Detecting Malware in Encrypted Network Traffic Using Machine Learning and TLS Fingerprints

Eduardo Polo-Peyres, Jorge E. López de Vergara, Gustavo Sutter, Iván González, and Luis de Pedro

Escuela Politécnica Superior, Universidad Autónoma de Madrid, Spain
eduardo.polo@estudiante.uam.es, {jorge.lopez_vergara, gustavo.sutter, ivan.gonzalez, luis.depedro}@uam.es

Abstract. The increasing use of encryption protocols such as Transport Layer Security (TLS) has led to enhanced privacy and security for internet users. However, this development has also enabled malware to conceal its communications within encrypted channels. In this work, we explore the application of machine learning techniques to detect malware in encrypted network traffic. To this end, we compare two distinct approaches: one based on statistical flow features and the other one based on TLS fingerprinting (JA4+). In order to accomplish this objective, we have developed and evaluated two state-of-the-art solutions—a MalDIST-inspired model and JA4+ fingerprint-based classification. The experimental results, based on a curated dataset, show that both models exhibit high levels of accuracy. Notably, JA4+ fingerprinting offers a favorable trade-off between accuracy metrics and overall processing speed compared to the MalDIST-inspired model, making it a promising candidate for deployment in real-world environments. Furthermore, we introduce JA4TS, another fingerprinting technique that focuses on TCP SYN-ACK packets, enhancing the capability of the JA4+ framework to predict malware by identifying TCP/IP stack characteristics, a subject of particular relevance as SNI encryption becomes more prevalent. These findings underscore the efficacy of lightweight, metadata-based models for malware detection in encrypted traffic, particularly in the context of IoT and IIoT networks, where privacy and efficiency are paramount.

Keywords: TLS, JA4+, JA4TS, Malware Detection, Machine Learning, Encrypted Traffic, Flow Statistics

1 Introduction

The continuous evolution of network traffic presents an ongoing challenge for malware detection systems. Malicious actors increasingly exploit encrypted protocols like TLS (Transport Layer Security) to evade detection, embedding control commands and exfiltrating data in hidden ways. According to [6], over 71% of malware today uses SSL/TLS to mask its operations. While encryption is essential for user privacy, it also introduces challenges for security monitoring.

The adoption of encryption protocols such as HTTPS and VPNs has become nearly ubiquitous, particularly following regulatory pushes for data protection and the widespread use of cloud services. These encryption techniques, while necessary for

confidentiality and integrity, reduce visibility for intrusion detection systems (IDS), which traditionally rely on payload inspection. Consequently, many conventional security solutions are rendered ineffective, creating a need for new paradigms focused on metadata analysis and behavior-based profiling.

Machine learning has emerged as a promising solution in this context, offering the capacity to recognize complex patterns in encrypted traffic based on side-channel information. Rather than attempting decryption or deep packet inspection, machine learning algorithms can use flow-level statistics, timing information, and TLS fingerprinting to distinguish between benign and malicious behavior. In particular, fingerprinting techniques like JA4+ [1] allow the identification of applications and actors based on the structure of their TLS handshakes, even when payloads remain encrypted.

In this work, we conduct a comprehensive study and implementation focusing on two prominent detection models: a statistical model inspired by MalDIST and a fingerprinting-based classifier using JA4+. By rigorously evaluating and comparing these methods on diverse datasets, we aim to determine the strengths and trade-offs of each approach, revealing the efficiency of JA4+. We also explore the emerging JA4TS fingerprint, which analyzes TCP SYN-ACK packets to infer server TCP/IP stack characteristics, presenting it as a crucial advancement for future-proofing malware detection against evolving encryption practices like SNI encryption.

This work is particularly relevant in the context of emerging IoT and IIoT environments, where encrypted communications are pervasive and lightweight, and real-time detection systems are essential. By leveraging TLS metadata rather than payloads, the proposed models align with privacy-preserving monitoring practices while ensuring robustness and efficiency—key requirements in sustainable and secure cyber-physical systems.

The remainder of this paper is organized as follows. Section 2 presents the state of the art in network malware detection and fingerprinting techniques. Next, Section 3 describes the methodology, including dataset preparation and model architecture. Then, Section 4 reports the experimental results and performance analysis. Later, Section 5 discusses the implications of the findings, and finally, Section 6 concludes with future directions for improvement and deployment.

