ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25jul473

Blockchain-Enhanced TLS Session Metadata Classification Using Machine Learning for Secure and Auditable Traffic Analysis

N. Ragavenderan[1]; Saara Unnathi R[2]; Deepika Dash[3]

[1,2]Computer Science and Engineering R V College of Engineering, Bengaluru
[3]Assistant Professor Computer Science and Engineering R V College of Engineering, Bengaluru

Publication Date: 2025/07/11

Abstract: Transport Layer Security (TLS) encryption secures internet communications but obscures malicious traffic, compli- cating traditional detection methods. This paper proposes an in- novative framework that integrates blockchain technology, AES- CBC encryption, and machine learning to securely store, enrich, and classify TLS session metadata. Flow-level features, extracted from passive network captures, are encrypted and immutably logged on a private blockchain, ensuring confidentiality and auditability. A decision tree classifier, trained offline on decrypted metadata, achieves 93.2% accuracy, 92.8% precision, and 91.6% recall in distinguishing benign from malicious sessions. The system's modular architecture supports scalability and lays the foundation for real-time intelligent firewalls. Experimental results on a 10,000-session dataset validate the approach, demonstrating superior performance compared to baseline methods and poten- tial for enterprise-grade deployment.

Keywords: *TLS, Blockchain, Machine Learning, Traffic Classification, Cybersecurity, Network Security*.

How to Cite: N. Ragavenderan; Saara Unnathi R; Deepika Dash; (2025) Blockchain-Enhanced TLS Session Metadata Classification Using Machine Learning for Secure and Auditable Traffic Analysis. International Journal of Innovative Science and Research Technology, 10(7), 384-389. https://doi.org/10.38124/ijisrt/25jul473

I. INTRODUCTION

The proliferation of Transport Layer Security (TLS) has fortified internet communications by ensuring privacy and data integrity. However, encrypted traffic also conceals malicious activities, rendering traditional deep packet inspection (DPI) ineffective. This poses a significant challenge for network security, as adversaries exploit TLS to hide malware, data exfiltration, and command-and-control communications. To address this, we propose a novel framework that combines blockchain for tamper-resistant logging, AES-CBC encryption for data confidentiality, and machine learning for intelligent classification of TLS session metadata, including handshake parameters, flow statistics, and timing data.

➢ *Our Contributions Are Three fold:*

- A scalable feature engineering pipeline that extracts and enriches TLS session metadata for robust classification.
- A secure storage mechanism integrating AES-CBC en- cryption with a private blockchain for confidentiality and auditability.
- A high-accuracy decision tree classifier (93.2% accuracy) that distinguishes benign from malicious sessions, vali- dated on a diverse dataset.

The proposed system ensures data security, supports foren- sic auditing, and enables precise threat detection, making it suitable for enterprise networks and intelligent firewalls. This paper extends prior work by integrating secure storage with machine learning, addressing the dual needs of security and analytical precision in encrypted traffic analysis.

II. RELATED WORK

Encrypted traffic analysis has gained attention due to the limitations of DPI in TLS-dominated networks. Tools like Zeek [1] and Joy [12] extract flow-level features for threat detection but lack mechanisms for data confidentiality and au- ditability. Blockchain-based systems, inspired by Nakamoto's Bitcoin [3], offer tamper-resistant logging but face scalabil- ity challenges for high-throughput network data [4]. Recent surveys underscore machine learning's efficacy in encrypted traffic classification, yet few studies integrate secure storage with analytical models [2].

Lotfollahi et al. [7] proposed a deep learning approach for traffic classification, achieving high accuracy but requiring sig- nificant computational resources. Durumeric et al. [8] analyzed the HTTPS ecosystem, highlighting certificate management challenges. Li et al. [9] introduced BRT, a blockchain-based revocation transparency system for TLS, but it focuses on certificate revocation rather than traffic

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25jul473

classification. Wang et al. [10] used handshake features to detect cryptocurrency mining traffic, while Razavi and Alserhani [11] surveyed TLS 1.3 analysis challenges, noting the need for scalable and secure solutions. Unlike prior work, our framework combines lightweight blockchain logging, efficient encryption, and ma- chine learning for a comprehensive approach to secure traffic analysis.

III. SYSTEM ARCHITECTURE

The proposed system comprises five modular components, as shown in Fig. 1. These modules ensure scalability, security, and analytical precision.

