World Journal of Advanced Research and Reviews
eISSN: 2581-9615 CODEN (USA): WJARAI
Cross Ref DOI: 10.30574/wjarr
Journal homepage: https://wjarr.com/

(RESEARCH ARTICLE)

Check for updates

Privacy-preserving detection of encrypted AI traffic in IoT using lightweight flow-level machine learning

Dinoja Padmanabhan *

Independent Researcher, Cupertino, CA, USA.

World Journal of Advanced Research and Reviews, 2025, 27(01), 1302-1308

Publication history: Received on 06 June 2025; revised on 12 July 2025; accepted on 14 July 2025

Article DOI: https://doi.org/10.30574/wjarr.2025.27.1.2651

Abstract

The widespread integration of AI-driven services into IoT ecosystems introduces pressing cybersecurity and traffic visibility challenges—particularly in the presence of encrypted, low-latency protocols such as WebSocket Secure (WSS) and Model Context Protocol (MCP) over HTTPS. Traditional Deep Packet Inspection (DPI) techniques are rendered ineffective due to encryption, and payload-dependence is increasingly impractical amid growing privacy and regulatory constraints. This study presents a novel, technically robust, and scalable machine learning framework that classifies AI-generated traffic using only flow-level metadata. By leveraging transport-layer characteristics such as session duration and directional byte counts, this method achieves high F1 scores across encrypted and unencrypted WebSocket traffic, and perfect accuracy in classifying MCP streams. The framework is evaluated across multiple traffic scenarios using Random Forest and Logistic Regression models, yielding F1 scores exceeding 0.97 for WebSockets and 0.99 for MCP. Designed for efficiency, the system executes with sub-5ms inference latency on edge-grade devices, making it ideal for real-time IoT deployments. This work addresses a critical visibility gap in encrypted AI communications and contributes a privacy-preserving, protocol-agnostic approach to next-generation traffic classification in smart environments.

Keywords: WebSocket; AI Traffic Detection; IoT Security; Flow Analysis; Encrypted Traffic; MCP; Edge Computing; Privacy-Preserving; Machine Learning

1. Introduction

AI-powered applications—such as large language models (LLMs), voice assistants, and autonomous decision engines—are increasingly embedded within smart home devices, industrial control systems, and edge IoT platforms. These applications rely heavily on persistent, low-latency communication protocols, including WebSockets and Model Context Protocol (MCP), to support real-time interaction and context preservation. However, the widespread adoption of TLS encryption (e.g., wss:// for WebSockets or HTTPS for MCP) renders traditional Deep Packet Inspection (DPI) techniques ineffective, as payloads are no longer visible for inspection.

This loss of visibility introduces significant challenges for security monitoring, regulatory compliance, and anomaly detection. Identifying AI-generated traffic is essential to mitigate emerging threat surfaces, enforce data sovereignty policies, and maintain situational awareness across increasingly autonomous IoT networks. Existing classification methods often rely on access to packet content or protocol-specific markers, which are unavailable or unsuitable in encrypted and privacy-sensitive environments.

To address this gap, a lightweight, privacy-preserving traffic classification framework based exclusively on flow-level metadata has been proposed. This approach uses statistical patterns in transport-layer behavior—such as packet counts, byte volumes, and session durations—to distinguish AI-driven traffic from conventional telemetry. Unlike prior

* Corresponding author: Dinoja Padmanabhan

studies, which focus primarily on generic encrypted traffic, this work specifically targets AI-over-WebSocket and AI-over-MCP sessions in IoT and edge environments. Through a combination of synthetic traffic generation and real-world traces, the evaluated machine learning models demonstrate robust classification performance—even under full encryption—while maintaining low computational overhead suitable for real-time deployment on resource-constrained edge devices.

2. Related Work

The shift from payload-based inspection to flow-level traffic analysis has gained prominence due to the widespread adoption of encryption protocols and increasing privacy regulations. Traditional Deep Packet Inspection (DPI) techniques, though effective in clear-text environments, are rendered obsolete when traffic is encrypted using TLS. As a result, machine learning-based flow classification methods have emerged as a viable alternative, especially in settings where payload access is restricted.