2 State of the Art

The rise of encrypted traffic has shifted the landscape of network security. Traditional techniques, such as Deep Packet Inspection (DPI), rely heavily on payload visibility. However, encryption protocols like TLS prevent such inspection, thus reducing the efficacy of conventional IDS. As a result, research has shifted towards leveraging metadata and side-channel information to detect anomalous or malicious behavior without decrypting content.

This challenge is even more pronounced in Internet of Things (IoT) and Industrial IoT (IIoT) environments, where devices often exhibit inconsistent TLS implementations. Paracha et al. [12] analyzed a wide range of consumer IoT devices and observed frequent failures to validate certificates, poor cipher suite support, and variability in handshake behavior. These shortcomings make IoT systems particularly attractive to

malware authors and highlight the urgent need for detection methods that do not rely on payload inspection. In this context, lightweight and privacy-preserving approaches based on TLS fingerprinting and flow-level statistics—such as those explored in this work—have become increasingly relevant.

TLS fingerprinting has emerged as a solution to this visibility problem. JA3, introduced by John Althouse et al. in 2017 [2], generates a fingerprint by hashing ordered values from the Client Hello message (including TLS version, cipher suites, extensions, elliptic curves, and point formats). JA3S, its server-side counterpart, hashes fields from the Server Hello.

Despite its utility, JA3 had limitations. For instance, the same application could produce different JA3 hashes by changing the order of the cipher suites, making it hard to uniquely identify software. To address this, JA4+ was proposed in 2023 by the same authors [1]. It expands on JA3 by creating human-readable and structured fingerprints for various protocols: JA4 (client-side TLS), JA4S (server-side TLS), JA4X (certificate fingerprinting), and JA4H (HTTP). This allows more granular classification and supports composite analysis using multiple fingerprint types. An additional extension, JA4T and its server-side counterpart, JA4TS, applies these fingerprints to the TCP handshake parameters [8]. This capability is very important for malware detection, as distinct OS configurations can often indicate the presence of specific malware families or their associated command-and-control infrastructure.

Fingerprinting methods have proven valuable for identifying both benign and malicious applications based solely on their encrypted handshake behavior. For example, Matoušek et al. [11] demonstrated that combining JA4 with JA4S and SNI allows over 89% of flows to be uniquely identified. Their analysis focused primarily on the distinctiveness of TLS fingerprints across applications and families, providing a taxonomy of fingerprint structures observed in malware communications. However, their work did not evaluate the classification performance of these fingerprints within a machine learning framework, nor did it compare fingerprint-based approaches with statistical models. In contrast, our work builds on these insights by analyzing the uniqueness of JA4+ fingerprints, and also integrating them into practical classification pipelines, assessing their detection accuracy, and exploring hybrid models that fuse fingerprinting with statistical flow features.

In parallel to fingerprinting, statistical analysis of network flows provides a powerful framework for identifying malware. Flow-level features such as packet sizes, inter-arrival times, and TCP window sizes can be aggregated and modeled to uncover anomalous behavior. Bader et al. [3] proposed MalDIST, a multimodal deep learning model that combines payload bytes, protocol fields, and statistical matrices computed from the first 32 packets of each flow. Their approach achieves high detection accuracy by leveraging both content-based and statistical features. However, the inclusion of payload-level data, while informative, limits its applicability in encrypted environments, where payloads are inaccessible or obfuscated. Furthermore, MalDIST relies on deep convolutional architectures applied to matrix-encoded flows, which increases computational overhead and may hinder deployment in latency-sensitive or resource-constrained settings such as IoT networks.

In contrast, our work focuses exclusively on metadata and statistical features available regardless of encryption, avoiding dependency on payload inspection. By simplifying the model architecture and emphasizing inference efficiency, we aim to preserve robustness while enabling practical deployment. This aligns with recent research trends that prioritize interpretable, resilient, and lightweight detection strategies suitable for encrypted traffic scenarios [14].