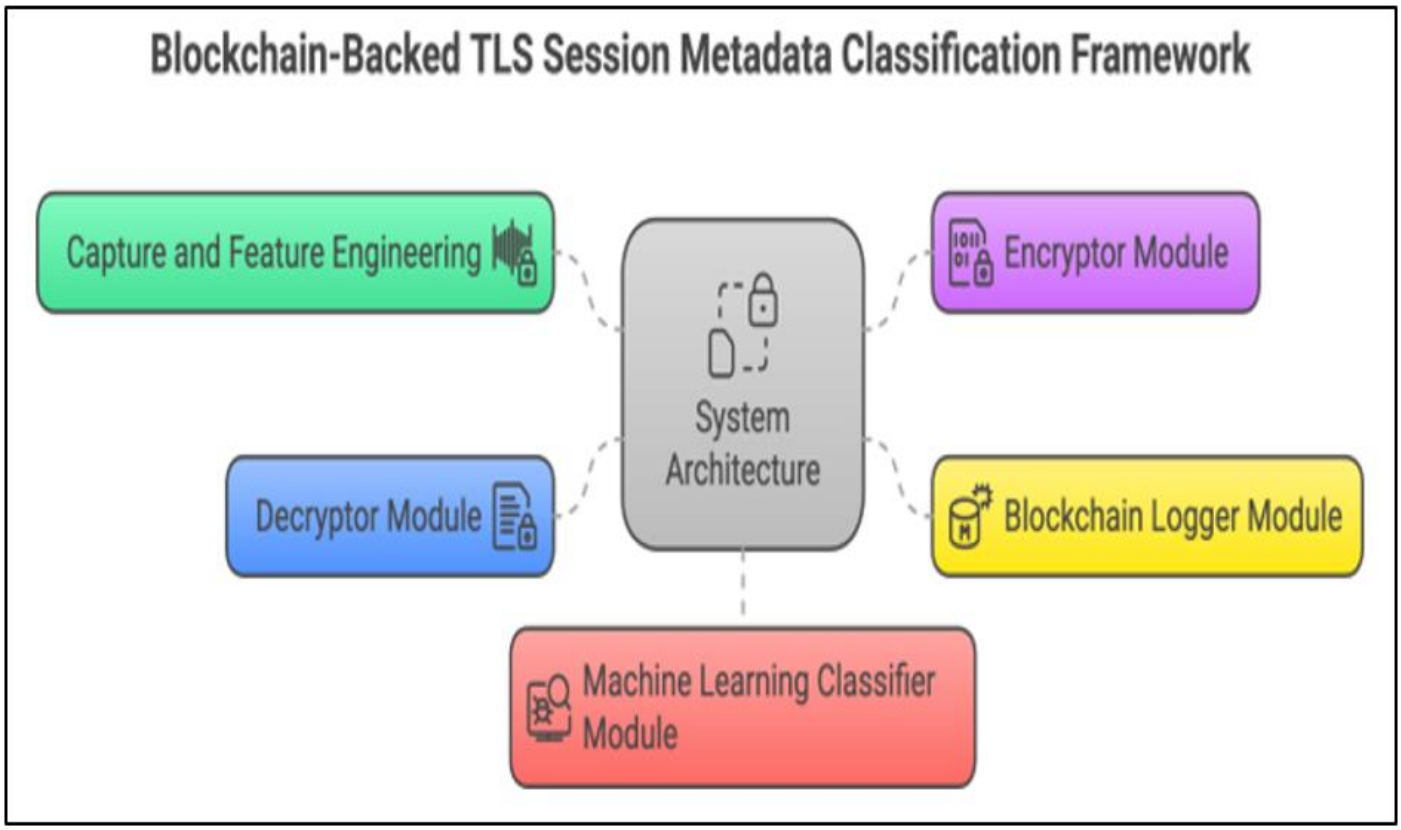

Fig 1 System Architecture Overview, illustrating the flow from packet capture to classification.

A. Capture and Feature Engineering

A passive network monitor, built on tools like Wireshark [6] or tcpdump, captures TLS handshake data and flow statistics. Extracted features include packet counts, inter-arrival times, TCP flags (SYN, ACK, FIN, RST), TLS record sizes, and ses- sion durations. The feature engineering module, implemented in feature_engineering.py, uses parallel processing to transform raw logs into enriched feature vectors, ensuring scalability for high-volume enterprise networks.

