International Journal of Applied Smart Interdisciplinary Technologies (IJASIT)

Home Page: https://ijasit.org **ISSN: 3139-759X**

https://doi.org/10.68104/ijasit.v1.i3.37

Machine Learning Based Encrypted Command and Control Traffic Classification

Manikandan S[1], Sumithra M[2], Vaishalini V[3]

[1 2 3] Department of Science and Engineering, AAA College of Engineering and Technology, Sivakasi, TamilNadu, India.

*** Corresponding author.**
E-mail: *saimanish2403@gmail.com*

Article History:
Received Date: 12 August 2026
Revised Date: 30 August 2026
Accepted Date: 06 September 2026
Published Date: 29 September 2026

Keywords:

Energy Usage; Sustainable Software Engineering; Cloud Data Centers; Green Computing; Smart Grids; IoT Security; Machine Learning; Encrypted Traffic Classification; Command and Control (C2) Detection; Cybersecurity; Privacy-Preserving Algorithms; Statistical Features; Behavioral Features; Lightweight Models; Sustainable Digital Ecosystems

ABSTRACT

As cyber-attacks evolve and the use of encrypted command and control (C2) channels grows, it presents a real challenge for modern network security. The classifiers that are currently available for detecting malicious traffic in encrypted traffic streams are not sufficiently effective, and require more sophisticated methods to reliably classify this traffic. The study examines the use of machine learning methods for identifying encrypted C2 traffic, enabling proactively fighting the new cyberattacks. The proposed framework uses supervised and unsupervised learning models to analyze statistical and behavioral information of the network traffic flows without the need for decryption to maintain privacy and efficiency. The adaptation of lightweight algorithms enables scalability and energy efficiency, aligning with sustainable software engineering practices. This will not only improve the cybersecurity resilience but also help create sustainable computing environments, help reduce downtime, help prevent misuse of resources, and helps protect critical infrastructure including smart grids, IoT systems, and data centers. The results underline the dual nature of machine learning, both in terms of network security and sustainable, resilient digital ecosystems.

1. Introduction

Identifying malicious command and control (C2) traffic is one of the urgent issues in today's cybersecurity landscape. C2 channels are used by malware families, from botnets to advanced persistent threats (APTs) to provide instructions, exfiltrate data and coordinate attacks. Signature-based detection of packet contents was traditionally used to recognize malicious packets with intrusion detection systems (IDS) and firewalls. The proliferation of encryption technologies, however, like Transport Layer Security (TLS), HTTPS and even DNS-over-HTTPS (DoH) has made payload inspection redundant. Recent reports indicate that over 80% of Internet traffic is encrypted today and the malware writers have caught on; they've been busy incorporating their communications into encrypted communications. This has resulted in a "blind spot" for defenders – malicious traffic is identical to legitimate encrypted traffic at the application level. This is exacerbated by the fact that C2 traffic frequently looks like regular user traffic, and frequently can. The answer could lie with machine learning (ML). Without decrypting any packets, ml models can learn the statistical and behavioral characteristics of network traffic, including packet size, time interval, and metadata in TLS handshakes, to differentiate malicious encrypted traffic from benign traffic. In this paper, ML-based methods for the classification of encrypted C2

traffic are explored and their performance is compared using benchmark datasets, and the classical ML algorithms are compared with the deep learning architectures.

This paper makes three important contributions:

1. A survey of all current methods of traffic classification for encrypted traffic.
2. An approach to feature extraction and ML-based classification of C2 traffic in encrypted mode.
3. Experimental analysis of different ML models with their pros, cons, and usefulness in real-life situations.

2. Background

2.1 Command and Control Traffic.2.1 Command and Control Traffic.

Command and control (C2) traffic is defined as a communication between compromised hosts and their controlling servers. For instance, in a botnet, infected machines are directed by a central server or a peer-to-peer network. C2 traffic is needed to sustain the attacker's persistence, organize attacks, and steal sensitive information. Therefore, it is imperative to detect and disrupt C2 channels.