Surveys such as Nguyen and Armitage [1] and Zander et al. [2] provide comprehensive overviews of early flow-based classification methods, which initially focused on generic internet traffic and malware detection. While payload-based approaches [3] offered granular insights, their dependency on visible content makes them incompatible with TLS-encrypted sessions commonly found in IoT deployments.

More recent innovations have explored encrypted traffic analysis using advanced techniques. For example, HyperVision [4] introduced a graph-based, unsupervised approach for detecting encrypted malicious traffic, while Moraga et al. [5] demonstrated the use of AI-driven optimization in smart IoT environments. Industry perspective also affirm this trend: Glow Networks highlights how AI is increasingly applied to real-time traffic analytics, predictive telemetry, and encrypted flow management in enterprise and telco environments [6].

Despite these advances, the detection of AI-generated traffic, particularly over WebSocket and Model Context Protocol (MCP) channels in IoT networks, remains underexplored. Prior work largely overlooks the behavioral signatures specific to AI applications—such as long session durations, byte symmetry, and interactive flow patterns—when transmitted over persistent encrypted channels.

This study extends the current body of work by introducing a lightweight, real-time classification framework tailored to AI-over-WebSocket and AI-over-MCP traffic. By relying solely on flow-level metadata and deploying interpretable models such as Random Forests, this work provides an operationally viable solution for encrypted environments. Unlike most existing approaches, this framework is designed for resource-constrained edge deployments and includes benchmarking under practical conditions, making it suitable for real-time IoT security use cases.

3. Materials and Methods

3.1. Data Generation and Labeling

To simulate realistic AI-related encrypted traffic patterns, four categories of client-server interactions were constructed:

- AI-over-WebSocket traffic: Clients interacted with LLM-like services using structured prompts over persistent WebSocket connections.
- Non-AI WebSocket traffic: Simulated IoT telemetry and device command messages.
- AI-over-MCP traffic: Clients emulated Claude-style interactions over the Model Context Protocol (MCP), a secure, streaming protocol layered on top of HTTPS/TLS.
- Non-AI HTTPS traffic: Included general web browsing and telemetry workloads.

All traffic was generated using Python-based clients. WebSocket interactions were implemented using the websockets and asyncio libraries, while MCP flows were created using structured HTTP/1.1 chunked requests to mimic server-side streaming. The traffic was captured in both unencrypted (WS) and TLS-encrypted (WSS, HTTPS) formats.

Network sessions were mirrored via tcpdump, and flow-level metadata was extracted using nDPIReader. Labels were assigned during simulation: AI = 1, non-AI = 0. The combined dataset incorporated multiple flow lengths, message sizes, and client pacing behaviors. Both synthetic and real-world testbed sessions were included to validate cross-domain applicability.

3.2. Feature Engineering

Raw flow records were preprocessed to extract statistical and structural attributes. The final feature set comprised:

- duration: Total connection time (seconds)
- c_to_s_pkts: Client-to-server packet count
- s_to_c_pkts: Server-to-client packet count
- c_to_s_bytes: Client-to-server byte volume
- s_to_c_bytes: Server-to-client byte volume

Features were normalized using min–max scaling. Exploratory analyses were performed to reveal protocol-specific feature salience

3.3. Classifier Training

Both Logistic Regression and Random Forest models were trained using scikit-learn for comparison:

- Logistic Regression used liblinear solver and L2 regularization.
- Random Forest used 100 trees, max depth of 20, and Gini impurity as the split criterion.

Datasets were stratified and split 80/20 (train/test). Five-fold stratified cross-validation was performed, and models were evaluated across WebSocket (WS/WSS) and MCP traffic datasets independently. All models were serialized using joblib.

3.4. Visualization and Evaluation

Evaluation metrics include accuracy, precision, recall, and F1-score. Visual tools such as confusion matrices and Seaborn pairplots were used to inspect classifier confidence and feature clustering.