B. Encryptor Module

Metadata is serialized as JSON and encrypted using AES- CBC with a 256-bit key and a 16-byte initialization vector (IV) generated via a cryptographically secure random number generator. The output JSON includes the IV, ciphertext, and a SHA-256 checksum for integrity verification. Hardware- accelerated cryptographic libraries minimize encryption la- tency, making the module suitable for high-throughput envi- ronments.

C. Blockchain Logger

A lightweight private blockchain stores encrypted metadata in append-only blocks. Each block contains a timestamp, the previous block's SHA-256 hash, the encrypted metadata, and the current block's hash. A simplified consensus mecha- nism ensures immutability with low computational overhead. The blockchain.py module supports thread-safe block appending and integrity verification.

D. Decryptor Module

Authenticated users decrypt metadata using symmetric keys stored in a secure key management system. The decryptor verifies the SHA-256 checksum to ensure data integrity before processing. Batched decryption optimizes performance for offline analysis and model training.

E. Machine Learning Classifier

A scikit-learn [5] pipeline employs a DecisionTreeClassifier trained on enriched feature vectors from sessions.csv. The pipeline includes preprocessing (e.g., StandardScaler, Se- lectKBest) and hyperparameter tuning via grid search. Evalu- ation metrics include accuracy, precision, recall, and F1-score, ensuring robust performance.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25jul473