Recent works have also addressed the limitations of deep learning models in the context of encrypted traffic. Luis-Bisbé et al. [9] applied explainable AI (XAI) techniques to convolutional neural networks trained on statistical flow data, revealing that these models often struggle to generalize in the presence of encryption, as critical payload information is unavailable. Their findings underscore the difficulty of relying solely on deep architectures like CNNs for encrypted traffic classification, and highlight the value of interpretable features and model simplicity.

Nevertheless, all metadata-based approaches face challenges. For instance, SNI fields used in JA4+ may become encrypted in future TLS versions (e.g., via Encrypted Client Hello, ECH). However, the inclusion of JA4TS, which derives fingerprints from TCP handshake packets, offers a resilient alternative for identifying network characteristics independent of TLS layer encryption. Statistical features can also vary due to benign software updates or operating system changes. As such, combining complementary feature types and modeling strategies—as proposed in this work—offers a path toward more robust and resilient detection systems.

3 Methodology

We developed and evaluated two machine learning models for malware detection over encrypted traffic: a statistical model inspired by MalDIST and a JA4+-based fingerprint classifier. This section outlines the dataset, feature extraction, and training process done to create these models.

3.1 Dataset and Preprocessing

The dataset comprises:

- **Malware samples** from malware-analysis.net [10] and Tria.ge [13], covering several malware families, including Dridex[1], Emotet[2], Hancitor[3], and Valak[4].
- **Benign samples** from ISCX VPN-nonVPN [4] and clean traffic captured via Tria.ge.

Additionally, a second dataset from [11] has been used to compare our solution with that work.

Each PCAP file with the traffic captures was split into sessions and used to extract CSVs with statistical features and JA4+ fingerprints. Session labels were assigned and balanced with SMOTE [5] to mitigate class imbalance.

[1] https://malpedia.caad.fkie.fraunhofer.de/details/win.dridex
[2] https://malpedia.caad.fkie.fraunhofer.de/details/win.emotet
[3] https://malpedia.caad.fkie.fraunhofer.de/details/win.hancitor
[4] https://malpedia.caad.fkie.fraunhofer.de/details/js.valak

3.2 Feature Extraction

Feature extraction was different for each approach:

- **Statistics:** 198 flow-level features were computed using `scapy` and custom Python tools. They include statistics over packet sizes, timings, TCP window size, and flow duration, organized across five subgroups, as proposed in the original MalDIST paper [3], namely: bidirectional packets, source to destination packets, destination to source packets, TCP handshake packets, and data transfer packets.
- **JA4+ Fingerprints:** TLS handshake fields were parsed via `tshark`, then hashed to generate JA4, JA4S, JA4TS, as well as the SNI. Extraction tools were built using Python and shell scripts, with efficient processing times, as detailed in subsection 4.3.

3.3 Model Descriptions

Two different approaches were implemented:

- **MalDIST-inspired model:** A Random Forest classifier (50 estimators) with default hyperparameters trained on statistical features. It provided high accuracy and explainability.
- **JA4+ model:** JA4+ features were vectorized with Scikit-learn `FeatureHasher` (dimension 1024) and classified using a Random Forest with default hyperparameters. This model favored simplicity with respect to the statistical model.

Although applying optimization techniques to each model could improve results, the default hyperparameters were kept to ensure a fair comparison and reproducibility.

3.4 Training and Evaluation

Models used 80/20 stratified train-test splits. Metrics included accuracy, precision, recall, and F1-score. All implementations used Python 3.10 with `scikit-learn`, `tensorflow`, and `imbalanced-learn`.

4 Results and Evaluation

This section presents the empirical evaluation of the three proposed models: the MalDIST-inspired statistical model, the JA4+-based fingerprint model, and the hybrid model. We compare them in terms of classification accuracy, processing efficiency, and model robustness.

4.1 Classification Performance

The binary classification task distinguishes between malicious and benign traffic. Table 1 summarizes the performance of the three models for binary classification. The results show that both the MalDIST-inspired statistical model and the JA4+-based fingerprint model achieve high accuracy in distinguishing between malicious and benign traffic. The JA4+ model, in particular, maintains highly competitive performance across all metrics. The inclusion of JA4TS in the JA4+ fingerprinting, while showing a slightly lower accuracy in this specific binary classification context, offers distinct advantages with encrypted SNI and detailed malware family identification as discussed in Section 5.