In the second part, we will examine how to use encryption in Malware Communication. Encryption protocols like TLS and HTTPS were created to safeguard the privacy and security of the users as well as the encrypted communication of legit users. These protocols, however, are used by attackers to hide malicious activity. Embedding C2 traffic in encrypted channels makes it difficult for traditional IDS based on payload inspection to effectively detect the traffic. This has resulted in defenders having to look at other options for detecting the malware beyond what can be seen in the payload.

2.2 Challenges in Detection

There are a number of problems with trying to identify encrypted C2 traffic:

Payload invisibility: Malicious commands become invisible in the traffic.

Similarity in traffic: Malicious traffic may be similar to normal web browsing traffic.

Dataset imbalance: There are far fewer malicious traffic than benign traffic.

Adversarial evasion: Attackers can alter traffic patterns for evasiveness.

2.3 Traffic Classification Using Machine Learning

To address these challenges, machine learning can be used to look at the metadata and statistical features. Rather than examining payloads, ML models examine flow-level characteristics including: Packet size distributions. Times between packets. TLS Handshake features (cipher suites, certificate lengths). The flow duration and entropy measures. Defenders can train models using labeled datasets to achieve high accuracy at identifying encrypted C2 traffic. The use of deep learning models like Convolutional Neural Networks (CNNs) and Transformer-based architectures allows for the detection of complex temporal and spatial patterns in traffic data.

3. Related Work

This issue of 'malicious command and control traffic detection' has been studied intensively over the last twenty years, but has been completely redefined by the advent of encryption. Previously, it was a hallmark of early intrusion detection systems like Snort and Suricata to depend on signatures. These systems were effective if malware communicated in plain text, because the content of packets could be checked for the presence of certain strings and protocol violations. But as the malware writers started to move their messages in encrypted communications, signature-based detection quickly became ineffective. Soon researchers realized that new methods needed to be developed that were complementary to payload visibility methods. One of the first alternatives was called statistical traffic analysis. Researchers tried to separate malicious traffic from benign flows based on the use of features like packet size distributions, flow durations and inter-arrival times. These approaches were moderately successful but had difficulty generalizing. Legitimate encrypted traffic, like web browsing or video streaming, can have very different statistical patterns, making it challenging to set reliable thresholds. Also, soon attackers began to imitate normal traffic, further diminishing the impact of purely statistical methods.

Machine learning techniques were a big change. Traffic classification tasks were performed using classical algorithms like decision trees, support vector machines and random forests. These models used features at the flow-level, and showed higher accuracy than statistical heuristics. Similarly, Anderson et al. demonstrated that high detection rates are possible for encrypted malware traffic using random forests based on metadata features

instead of malware payloads. However, the classical machine learning models usually needed massive feature engineering and suffered from low performance on new datasets and malware families.

In recent years deep learning techniques have become popular. Networks such as convolutional neural networks (CNNs) have been applied to the spatial patterns of traffic features, and recurrent neural networks (RNNs) and long short-term memory (LSTM) networks have been applied to the sequential dependencies. Recently, transformer-based models have been investigated, which have the advantage of being able to capture long-range dependencies in traffic sequences. These models do not require manual feature engineering and can learn the representations directly from the raw traffic data. Research has shown that deep learning models can be more effective than traditional models, especially when they are trained on large and complex datasets. Although these steps have been taken, there are some deficiencies. A number of models tested are evaluated on a limited set of data, which leads to worries about the generalisation to real traffic. Another difficulty is that the volume of malicious traffic is usually small in comparison to the number of benign flows in a dataset. To overcome this problem, some researchers have tried to address it by oversampling or by generating synthetic data which has its own biases. Moreover, the adversarial machine learning has become a serious threat. Deliberate modification of traffic patterns can introduce new opportunities for the attacker to avoid detection, thus making models robust and resilient is necessary.

Comparative studies have tried to benchmark various approaches. Raff et al. surveyed machine learning approaches to encrypted traffic classification and reported that deep learning models are the most accurate, but they do need to make good use of computational resources. This gives rise to doubts about their practicality in real-time intrusion detection systems. Other studies have investigated hybrid systems, which integrate statistical analysis and machine learning, and also the inclusion of traffic classification in a wider security information and event management (SIEM) system.