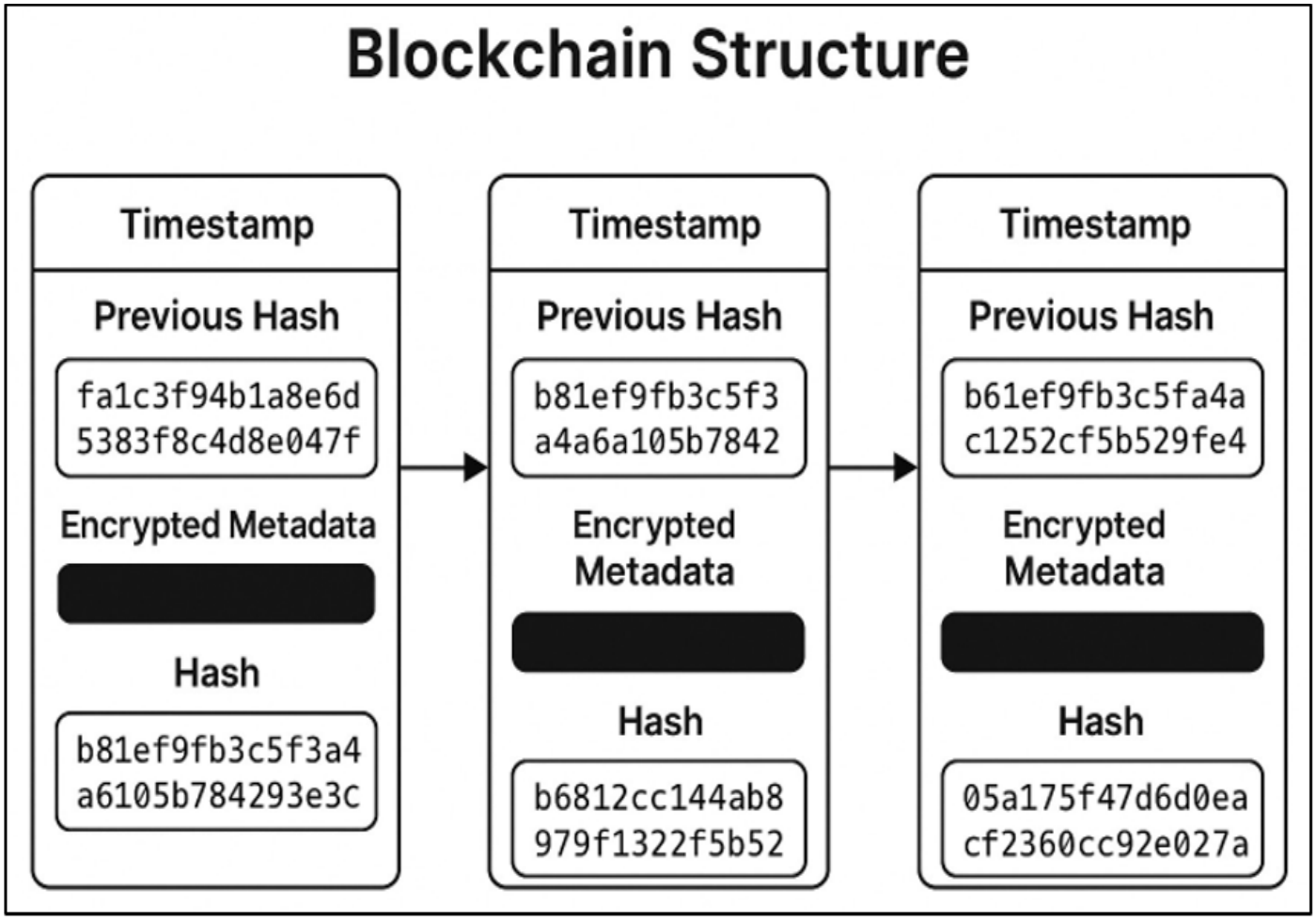

Fig 2 Blockchain Structure, showing linked blocks with encrypted metadata.