Table 1. Binary classification performance (malware vs benign)

Model	Accuracy	Precision	Recall	F1-score
MalDIST-inspired	98.28%	98.10%	98.67%	98.38%
JA4+ (JA4+JA4S)	95.12%	95.11%	95.32%	95.11%
JA4+ (JA4+JA4S+SNI)	98.25%	98.30%	98.18%	98.24%
JA4+ (JA4+JA4S+JA4TS)	96.28%	96.28%	96.28%	96.28%

4.2 Multiclass Malware Classification

For multiclass classification, the models attempted to classify traffic flows into one of the five labels: (0) Benign, (1) Dridex, (2) Emotet, (3) Hancitor, and (4) Valak. Figure 1 shows the confusion matrix for the MalDIST model using Random Forest, which was identified as the best performing classifier for this task. We also tested other algorithms, such as XGBoost, Extra trees, KNN and logistic regression.

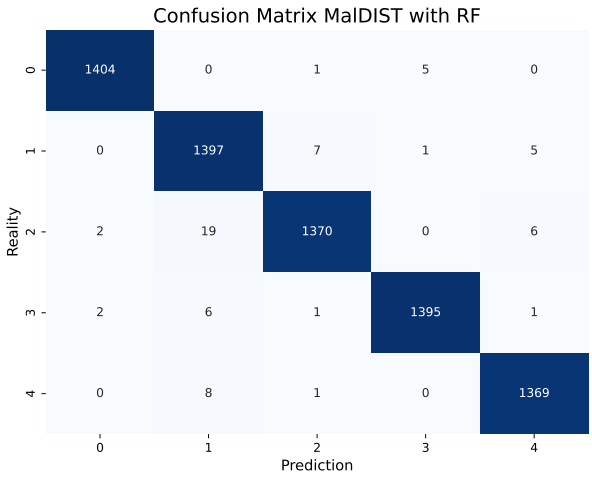
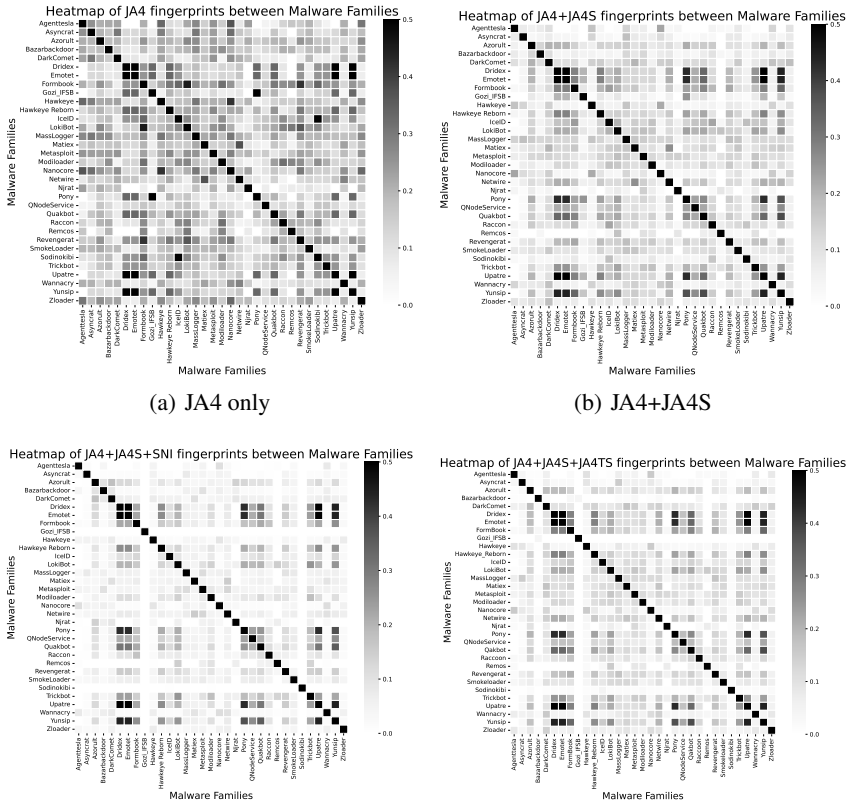


Fig. 1. Confusion matrix of the MalDIST-inspired model with Random Forest classifier. Labels are related to (0) Benign, (1) Dridex, (2) Emotet, (3) Hancitor, and (4) Valak.