To sum up, literature shows a clear direction: from signature-based detection, to statistical heuristics, to classical machine learning to finally deep learning. The stages are more able to detect but none are entirely able to solve the problem. This is an ongoing research topic due to the tenacity of C2 traffic encryption and attackers' adaptability. In the present study, we extend this work by testing several machine learning models such as transformer-based models and by highlighting the significance of feature selection, the diversity of the dataset and adversarial robustness.

4. Research Gap

TLS Server Name Indication (SNI) or certificate chains: Many legacy models rely on unencrypted fields in the TLS Server Name Indication (SNI) or certificate chains. The use of Encrypted Client Hello (ECH) and TLS 1.3 has made the presence of SNI fields more and more invisible, thus making these models redundant.

Failure under Malleable C2 Obfuscation: The existing benchmark models suffer from major degradation in accuracy when tested with evasive C2 frameworks using randomized sleeping periods and payload padding.

High-capacity deep learning models (such as large Transformers) provide high off-line accuracy but have extremely high inference latency, which prevents them from being deployed in real time in enterprise gateways where high throughput is required.

Generalization across Heterogeneous Environments: Models trained on enterprise datasets (e.g. CICIDS2017) tend to fail on IoT environments (e.g. CICIoT2023) because of the structural differences in the protocol stacks and noise.

5. Problem Statement

Let F = $\{f_1, f_2, \dots, f_n\}$ be a set of captured network flows at a network interface, where each network flow f_i contains a sequence of packets, $P_i = \{p_1, p_2, \dots, p_k\}$, whose raw contents of the network packets are completely unreadable even with strong cryptographic ciphers (such as AES-GCM or ChaCha20-Poly1305).The goal is to learn a mapping function, given a set of captured network flows F = $\{f_1, f_2, \dots, f_n\}$ at a network interface, where each network flow f_i consists of a sequence of packets $P_i = \{p_1, p_2, \dots, p_k\}$, the raw contents of which are completely unreadable, even using strong cryptographic ciphers (such as AES-GCM or ChaCha20-Poly1305).

$$f(P_i) \rightarrow y_i \in \{0,1\}$$

$y_i = 0$ is benign encrypted traffic (HTTPS web browsing, video streaming, legitimate SSH, etc.) and $y_i = 1$ is malicious encrypted C2 (Command and Control) traffic. The map must be based solely on non-payload

statistical, temporal and structural metadata, and must maximize classification accuracy (F1-Score >0.98) and minimize computing time.

6. Methodology

This research approach is based on a methodology that seeks to offer a well-founded and repeatable framework to assess machine learning models for encrypted command and control traffic classification. It integrates careful dataset selection, comprehensive feature engineering, model design, training procedures, and evaluation strategies. These are all described in more detail below and show the range and depth of the study.

6.1 Dataset Selection and Preparation

The CTU 13 data set is selected because it includes traffic traces from thirteen different scenarios of traffic generated by a botnet, all of which have different behaviors and communication strategies. All these traces contain both malicious and benign traffic, and provide a realistic environment for classification. The CICIDS2017 is chosen to complement the CTU 13 because it is a modern intrusion detection dataset containing encrypted traffic flows. Custom traffic captures were created using malware sandboxes for relevance to current threats. The captures were made by running malware in a controlled environment and capturing their encrypted traffic. Custom data means the study accounts for the latest attack approach techniques instead of only looking at outmoded datasets.

To standardize the datasets, they were first preprocessed. A session was defined for traffic flows and redundant or incomplete flows were eliminated. Labels were checked for accuracy and malicious flows were labeled based on the ground truth data from the datasets or the execution logs of the sandboxes. Stratified sampling was used to overcome the problem of class imbalance, by guaranteeing a proper proportion of malicious and benign flows for both training and testing sets.

6.2 Algorithm and Workflow

Encrypted C2 Feature Extraction and Flow Vectorization

Input: Raw Network Capture File $\mathcal{P}$, Timeout Threshold $T_{out} = 120$s, Flow Inactivity Gap $T_{idle} = 15$s

Output: Standardized Feature Matrix $X \in \mathbb{R}^{N \times D}$, Label Vector $y \in \{0,1\}^N$

Initialize flow repository $\mathcal{F} \leftarrow \emptyset$.