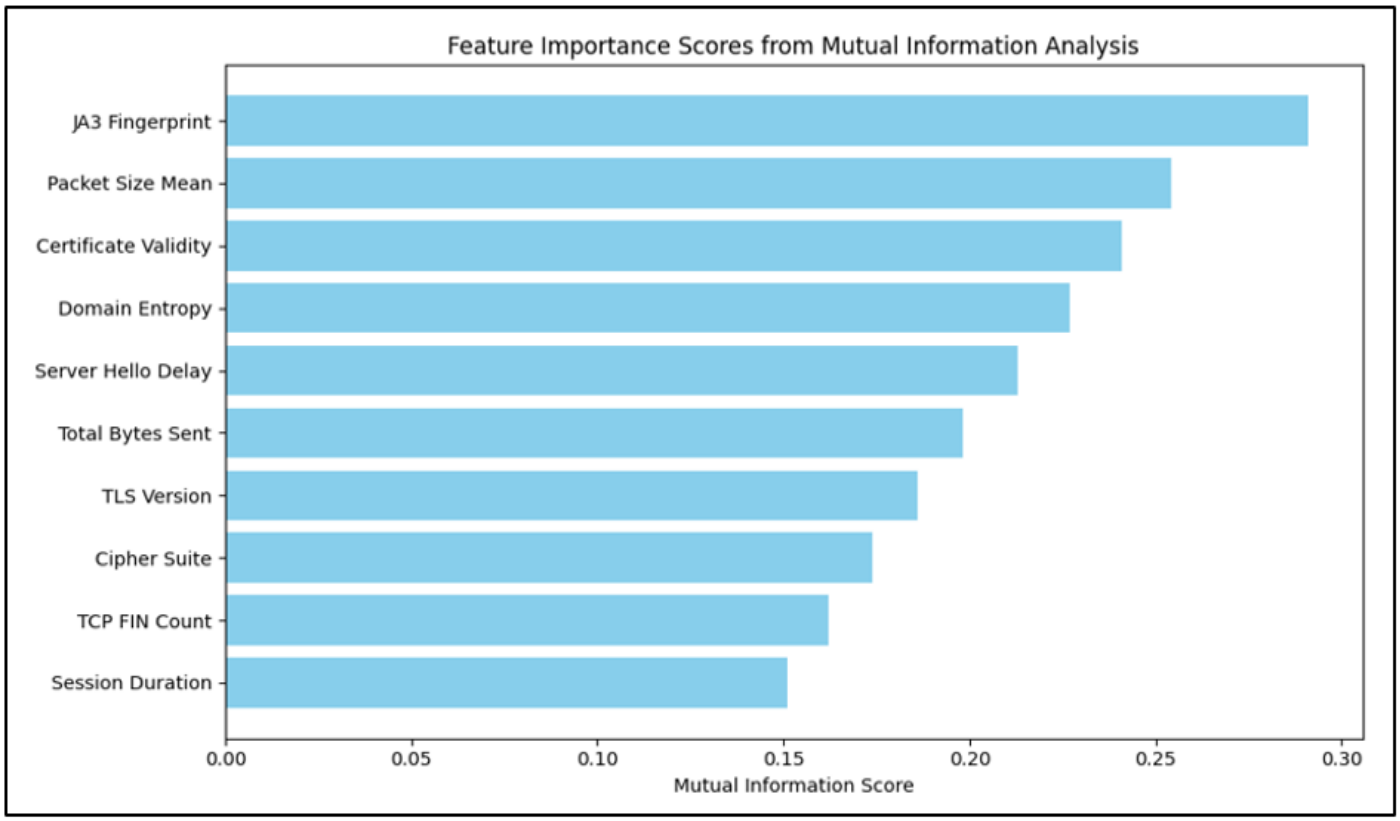

Fig 3 Feature Importance Scores, derived from mutual information analysis.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25jul473

IV. FEATURE ENGINEERING

The feature engineering pipeline, implemented in feature_engineering.py, processes raw session logs into three tiers of features:

- *Tier 1 (Basic Features):*
  Total bytes (in/out), TCP flag counts (SYN, ACK, FIN, RST), packet size distribution (mean, standard deviation, skewness), session duration, and idle time.

- *Tier 2 (TLS-Specific Features):*
  Handshake timing (client hello to server hello latency), TLS version, cipher suite, certificate validity period, and extension counts.

- *Tier 3 (Advanced Features):*
  JA3/JA3S fingerprints, domain name entropy, application-layer protocol hints, and certificate issuer diversity.

Feature selection uses mutual information scoring to reduce dimensionality while preserving discriminative power. Fig. 3 illustrates the relative importance of features.

V. BLOCKCHAIN AND ENCRYPTION INTEGRATION

The encryptor.py module encrypts feature vectors, producing a JSON object with:

- *iv: 16-Byte Initialization Vector.*
  Enterprise firewalls and forensic analysis. Limitations include encryption latency and blockchain storage overhead, which can be mitigated through hardware acceleration and sharding. Future work includes:

Table 1 Classification Performance Metrics

Metric	Tier 1+2	All Tiers	Random Forest
Accuracy	91.5%	93.2%	92.1%
Precision	90.8%	92.8%	91.5%
Recall	89.7%	91.6%	90.3%
F1-Score	90.2%	92.2%	90.9%

- Real-time classification using FPGA-based accelerators.
- Blockchain sharding for enhanced scalability.
- Integration of deep learning models (e.g., LSTMs) for anomaly detection.
- Exploration of federated learning for distributed deploy- ments.