As seen in the matrix, the MalDIST-inspired model demonstrated strong accuracy across all malware families, particularly for Emotet and Hancitor. The confusion matrix also reveals that the most frequent misclassifications occurred between Dridex and Valak, as evidenced by a notable overlap in their predictions. This aligns with the statistical similarity between their traffic patterns noted during feature analysis. Such confusion may be due to similarities in TLS usage or behavior of the underlying malware loaders, which emphasizes the importance of richer or complementary features when attempting to distinguish between closely related families.



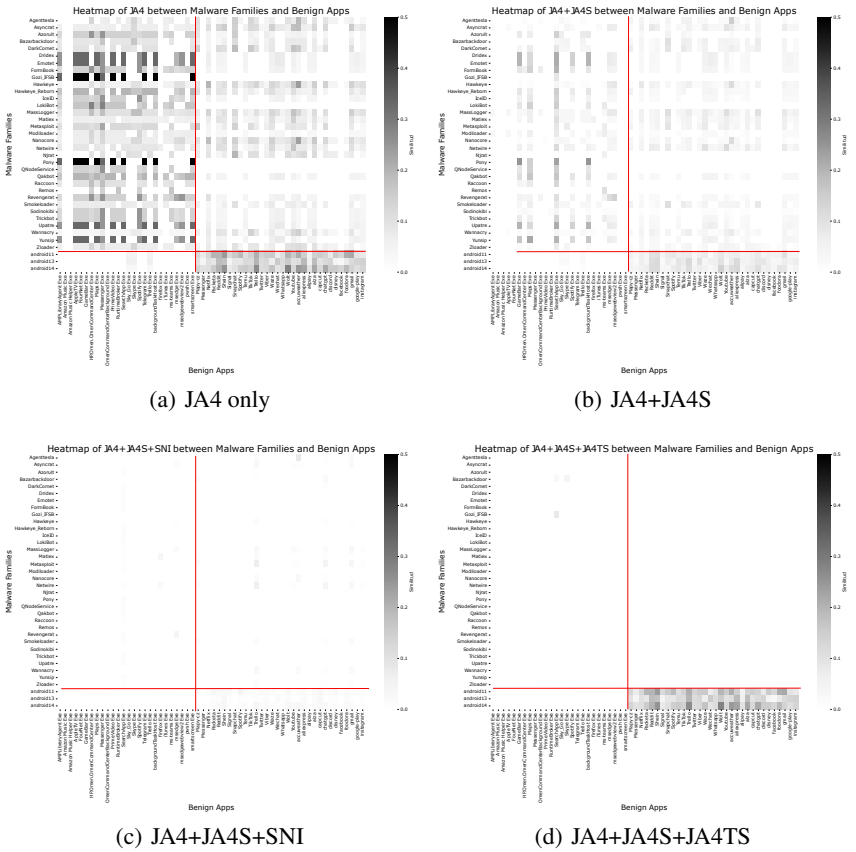
(a) JA4 only (b) JA4+JA4S

(c) JA4+JA4S+SNI (d) JA4+JA4S+JA4TS

Fig. 2. JA4+ fingerprint heatmaps for different malware families.

Figures 2 and 3 illustrate the distribution of JA4+ fingerprints across malware families and between malware and benign applications, respectively, using the second dataset from [11]. These heatmaps provide visual confirmation of the clustering effect observed in fingerprint-based models and help to justify their high performance despite reduced input dimensionality. Notably, combinations that include {JA4S, SNI}, and {JA4S, JA4TS} result in tighter and more distinguishable clusters across families, suggesting that these components carry discriminative features relevant for classification. Conversely, JA4-only representations show more overlap between classes, particularly when comparing malware against benign traffic. This emphasizes the advantage of using composite fingerprints to enhance separation between classes.