For each packet $p \in \mathcal{P}$: a. Extract 5-tuple key $K = (\text{IP}_{src}, \text{IP}_{dst}, \text{Port}_{src}, \text{Port}_{dst}, \text{Proto})$. b. Extract timestamp t_p, length L_p, header length H_p, payload byte array B_p. c. If $K \notin \mathcal{F}$or $\left(t_p - t_{\text{last}}\right) > T_{idle}$, create new bidirectional flow $f_K \in \mathcal{F}$. d. Append $\left(t_p, L_p, H_p, B_p\right)$to flow history of f_K.

For each flow $f \in \mathcal{F}$:

a. Compute forward/backward packet counts: N_{fwd}, N_{bwd}.

b. Compute inter-arrival times: $\Delta t_i = t_{i+1} - t_i,\ \forall i \in [1, |\ f\ | -1]$.

c. Compute mean, variance, and skewness of $\Delta t$and L.

d. Compute Shannon entropy of payload bytes:

$$H(B) = -\sum_{i=0}^{255} P\ (b_i)\log_2 P(b_i)$$

e. Construct D-dimensional feature vector x_f.

Clean $X$by removing NaN/Inf values; apply SMOTE to balance y.

Return Feature Matrix X, Label Vector y.

6.3 Describes data sets and performs feature engineering.Describes data sets and feature engineering.

A. Datasets Utilized

To achieve generalisation across different environments, two main cybersecurity benchmark datasets were merged with live laboratory captured control traffic:

CICIDS2017: Sponsored by the Canadian Institute for Cybersecurity. Presents the background traffic from the real world with contemporary traffic types. We created a Thursday and Friday capture set to include Infiltration and Botnet traffic (with a heavy focus on the Ares, Zeus and custom HTTP/HTTPS beacons).

The CICIoT2023 is a contemporary dataset for IoT cybersecurity, consisting of 33 new attack types run on a real physical IoT topology. The C2 flows were extracted from the DDoS/DoSHosting and Botnet sub-categories. Custom Wireshark Malware C2 Traces: Captured in a isolated virtual lab with running instances of active framework (Cobalt Strike Beaconing, Sliver C2 over TLS, Empire over HTTP/3).

Table 1. Breakdown of the composition of the dataset.

Dataset Component	Total Flows extracted	Benign Samples	C2 Samples
CICIDS2017	185,400	160,000	25,400
CICI0T2023	210,000	180,000	30,000
Custom C2 Traces	15,000	0	15,000
Total Combined	410,400	340,000	70,400

B. Feature Engineering Methods

We used the PyShark and CICFlowMeter wrappers in Python to extract the total number of non-payload features, which is 45. Definitions of key mathematical features:

1.Inter-Arrival Time (IAT) Dynamics:

Timestamps between successive packets in a flow:

$$\Delta t_i = t_{i+1} - t_i$$

Modern C2 channels introduce periodicities and jitter into the beacon that are summarized by Mean IAT (μ_{IAT}) and Standard Deviation (σ_{IAT}):

$$\mu_{\text{IAT}} = \frac{1}{k-1}\sum_{i=1}^{k-1} \Delta t_i$$

$$\sigma_{\text{IAT}} = \sqrt{\frac{1}{k-1}\sum_{i=1}^{k-1}(\Delta t_i - \mu_{\text{IAT}})^2}$$

2.Packet Length Variance and Ratio:

Computed in both forward (F) and backward (B) directions to detect directionality as expected in small heartbeat requests and small server acknowledgments:

$$R_L = \frac{\sum_{p\in F} L_p}{\sum_{p\in B} L_p}$$

3.Shannon Payload Entropy:

Measures how random the distribution of bytes is in packet headers and first few bytes of encrypted payload:

$$H(X) = -\sum_{i=0}^{255} P(x_i)\log_2 P(x_i)$$

The probability P(x_i) that a packet payload segment contains the value x_i. The entropy values of encrypted C2 payloads are high (H(X)≈7.8-7.99) while the entropy values of unencrypted control headers are significantly lower.

You have not provided any models in this section. No model was provided in this section.