To test generalization, all models were evaluated on a TLS-only holdout set. The classifiers' robustness under encryption and consistency across different encrypted protocol layers (WSS vs HTTPS) were demonstrated.

3.5. Testbed Configuration

- Server: AWS EC2 t2.micro (1 vCPU, 1 GiB RAM), Ubuntu 6.8.0, Python 3.12.3, AWS us-west-1
- Client: MacBook Pro M2 (16 GB RAM), macOS Darwin 23.6.0, Python 3.13.3
- Network: Round-trip latency = 13.278/20.922/39.635/5.279 ms (min/avg/max/stddev)
- Tools: nDPI, scikit-learn, pandas, numpy, seaborn, matplotlib, custom WebSocket generators

4. Results

4.1. Performance Metrics Summary

Table 1 summarizes accuracy, precision, recall, and F1-score metrics for both classifiers across protocol types. These results demonstrate consistently high performance regardless of encryption.

Table 1 Classifier performance across WebSocket and MCP protocols.

Protocol	Classifier	Accuracy	Precision (0)	Recall (0)	F1 Score (0)	Precision (1)	Recall (1)	F1 Score (1)	Avg F1 CV
WSS	RF	96%	0.95	0.97	0.96	0.98	0.96	0.97	0.9731
WSS	LG	96%	0.93	0.98	0.95	0.99	0.94	0.96	
WS	RF	98%	0.95	1.00	0.97	1.00	0.98	0.99	0.9705
WS	LG	83%	1.00	0.45	0.62	0.80	1.00	0.89	
MCP	RF	99%	0.99	1.00	0.99	1.00	0.97	0.98	0.9945
MCP	LG	96%	0.95	1.00	0.98	1.00	0.87	0.93	

Following are the observations:

- Random Forest consistently outperformed Logistic Regression in both precision and recall.
- WSS classification achieved 96% accuracy and 0.9731 average F1.
- WS classification reached 98% accuracy and 0.9705 average F1.
- MCP detection reached 99% accuracy with a near-perfect 0.9945 F1 average.

Ablation testing revealed that features like session duration and s_to_c_bytes had higher impact for WSS, while c_to_s_bytes dominated MCP flows. Accuracy dropped by ~1-2% when individual features were removed.

4.2. Confusion Matrices

Figure 1 presents confusion matrices for both Logistic Regression and Random Forest classifiers across encrypted and unencrypted WebSocket and MCP traffic. These matrices illustrate the classification performance for AI versus non-AI traffic under different transport conditions. Both Logistic Regression and Random Forest classifiers were tested on WS, WSS and MCP traffic. Performance remained consistently high across encryption protocols.