Furthermore, empirical analysis shows that JA4TS is effective in differentiating between diverse malware families and benign applications, as distinct TCP stack behaviors often leave unique fingerprints. This can be visually confirmed in the heatmaps (e.g., Figure 3), where JA4TS-enhanced clusters show clearer separation. However, it is also notable that JA4TS may struggle with certain types of mobile malware. This



(a) JA4 only (b) JA4+JA4S

(c) JA4+JA4S+SNI (d) JA4+JA4S+JA4TS

Fig. 3. Comparison of JA4+ fingerprints between malware and benign applications. Upper rows are PC malware and lower rows are mobile phone malware. Left columns are PC applications and right columns are mobile apps. Red lines divide each case.

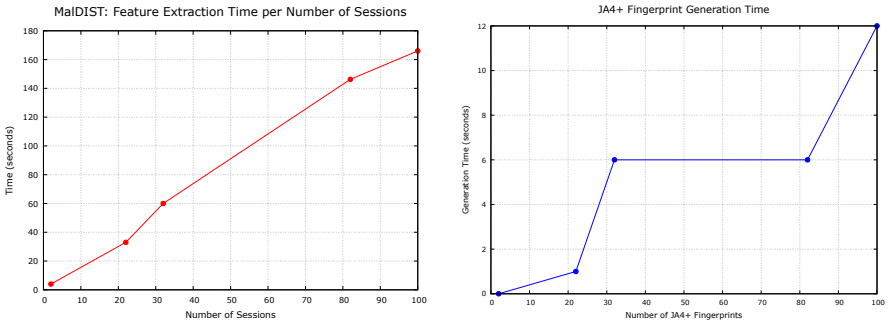
limitation arises because mobile operating systems often employ more uniform or constrained TCP/IP stack implementations, leading to less distinct fingerprint variations among different applications or malware running on them.

These results emphasize the importance of a multi-faceted approach. As the network security landscape evolves towards greater privacy, with initiatives like Encrypted Client Hello potentially encrypting SNI fields in future TLS versions, JA4TS provides a vital, independent layer of network fingerprinting that does not rely on TLS-specific metadata [7]. This makes JA4TS a robust component for maintaining visibility in increasingly encrypted traffic flows.

4.3 Execution Time Analysis

Processing and inference times were measured for each model. Figures 4 and 5 show the comparison in time required for data preprocessing and model inference. Experiments

were executed on a laptop computer equipped with an Intel i7-1065G7[5] , 16 GB RAM, Ubuntu 22.04, Python 3.10.



(a) MalDIST-inspired model flow separation and feature extraction time per number of flows

(b) JA4+ fingerprints construction time per number of fingerprints

Fig. 4. Average data processing time per flow for each model.

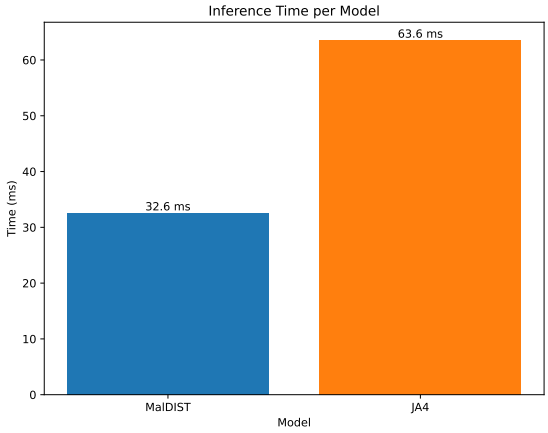


Fig. 5. Average prediction time per flow for each model.

As depicted in Figure 4, the JA4+ model exhibited significantly faster preprocessing and feature extraction times compared to the MalDIST-inspired model. This is primarily because JA4+ focuses on extracting lightweight metadata from the first packets of each

[5] https://www.intel.com/content/www/us/en/products/sku/196597/intel-core-i71065g7-processor-8m-cache-up-to-3-90-ghz/specifications.html

connection, whereas the MalDIST-inspired approach involves more extensive statistical calculations over the entire flow. Despite MalDIST-inspired model potentially having a lower inference time per flow, the substantial reduction in preprocessing time for JA4+ results in a much lower global time (total time from raw traffic to prediction), making it more efficient for real-time or high-throughput environments. It should also be noted that the case for 30 sessions in JA4+ is larger than expected. This is because in this case there are more JA4X fingerprints to be generated than in the other cases, which were calculated together with other JA4+ signatures (although we finally considered them unnecessary, as certificates are usually encrypted in latest TLS versions).