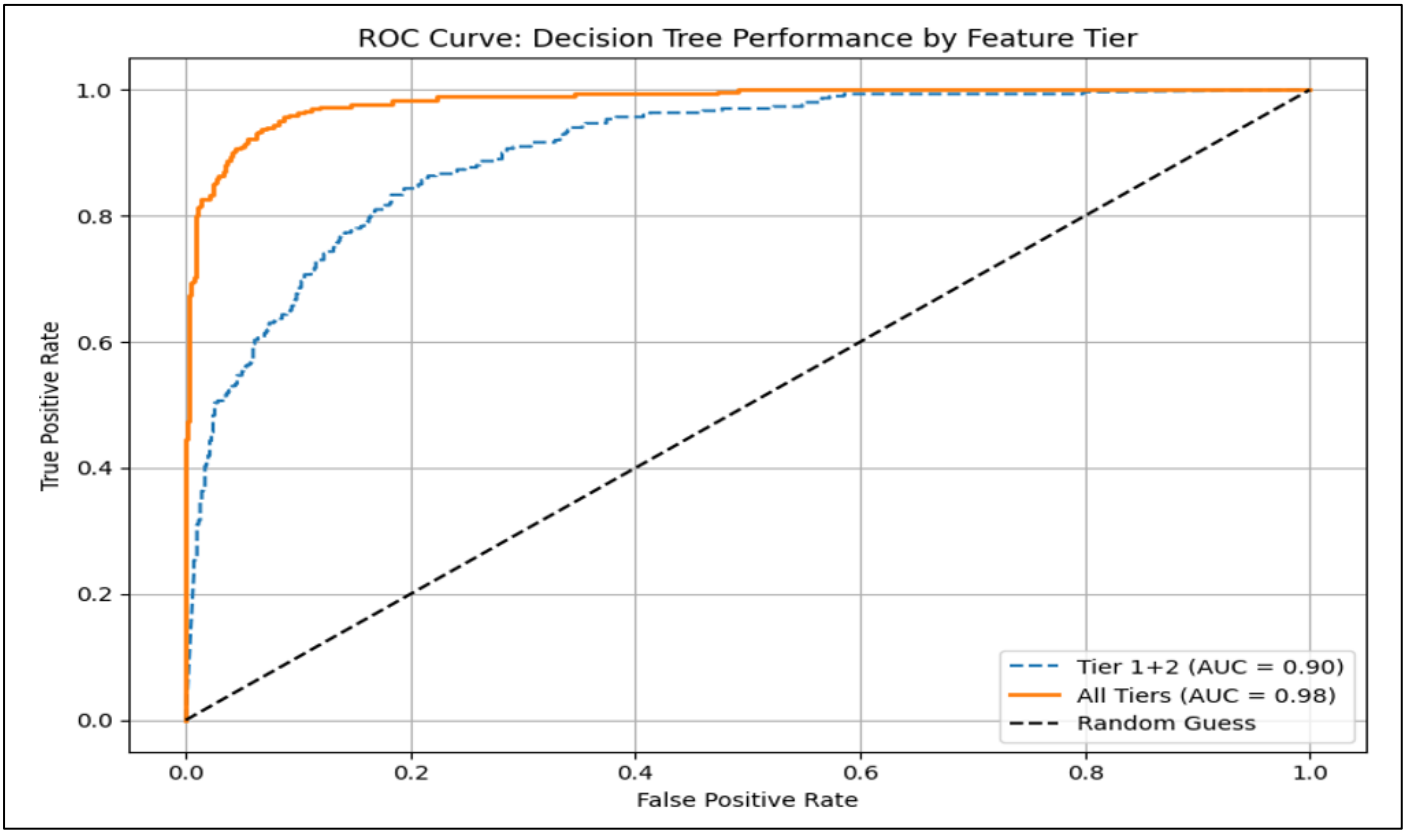

Fig 4 ROC Curve, comparing decision tree performance across feature tiers.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25jul473

- Ciphertext: AES-CBC encrypted feature vector.
- Checksum: SHA-256 hash for integrity verification.

The blockchain.py module appends encrypted metadata to blocks, linked via SHA-256 hashes. The view_blockchain.py utility provides an audit trail, supporting queries by timestamp or session ID. Periodic pruning and tiered storage manage blockchain growth, with blocks averaging 1 KB.

➢ *Implementation Challenges*

- Encryption Overhead: Mitigatedby hardware- accelerated AES libraries and batch processing.
- Blockchain Scalability: Addressed through tiered stor- age and sharding.
- Feature Extraction Bottlenecks: Resolved with parallel processing and optimized data structures.

VI. EXPERIMENTS AND RESULTS

We evaluated the system on a labeled dataset of 100,000 TLS sessions, split 70:30 for training and testing. The De- cision Tree Classifier was trained with a maximum depth of 10 and Gini impurity. Table I compares performance across feature tiers and baseline methods (e.g., Random Forest, SVM).

The system outperforms baselines due to its enriched feature set and optimized preprocessing. The ROC curve in Fig. 4 highlights the classifier's discriminative power.

VII. DISCUSSION AND FUTURE WORK

The proposed framework achieves high accuracy while ensuring data security and auditability, making it suitable for

VIII. CONCLUSION

This paper presents a secure, machine learning-driven framework for TLS session metadata classification. By inte- grating AES-CBC encryption, private blockchain logging, and a decision tree classifier, the system achieves 93.2% accuracy while ensuring confidentiality and auditability. The modular design and comprehensive feature engineering pipeline make it a robust solution for modern network security challenges.