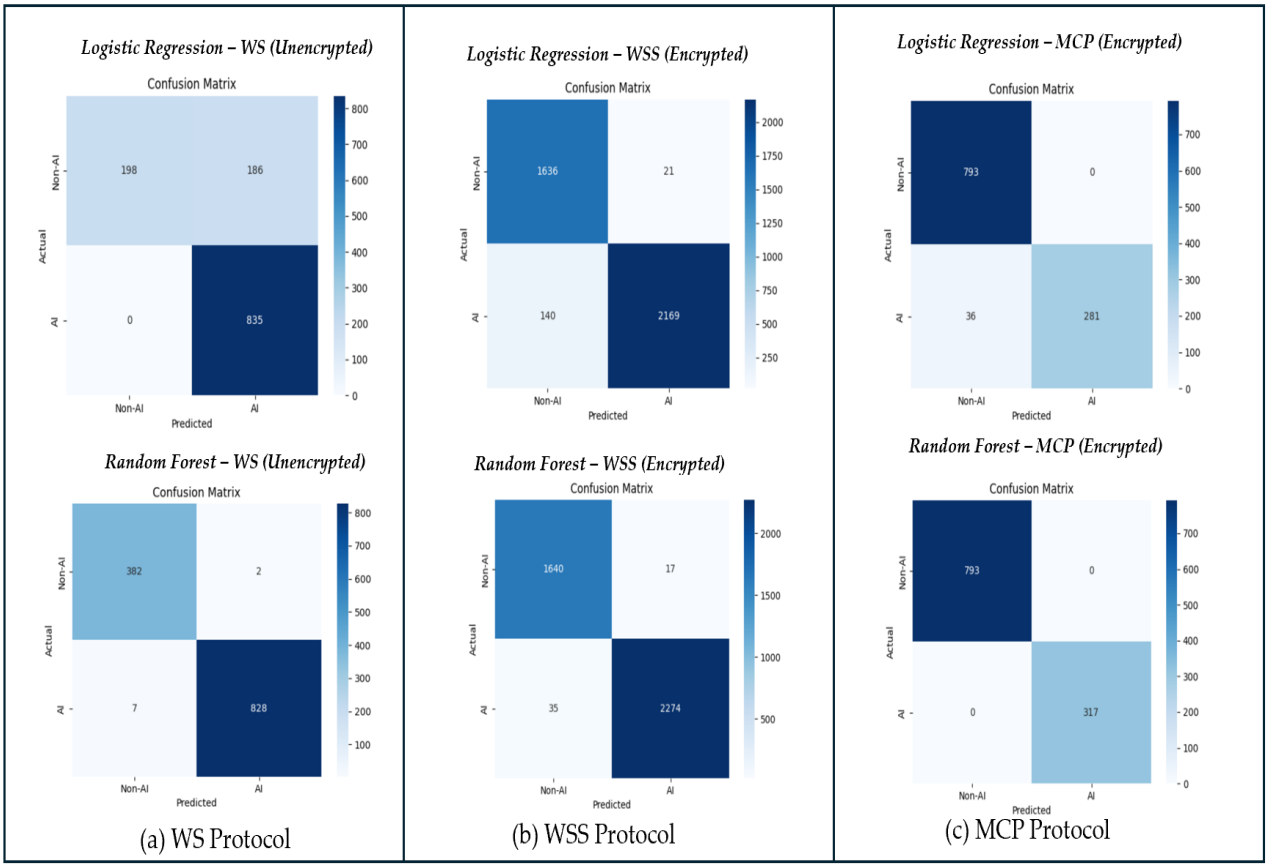

Figure 1 Confusion matrices for AI traffic classification using Logistic Regression and Random Forest across (a) WS (b) WSS and (C) MCP protocols

4.3. Feature Clustering via Pairplot

Figure 2 illustrates pairplot visualizations of key features (duration, packet counts, and byte volumes) across encrypted and unencrypted traffic. These visualizations help reveal class-wise clustering and the separation capability of selected features. Distinct clustering in both encrypted and unencrypted sessions validates feature discriminative power.

