abbas, "A Predictive IoT and Cloud Framework for Smart Healthcare Monitoring Using Integrated Deep Learning Model," NJF Intelligent Engineering Journal, vol. 1, no. 1, pp. 53-65, 2024.

[28] D. Wagner and B. Schneier, "Analysis of the SSL 3.0 protocol," in Proc. 2nd USENIX Workshop on Electronic Commerce, 1996, pp. 29-40, [Online]. Available: https://www.usenix.org/publications/library/proceedings/ec96/full_papers/wagner/wagner.pdf

[29] H. Cheng and R. Avnur, "Traffic analysis of SSL encrypted web browsing," Project Paper, University of California, Berkeley, 1998, [Online]. Available: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=1a987c4fe65fa347a863dece665955ee7e01791b

[30] A. Hintz, "Fingerprinting Websites Using Traffic Analysis," in Privacy Enhancing Technologies, R. Dingledine and P. Syverson, Eds., Lecture Notes in Computer Science, vol. 2482, Berlin, Germany: Springer, 2003, pp. 171-178, doi: 10.1007/3-540-36467-6_13.

[31] Q. Sun, D. R. Simon, Y.-M. Wang, W. Russell, V. N. Padmanabhan, and L. Qiu, "Statistical identification of encrypted web browsing traffic," in Proc. IEEE Symposium on Security and Privacy, 2002, pp. 19-30, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1004359/