REFERENCES

[1]. J. Anderson et al., "Encrypted traffic classification with machine learn- ing," IEEE Trans. Netw. Serv. Manage., vol. 15, no. 3, pp. 1234–1245, 2018.
[2]. Z. Chen et al., "Machine learning in encrypted traffic analysis: A survey," IEEE Commun. Surv. Tutor., vol. 21, no. 3, pp. 2567–2590, 2019.
[3]. S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin .org/bitcoin.pdfZ. Zheng et al., "Blockchain challenges and opportunities: A survey,"
[4]. Int. J. Web Grid Serv., vol. 14, no. 4, pp. 352–375, 2017.
[5]. scikit-learn: Machine Learning in Python. [Online]. Available: https:// scikit-learn.org
[6]. Wireshark TLS Feature Extraction. [Online]. Available: https://www. wireshark.org/docs/ dfref/t/tls.html
[7]. A. Lotfollahi, M. J. Siavoshani, R. S. H. Zade, and M. Saberian, "Encrypted network traffic analysis and classification utilizing deep learning," arXiv preprint arXiv:1708.03017, 2017.
[8]. Z. Durumeric et al., "The HTTPS ecosystem: An analysis of certificate issuance, usage, and validation," in Proc. ACM Internet Measurement Conf. (IMC), Barcelona, Spain, 2013, pp. 291–304.
[9]. Z. Li, W. Xu, Y. Fu, and Z. Lin, "BRT: An efficient and scalable blockchain-based revocation transparency system for TLS connections," in Proc. IEEE INFOCOM, Paris, France, 2019, pp. 2076–2084.
[10]. L. Wang, Y. Jin, and K. G. Shin, "Encrypted mining traffic detection mechanism based on TLS handshake message and machine learning," in Proc. IEEE Int. Conf. Commun. (ICC), 2020, pp. 1–6.
[11]. R. Razavi and F. Alserhani, "Challenges and advances in analyzing TLS 1.3-encrypted traffic: A comprehensive survey," IEEE Commun. Surv. Tutor., vol. 24, no. 3, pp. 1502–1525, 2022.
[12]. Joy: A package for capturing and analyzing network flow data. [Online].
Available: https://github.com/cisco/joy
[13]. W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks," in Proc. IEEE Int. Conf. Intelligence and Security Informatics (ISI), Beijing, China, Jul. 2017, pp. 43–48.
[14]. M. Zhang, H. Zhang, B. Zhang, and G. Lu, "Encrypted traffic clas- sification based on an improved clustering algorithm," in Trustworthy Computing and Services (ISCTCS 2012), Y. Yuan, X. Wu, and Y. Lu, Eds. Berlin, Germany: Springer, 2013, vol. 320.
[15]. S. Rezaei and X. Liu, "Deep learning for encrypted traffic classification: An overview," IEEE Commun. Mag., vol. 57, no. 5, pp. 76–81, May 2019.
[16]. M. Shen et al., "Machine learning-powered encrypted network traffic analysis: A comprehensive survey," IEEE Commun. Surv. Tutor., vol. 25, no. 2, pp. 791–824, 2023.
[17]. A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," IEEE Commun. Surv. Tutor., vol. 18, no. 2, pp. 1153–1176, 2016.
[18]. G. Fernandes, J. J. Rodrigues, L. F. Carvalho, J. F. Al- Muhtadi, and M. L. Proença, "A comprehensive survey on network anomaly detec- tion," Telecommun. Syst., vol. 70, no. 3, pp. 447–489, 2019.
[19]. D. Kwon et al., "A survey of deep learning-based network anomaly detection," Cluster Comput., vol. 22, no. S1, pp. 949–961, 2019.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25jul473

[20]. A. Boukerche and J. Wang, “Machine learning-based traffic prediction models for intelligent transportation systems,” Comput. Netw., vol. 181, p. 107530, Nov. 2020.
[21]. P. Baldi, “Autoencoders, unsupervised learning, and deep architectures,” in Proc. ICML Workshop Unsupervised and Transfer Learning, Bellevue, WA, USA, Jul. 2011, pp. 37–49.
[22]. I. A. Alharbi, A. J. Almalki, M. Alyami, C. Zou, and Y. Solihin, “Profiling attack on WiFi-based IoT devices using an eavesdropping of encrypted data frames,” Adv. Sci. Technol. Eng. Syst. J., vol. 7, no. 1, pp. 49–57, 2022.
[23]. S. Khan, “Towards interoperable blockchains: A survey on the role of smart contracts in blockchain interoperability,” IEEE Access, vol. 9, pp. 116672–116691, 2021.
[24]. F. Chollet, “Xception: Deep learning with depthwise separable convo- lutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Honolulu, HI, USA, Jul. 2017, pp. 1251–1258.
[25]. S. Shen et al., “Joint differential game and double deep Q-networks for suppressing malware spread in Industrial Internet of Things,” IEEE Trans. Inf. Forensics Secur., vol. 18, pp. 5302–5315, 2023.