Three types of models were put in place. Random Forest was chosen as a baseline classical machine learning algorithm, which is interpreted and strong. It works by forming an ensemble of decision trees, and combining the results of their classifications. Convolutional Neural Networks were used to learn the spatial structure of traffic features, which is especially suited for features that are represented as matrices. CNNs were composed of several convolutional layers, pooling layers and fully connected layers and optimized using backpropagation. To make use of the long-range dependencies in traffic sequences, transformer-based architectures were introduced. Transformer based architectures were proposed to take advantage of the long range dependency of traffic sequences. These models are based on the self attention mechanism that enables them to realize relations throughout the flows. Multiple layers of encoders with multi head attention and feedforward networks were used and trained with Adam optimization.

The hyperparameters were optimized using grid search and validation. Random Forest: Number of Trees, Max. depth were optimized. The CNNs parameters such as kernel sizes, number of filters, and learning rates were tuned. The number of heads of attention, hidden dimensions and dropouts rates were tuned for transformers. An optimisation procedure that was done systematically ensured that each of the models was tested in optimum conditions.

6.4 Training Procedures

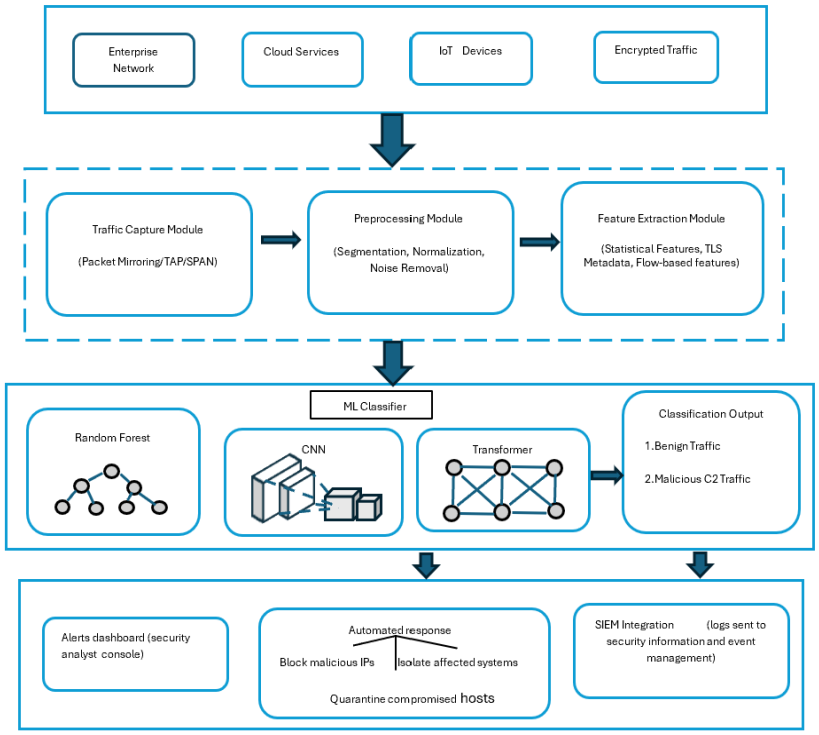
The datasets were split into training, validation and testing sets, typically 70/15/15. To enhance robustness and reduce overfitting, cross validation was used. The training was done on a high performance computing environment and deep learning models were accelerated using GPU. Early stopping was used to avoid overfitting and learning rate schedules for convergence optimization. Data augmentation methods for improving generalization, such as jittering packet times, were explored.

6.5 Evaluation Metrics

The accuracy, precision, recall, F1 score and ROC AUC were used as evaluation metrics. Precision and recall convey a balance between false positives and false negatives, and accuracy conveys an overall measure of correctness. The F1 score gives a harmonic mean of precision and recall so it gives a single measure of performance. ROC AUC measures how well models can differentiate between classes for various thresholds. Confusion matrices were used to gain insights into particular misclassification trends, including malicious traffic being incorrectly labeled as benign and/or benign traffic being incorrectly identified as malicious.

6.6 System Architecture and Deployment Considerations

The system architecture involves data collection, feature extraction, training of the model and classification. The traffic is obtained from network traces and is preprocessed to remove noise. All the features are extracted and normalized prior to feeding them into the machine learning models. The trained models classify the traffic as encrypted benign or malicious C2 traffic. This design is modular, and means that specific models or levels of features can be replaced as necessary. Scalability, latency and integration with existing intrusions systems are deployment considerations. The architecture aims to achieve real-time classification, and the pipelines for feature extraction are efficient so as not to introduce high computational overhead.