Figure 5 further illustrates that both models maintained reasonable inference times, making them suitable for near real-time and offline analysis scenarios. In this case, the MalDIST-inspired model is faster because it has a lower dimensionality than the 1024 dimensions generated by the `FeatureHasher` in the JA4+ fingerprint model. However, the overall efficiency gain from the faster preprocessing of JA4+ reinforces its suitability for lightweight deployment, such as inline detection systems.

5 Discussion

Table 2 summarizes key characteristics and performance.

Table 2. Summary of the models evaluated in this work

Model	Features	Algorithm	Accuracy	Extraction Time	Inference Time
MalDIST-inspired	198 statistics	Random Forest	98.28%	1,65 s/flow	32.6 ms/flow
JA4+ fingerprint	JA4, JA4S, SNI		98.25%	0,12 s/flow	63.6 ms/flow
JA4+ fingerprint	JA4, JA4S, JA4TS		96.28%		

The evaluation results reveal several important trade-offs. The JA4+-based model achieved classification accuracy comparable to the statistical model, but with much faster feature extraction. This makes it particularly well-suited for deployment in resource-constrained or time-sensitive environments, such as edge devices or IoT networks. The MalDIST-inspired statistical model, while slightly more accurate, incurs higher processing costs due to the complexity of extracting 198 features per flow. It remains useful for offline analysis and scenarios where model interpretability is essential, as it provides insights through feature importance rankings.

Notably, the confusion matrix highlights the difficulty of distinguishing between specific families like Valak and Dridex, suggesting the need for more discriminative features or additional context. The JA4+ heatmaps confirm that fingerprint combinations involving SNI and JA4S enhance class separation. However, their reliance on TLS metadata like SNI introduces a potential point of fragility as future TLS versions may encrypt these fields with ECH. This is precisely where JA4TS offers a notable advantage as the most robust solution for future encrypted traffic challenges. By deriving fingerprints from TCP SYN-ACK packets, JA4TS provides a resilient mechanism to

infer TCP/IP stack characteristics, which can be highly indicative of malware behavior, regardless of TLS layer encryption. This makes the full JA4 suite (JA4+JA4S+JA4TS) a more future-proof and comprehensive approach for encrypted traffic analysis.

Overall, the results confirm that metadata-based fingerprinting (JA4+) offers an effective and lightweight detection method. While hybrid models can incorporate diverse features, simplicity and robustness favor JA4+ for real-world, encrypted traffic analysis. This reinforces the role of TLS metadata as a viable foundation for sustainable and secure network malware detection.

Another important point concerns evasion and obfuscation. Techniques such as fingerprint spoofing, traffic morphing, or deliberate TLS handshake mutations may allow sophisticated malware to disguise itself and bypass classifiers. While these methods were not addressed in our experiments, they represent real-world challenges and underline the need for adaptive models and continued evaluation under adversarial scenarios.

6 Conclusion

This work presented the design and evaluation of machine learning models to compare two malware traffic detection approaches: a MalDIST-inspired statistical model and a JA4+-based fingerprint classifier. Experiments on a curated dataset of malware families and benign traffic revealed that models can achieve up to 98% accuracy in binary classification, with the fingerprint-based model offering the best trade-off between speed and accuracy. By leveraging readily extractable TLS fingerprints like JA4 and metadata such as SNI or the TCP handshake options, it is possible to build efficient classifiers suitable for real-world network defense scenarios.

Notably, JA4+ fingerprinting alone emerges as a practical and effective approach. It achieves a favorable trade-off between detection accuracy and computational efficiency, while preserving privacy by avoiding payload inspection. This makes it particularly suitable for real-world deployment in environments where encrypted traffic is prevalent and low-latency or resource-constrained processing is required, such as in IoT and IIoT networks. The ability to detect malware using only TLS metadata underscores the potential of lightweight, metadata-driven models for sustainable and secure network protection.