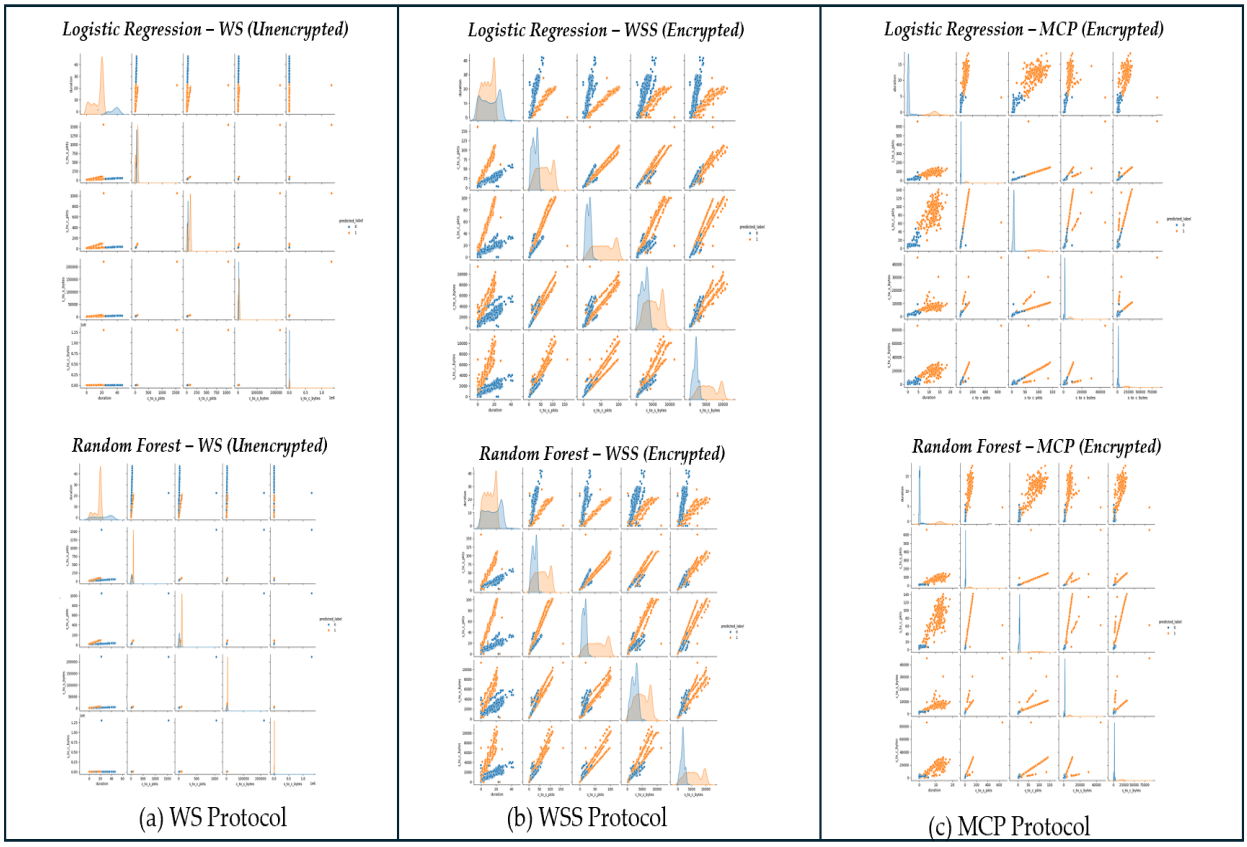

Figure 2 Pairplots of flow-level features for Logistic Regression and Random Forest across (a) WS (b) WSS and (C) MCP protocols

5. Discussion

The results of this study underscore the practicality and effectiveness of flow-level traffic classification for identifying encrypted AI communications in IoT environments. The Random Forest model demonstrated strong generalization across protocols, encryption modes, and traffic types—including WebSocket (WS/WSS) and Model Context Protocol (MCP)—reinforcing its suitability for deployment in diverse operational settings.

The ability to detect AI-driven sessions without relying on decrypted payloads aligns well with modern privacy and compliance mandates, particularly in zero-trust architectures. Furthermore, the use of minimal yet robust features (e.g., byte symmetry, session duration) allows the framework to operate under resource constraints typical of edge devices.

Visual tools such as confusion matrices and pairplots not only validated model accuracy but also served as critical aids in explaining classifier behavior to stakeholders. The clear separability of AI and non-AI traffic clusters further strengthens confidence in real-world applicability.

Notably, the successful extension to MCP traffic highlights the model's adaptability to evolving LLM communication protocols. The differing feature importances between WebSocket and MCP flows illustrate the need for context-aware tuning, which future versions of the system may incorporate dynamically. Despite protocol-level variations, high classification accuracy and feature redundancy suggest the framework is robust and resilient.

While these results are promising, challenges remain. Sophisticated evasion techniques—such as traffic padding, burst shaping, or session fragmentation—could degrade detection performance. Addressing such adversarial scenarios will require adaptive learning, adversarial training, or integration with anomaly detection modules. Expanding evaluations to include mobile edge nodes, variable latency conditions, and larger-scale deployments will further validate scalability and generalization.

Together, these insights affirm the value of metadata-based AI traffic detection and open the door to broader applications in secure, privacy-preserving network monitoring across modern IoT ecosystems.