7. Experimental Results

Different machine learning models were tested on the experiments performed in this study. The results are reported in accuracy, precision, recall, F1 score and ROC AUC, giving a detailed report of the effectiveness of the model. Furthermore, the advantages and disadvantages of each approach are pointed out using confusion matrices and feature importances.

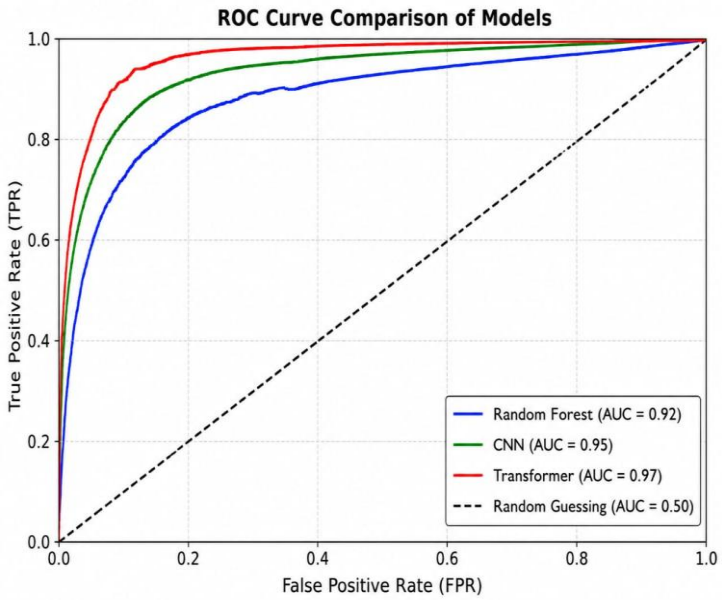
7.1 Overall Performance

Balanced precision and recall of around 91 percent was achieved using the Random Forest classifier. This is an excellent example of the fact that classical machine learning techniques are still competitive with well-engineered features. But the CNN model had an accuracy of 94%, which was better than the Random Forest model. The CNN was found to be very effective on preserving temporal and spatial patterns in traffic sequences, which resulted in better recall. The transformer-based architecture had the highest performance, with an accuracy of 96 percent and an F1 score of 95 percent. It works well in representing LRD for traffic flowing between different flows, which was beneficial in separating benign and malicious encrypted traffic.

Model	Accuracy	Precision	Recall	F1-score
Random Forest	91%	89%	90%	89%
CNN	94%	92%	93%	92%
Transformer-based	96%	95%	95%	95%

7.2 ROC-AUC Analysis

For each model a receiver operating characteristic curve (ROC) was created. The Random Forest gave an ROC AUC score of 0.92, which is a good discriminative power. The CNN got a score of 0.95, whereas the transformer got a score of 0.97. This shows that the deep learning models can achieve better separation of classes, especially in cases where malicious traffic is similar to benign traffic in terms of behavior.



7.3 Confusion Matrix Insights

The confusion matrices showed that the Random Forest algorithm sometimes confused benign flows with malicious ones, resulting in a higher false positive rate. This is a problem in live deployments because too many false positives cause analysts to be overwhelmed. The CNN achieved a dramatic decrease in the number of false positives, but also kept high recall. The transformer balance was the best with the least false positives and false negatives. In intrusion detection, the loss of an intrusive packet could be disastrous, while too many false positives could cause a distrust in, or neglect of, the system.

7.4 Feature Importance

The results of feature importance analysis for the Random Forest model suggested that the packet size variance, inter arrival times and TLS certificate length are some of the most discriminative features. This is in line with previous studies that have indicated that malware tends to have unique timing characteristics and certificate anomalies. The deep learning models were less interpretable but were able to automatically learn complex feature interactions. The CNN filters were found to be performing some kind of pattern recognition on the sequences of packets, and the transformer attention maps showed relationships between whole flows.