Future work may explore the resilience of these approaches under adversarial conditions, including evasion techniques such as TLS handshake mutation, or fingerprint spoofing. From an architectural standpoint, future research could explore novel methods for processing and encoding JA4+ fingerprints, leveraging their structure to further enhance their discriminative power for malware prediction, moving beyond traditional hashing or feature vectorization to more advanced representations. Moreover, other fingerprints can be necessary to better classify mobile malware. Assessing deployment on real-time systems—such as inline traffic analyzers or SIEM platforms—would help validate applicability in operational environments. In particular, extending the dataset to include more recent malware families (e.g., ransomware or mobile threats) and benign applications would improve generalizability.

Code availability. All scripts and models are available at: `https://github.com/fingopolo/MalJA4DIST`

Acknowledgement. This work was supported in part by the R&D activity program with reference TEC-2024/COM-504 and acronym RAMONES-CM, granted by the Comunidad de Madrid through the Directorate General for Research and Technological Innovation via Order 5696/2024.

References

1. Althouse, J.: JA4+ network fingerprinting. FoxIO via Medium (2023), https://medium.com/foxio/ja4-network-fingerprinting-9376fe9ca637, accessed: 2025-05-29
2. Althouse, J., Atkinson, J., Atkins, J.: TLS fingerprinting with JA3 and JA3S. Salesforce Engineering Blog (2017), https://engineering.salesforce.com/tls-fingerprinting-with-ja3-and-ja3s-247362855967/, accessed: 2025-05-29
3. Bader, O., Lichy, A., Hajaj, C., Dubin, R., Dvir, A.: MalDIST: From encrypted traffic classification to malware traffic detection and classification. In: IEEE 19th Annual Consumer Communications & Networking Conference (CCNC). pp. 527–533. IEEE (2022)
4. Canadian Institute for Cybersecurity: VPN-nonVPN dataset (ISCXVPN2016). https://www.unb.ca/cic/datasets/vpn.html (2016), accessed: 2025-05-29
5. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority oversampling technique. Journal of artificial intelligence research **16**, 321–357 (2002)
6. F5: Detect Encrypted Malware with SSL Inspection, [Online] Available: https://www.f5.com/resources/library/encrypted-threats/ssl-visibility/encrypted-malware [Accessed 2025-05-29]
7. Fernández-Terrasa, M., López de Vergara, J.E., Gómez-Arribas, F.J., de Pedro, L., González, I.: Mobile application identification in encrypted traffic using JA4+ fingerprints. In: Proceedings of the 4th International Conference on Computing, IoT and Data Analytics, ICCIDA 2025. Madrid, Spain (Jul 2025)
8. Fox-IT: JA4TS - extending JA4 for transport streams. https://blog.foxio.io/ja4t-tcp-fingerprinting (2024), accessed: 2025-06-30
9. Luis-Bisbé, E., Morales-Gómez, V., Perdices, D., López de Vergara, J.E.: No pictures, please: Using explainable artificial intelligence to demystify CNNs for encrypted network packet classification. Applied Sciences **14**(13) (2024). https://doi.org/10.3390/app14138224
10. malware-analysis.net: Malware sample repository. https://www.malware-analysis.net (2025), accessed: 2025-05-29
11. Matoušek, P., Ryšavy, O., Burgetová, I.: Experience report: Using JA4+ fingerprints for malware detection in encrypted traffic. In: 2024 20th International Conference on Network and Service Management (CNSM). pp. 1–5. IEEE (2024)
12. Paracha, M.T., Dubois, D.J., Vallina-Rodriguez, N., Choffnes, D.: IoTLS: understanding TLS usage in consumer IoT devices. In: Proceedings of the 21st ACM Internet Measurement Conference. p. 165–178. IMC '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3487552.3487830
13. Tria.ge: Tria.ge malware sandbox. https://tria.ge (2025), accessed: 2025-05-29
14. Yang, L., Fu, S., Wang, Y., Liang, K., Mo, F., Liu, B.: Dev-eta: An interpretable detection framework for encrypted malicious traffic. The Computer Journal **66**(5), 1213–1227 (03 2022). https://doi.org/10.1093/comjnl/bxac008, https://doi.org/10.1093/comjnl/bxac008