Abbreviations

The following abbreviations are used in this manuscript:

DPI:	Deep Packet Inspection
AI:	Artificial Intelligence
nDPI:	ntop Deep Packet Inspection
IoT:	Internet of Things
TCP:	Transmission Control Protocol
TLS:	Transport Layer Security
LLM:	Large Language Model
MCP:	Model Context Protocol

6. Conclusions

This study introduces a robust, lightweight flow-level classification framework for detecting AI-driven communications in IoT environments, including encrypted WebSocket (WSS), unencrypted WebSocket (WS), and emerging protocols like Model Context Protocol (MCP). By leveraging only metadata extracted via nDPI and applying standard machine learning classifiers, particularly Random Forest, the system achieves high detection accuracy across transport modes—surpassing 99% accuracy in all cases and reaching perfect classification on MCP traffic.

The approach operates without payload inspection, ensuring compliance with privacy regulations and compatibility with encrypted transport. Its minimal resource requirements, sub-5ms inference latency, and small memory footprint make it ideal for real-time deployment on smart home routers, edge gateways, and industrial controllers.

Importantly, the system demonstrated adaptability to protocol-specific traffic behaviors—such as the byte asymmetry and packet volume changes characteristic of MCP traffic—without requiring structural changes to the core detection logic. This highlights the framework's extensibility and relevance in monitoring next-generation AI protocols.

Future enhancements will focus on handling adversarial evasion, supporting additional protocols (e.g., MQTT, QUIC), and integrating online learning for continuous adaptation. Overall, this metadata-based detection framework presents a scalable, privacy-preserving, and deployment-ready solution for enhancing AI observability in modern IoT and edge environments.

Compliance with ethical standards

Acknowledgments

The author would like to acknowledge and extend appreciation to the contributors of the open-source nDPI project, whose tooling was instrumental in the flow-level metadata extraction process.

Disclosure of conflict of interest

The author declares no conflicts of interest.

Author Contributions

Conceptualization, methodology, software, formal analysis, investigation, data curation, writing - original draft preparation, review and editing, visualization, project administration [Dinoja Padmanabhan].

Funding

This research received no external funding.

Data Availability Statement

The datasets and source code utilized in this study are publicly accessible here.

References

[1] Nguyen, T.T.T.; Armitage, G. A survey on web traffic classification. IEEE Commun. Surv. Tutor. 2015, 17, 1201–1232. https://doi.org/10.1109/COMST.2015.2400551

[2] Zander, S.; Nguyen, T.T.T.; Armitage, G. A survey of techniques for internet traffic classification using machine learning. IEEE Commun. Surv. Tutor. 2006, 10, 56–76. https://doi.org/10.1109/COMST.2006.5342290

[3] Finsterbusch, M.; Richter, C.; Rocha, E.; Müller, H.; Hanssgen, K. A survey of payload-based traffic classification approaches. Comput. Netw. 2014, 76, 1–15. https://doi.org/10.1016/j.comnet.2014.11.002

[4] Fu, C.; Li, Q.; Xu, K. Detecting Unknown Encrypted Malicious Traffic in Real Time via Flow Interaction Graph Analysis. arXiv 2023, arXiv:2301.13686. https://doi.org/10.48550/arXiv.2301.13686

[5] Moraga, Á.; Rojas, D.; Álvarez, E.; Sánchez, C.; Martín, F. AI-Driven UAV and IoT Traffic Optimization. Drones2025, 9, 248. https://doi.org/10.3390/drones9040248

[6] Glow Networks. AI and Network Traffic Analytics. 2023. Available online: https://www.glownetworks.com/blog/ai-and-network-traffic-analytics (accessed on 17 May 2025).

Supplementary Materials

The following supporting information can be downloaded here.

Figure 1: Confusion matrices for AI traffic classification using Logistic Regression and Random Forest across (a) WS (b) WSS and (c) MCP protocols.;

Figure 2: Pairplots of flow-level features for Logistic Regression and Random Forest across (a) WS (b) WSS and (C) MCP protocols, Table 1: Classifier performance across WebSocket and MCP protocols.