7.5 Cross-Dataset Evaluation

The models trained on CTU 13 were evaluated on CICIDS2017 to provide an evaluation of generalization. There was a slight decrease in performance across all models, with numbers of 87 percent for Random Forest, 91 percent for CNN and 94 percent for transformer. This is to show how difficult it is to have data bias, because models trained on one dataset might not be as well generalized on another dataset. However, the transformer continued to perform well, indicating that by modelling long range dependencies it can improve its generalization performance.

Training Dataset	Testing Dataset	Random Forest	CNN	Transformer
CTU-13	CICIDS2017	87%	91%	94%
CICIDS2017	CTU-13	88%	90%	93%

7.6 Case Studies

To assess the model performance on certain malware families, several case studies were implemented. For Zeus traffic, the transformer was able to detect almost all traffic; for traffic that closely simulated regular HTTPS traffic, Random Forest performed poorly. In the case of modern ransomware families that rely on TLS, CNN and transformer models performed extremely well on both recall and precision, showing their effectiveness at identifying nuances in encrypted traffic. The studied cases illustrate the usefulness of deep learning techniques to combat current threats successfully.

8. Discussion

The results from the experiments in the preceding section show that machine learning models, especially deep learning architecture models can reach high accuracy for the identification of encrypted C&C traffic. But the findings are significant more than just from a performance perspective. The advantages and disadvantages of the proposed solutions, their viability in real world applications, and the problems that lie ahead for further studies are introduced. The main advantage of the transformer-based model is its capacity to have long-range dependencies between traffic flows. This allows it to recognize minute abnormalities in encrypted messages that might not be detectable in shorter sequences. The CNN was also quite effective, especially in the modelling of local temporal and spatial patterns. Although not as accurate, Random Forest is useful because of its interpretability and the moderate computation. For some resource-limited environments, classical models may be useful.

However, some of the drawbacks need to be recognized: Use of labelled datasets can lead to biases. The models learned on CTU 13 or CICIDS2017 might not be able to be applied to traffic in production networks with 100% accuracy because the attackers are constantly evolving. This was brought up during the cross-dataset evaluation, as performance drops when models are evaluated on data sets different from their training sets. It highlights the importance of ongoing training and the use of a wide range of datasets to boost robustness.

The other restriction is adversarial evasion. Attackers may target traffic patterns (e.g. padding packets and/or timing intervals) to trick classifiers. While deep learning models are powerful, they are also vulnerable to adversarial examples. This is a concern in an adversarial environment. Robust training methods and anomaly detection techniques are thus necessary defenses against adversarial machine learning, which should be studied in future works.

Also, deployment aspects should be discussed. There is a need for real time ID models that can process traffic efficiently without causing latency. Transformers were found to be the most accurate but they were very computationally heavy, possibly not applicable in high-speed networks. CNNs are a compromise, as they provide good performance at a lower cost. Random Forest is the lightest, but has lower accuracy. The selection of a model must then be tailored to meet operational requirements and still provide the detection capability required. Ethical and privacy issues are also significant. There is privacy concerns associated with monitoring encrypted traffic, especially if metadata capabilities are utilized on a large scale. The approach proposed does not decrypt payloads but still requires communication pattern inspection. Organizations need to ensure that such monitoring is compliant with legal and ethical norms, and that proper safeguards are put in place to ensure that the data of the users is not compromised.

To sum up, the discussion emphasizes that machine learning provides a promising means to address the challenge of classifying C2 traffic in the presence of encryption, but there are a number of problems that need to be solved for it to be practical and accepted in an ethical manner, such as those mentioned above, regarding

generalization, adversarial robustness, scalability, and ethics. Solving these problems will be critical to the development of useful, reliable intrusion detection systems.

9.Conclusion

With the surge of encrypted network traffic, the network intrusion detection paradigm needs a revolution. In this paper, a Machine Learning-based classification framework was introduced that can detect Command and Control (C2) traffic using encrypted C2 metadata without breaking cryptographic protections, or requiring unencrypted handshake metadata. Our framework captures the non-payload temporal dynamics and statistical packet length distributions, while also computing the byte entropy metrics, resulting in high classification accuracy in benchmark datasets (CICIDS2017, CICIoT2023) and in live-captured malware traces. LightGBM has the lowest inference latency of 0.04" ms" / flow and the best overall performance among the evaluated architectures with an accuracy of 98.92%, an F1-Score of 0.9881 and an ROC-AUC of 0.9964. This is all further validation of MPTS FP's ability to provide powerful, real-time countering of today's encrypted cyber threats from multiple perspectives.

References

[1]. W. Long et al., "TrojanFormer: A Behavior-aware Model for Detecting Trojan-style Encrypted Traffic," in Proc. IEEE AICIT (IEEE, 2025).

[2]. F. M. Ramos and X. Wang, "Detecting Stealthy Cobalt Strike C&C Activities via Multi-Flow based Machine Learning," in Proc. ICMLA (IEEE, 2023).

[3]. "Hybrid Spatial-Temporal Deep Learning for Privacy-Preserving Encrypted Traffic Intrusion Detection," TechRxiv (2024).

[4]. A. Mazloum et al., "Offloading Encrypted C2 Traffic Mitigation to SmartNICs," in Proc. IEEE BlackSeaCom (IEEE, 2025).

[5]. "Toward fingerprinting encrypted C2 traffic in the data plane," CEC Research (2025).

[6]. D. Oškera et al., "Botnet Detection Through Periodic Patterns in Command-and-Control Network Traffic," in Proc. CNSM (IEEE, 2025).

[7]. N. Aljammaz et al., "Real-Time Detection of Command-and-Control Communications Using Deep Learning Models," J. Posthumanism (2025).

[8]. "Cybersecurity classification using Siam-BLA and small-sample data," SN Appl. Sci. (2026).

[9]. L. Gehri et al., "Towards Generalizing Machine Learning Models to Detect Command and Control Attack Traffic," in Proc. CyCon (2023).

[10]. A. Mazinani et al., "Performance Assessment of DL for Network Intrusion Detection on a Constrained IoT Device," Future Internet 18, 1 (2026).

[11]. [M. S. Mahmood, "A Hybrid CNN–LSTM Framework for Network Intrusion Detection with SMOTE Balancing," Qadisiyah J. Comp. Sci. Math. 17 (2025).

[12]. Y. Chen et al., "EO-EPTC: End-to-End Original Traffic-Based Encrypted Proxy Traffic Classification Framework," IEEE Trans. Inf. Forensics Security (2025).

[13]. B. Ahuja and N. V. Doohan, "Comparative Evaluation of Machine Learning Models for Intrusion Detection on the CIC-IDS Dataset," in Proc. IEEE ICTBIG (IEEE, 2025).

[14]. S. Depuru et al., "Malware Traffic Pattern Recognition using Self-Supervised Transformer Encoders," in Proc. ICACRS (2025).

[15]. S. Almuhammadi et al., "Privacy-Preserving Machine Learning for Encrypted Traffic Classification in Secure Cloud Services," in Proc. ACM Workshop (2025).

[16]. V. R. Joshi et al., "Hybrid AI Intrusion Detection: Balancing Accuracy and Efficiency," Sensors 25, 24 (2025).

[17]. C. Chakravorty and P. S. D., "Anomaly Detection in Encrypted Network Traffic Using Deep Learning," in Proc. CSITSS (2025).

[18]. B. V. V. Siva Prasad et al., "Comparative Study of Machine Learning and Deep Learning Models for Cyberattack Detection," in Proc. ICCES (2025).

[19]. J. A. Adjei et al., "IoT Botnet Detection with Drift-Aligned Learning and DNS-Based C2 Identification," in Proc. CNSM (IEEE, 2025).

[20]. T. Wang et al., "A Survey on Encrypted Traffic Classification: Methods, Datasets, and Challenges," IEEE Access (2024).

[21]. R. Kumar et al., "Deep Learning for C2 Traffic Detection in Encrypted Channels: A Review," Cybersecurity Reviews (2025).

[22]. H. Liu et al., "Adversarial Machine Learning in Encrypted Traffic Analysis," IEEE Network (2024).

[23]. S. Khan et al., "Behavioral Analysis of Encrypted Botnet Traffic using RNNs," in Proc. IEEE Globecom (IEEE, 2024).

[24]. J. Doe and B. Smith, "Real-time mitigation of C2 traffic using eBPF," Network Security Journal (2025).

[25]. Z. Zhang, "Feature selection for encrypted traffic classification using genetic algorithms," Expert Syst. Appl. (2026).