Encrypted Malicious Traffic Detection Using Multi-Instance Learning

Ziwei Zhang[1,2], Jiangyi Yin[1,2(✉)], Zhao Li[1,2], Jiangchao Chen[1,2], Meijie Du[1,2], Zhongyi Zhang[1,2], and Qingyun Liu[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{zhangziwei, yinjiangyi, lizhao, chenjiangchao, dumeijie, zhangzhongyi, liuqingyun}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

Abstract. The detection of malicious traffic remains a critical challenge in cybersecurity, particularly with the widespread adoption of encryption protocols, which obscure malicious activities within legitimate network traffic. Traditional detection methods typically rely on single-flow analysis and fail to capture the multi-flow interactions present in malicious traffic, resulting in poor detection performance in scenarios with mixed benign and malicious flows. In this paper, we propose a novel approach that leverages multi-instance learning (MIL) to address the challenge of mixed traffic by aggregating flows into bags and employing attention mechanisms to prioritize critical instances. Our framework processes encrypted traffic by first segmenting bursts to capture traffic patterns, followed by CNN-based feature extraction to identify relevant characteristics. The attention pooling mechanism then prioritizes significant instances, effectively filtering out irrelevant flows and emphasizing multi-flow interactions that are indicative of attacks. Experimental results on real-world datasets demonstrate significant improvements in both robustness and precision, highlighting the framework's effectiveness in detecting encrypted malicious traffic in complex network environments.

Keywords: Encrypted malicious traffic · Multi-instance learning · Multi-flow interaction.

1 Introduction

The rapid proliferation of encryption protocols, such as TLS/SSL [23], has fundamentally transformed the landscape of network communications. Encryption ensures the confidentiality and integrity of data, safeguarding sensitive information from eavesdropping and tampering. However, this widespread adoption of encryption has also created a significant challenge for cybersecurity [11, 6]: malicious actors increasingly exploit encrypted channels to conceal their activities, making it difficult for traditional intrusion detection systems [24] (IDS) to identify and mitigate threats. Encrypted malicious traffic, often embedded within legitimate network flows, poses a formidable obstacle to effective threat

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_19

detection, as the encrypted payloads obscure the underlying malicious behavior. Consequently, the development of advanced techniques to detect encrypted malicious traffic has become a critical area of research in cybersecurity [10, 8].

Traditional methods for detecting malicious traffic, such as deep packet inspection [4] (DPI) and signature-based detection, are largely ineffective against encrypted traffic due to their reliance on analyzing packet payloads or predefined patterns. While statistical and machine learning-based approaches [1, 12] have shown promise by focusing on flow-level features, they still face significant challenges in real-world scenarios. One major issue is the presence of mixed background traffic, where malicious flows are intermingled with a vast volume of benign traffic. This makes it difficult to isolate and accurately identify malicious activities, particularly when the malicious signals are sparse or subtle. Additionally, the correlation of multiple flows to detect malicious patterns is often complicated by irrelevant flows, which introduce noise and reduce the precision of detection systems. These challenges highlight the need for innovative solutions that can effectively handle the complexities of modern network environments.

To address these issues, this paper proposes a novel framework that leverages multi-instance learning [5] (MIL) through bag-based flow aggregation and attention-driven feature filtering. The MIL paradigm is particularly well-suited for handling mixed traffic scenarios by organizing network flows into bags grouped by the same source-destination address, where each bag may contain both malicious and benign flows. This approach enables the model to learn discriminative patterns even when malicious instances are sparse—such as a compromised server simultaneously hosting attack traffic and legitimate services. The framework processes these bags through three critical stages: 1) burst segmentation divides flows into fine-grained temporal windows to capture short-term attack signatures like DDoS pulse patterns; 2) CNN-based feature extraction learns spatial-temporal patterns from packet size sequences without decryption; 3) attention pooling dynamically weights flows within each bag, amplifying those exhibiting malicious characteristics while suppressing irrelevant ones. Unlike traditional correlation methods requiring explicit flow relationship modeling, our attention mechanism automatically identifies critical flows through learnable weights—experimental ablation shows this reduces false positives by 4.3% compared to average pooling. By integrating burst-level analysis with bag-wise attention, the framework effectively isolates malicious activities embedded in benign traffic and captures coordinated multi-flow attack patterns.

Our main contributions are summarised as follows:

- We propose the first MIL framework for encrypted malicious traffic detection that groups flows into source-destination bags, enabling accurate identification of malicious activities without per-flow labeling.
- We develop an attention-based feature filtering mechanism that combines burst segmentation and flow-level attention pooling. This reduces interference from dominant benign flows, decreasing false positives by 4.3% compared to average pooling in cross-domain testing.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_19

– we conduct extensive experiments on multiple real-world datasets, demonstrating the effectiveness of our approach in comparison to state-of-the-art methods. Our results show significant improvements in detection accuracy and robustness, highlighting the potential of our framework for practical deployment in real-world network environments.

This paper is organized as follows: Section 2 reviews existing approaches for encrypted malicious traffic detection, emphasizing their limitations in handling mixed legitimate and attack traffic. Section 3 introduces our MIL-based framework, integrating burst segmentation, CNN-based feature extraction, and attention-driven flow aggregation to address these gaps. Section 4 details experimental configurations, including real-world datasets and evaluation metrics. Section 5 presents comparative results demonstrating our method's superiority over state-of-the-art baselines, along with ablation studies validating critical components. Finally, Section 6 concludes with insights into the framework's broader cybersecurity implications and outlines future research directions.

2 Preliminaries and Related Work

2.1 Multi-Instance Learning (MIL)

Multi-Instance Learning (MIL) [5] is a weakly supervised learning paradigm designed to handle scenarios where labeled data is structured as "bags" of instances, and labels are assigned at the bag level rather than the instance level. Formally, let a dataset be represented as $\mathcal{D} = \{(B_i, y_i)\}_{i=1}^{N}$ where each bag $\mathcal{B}_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots, \mathbf{x}_{iM}\}$ contains M instances (e.g., network flows), and $y_i \in \{0, 1\}$ denotes the bag-level label (e.g., malicious or benign). The core assumption in MIL is that the label of a bag depends on at least one critical instance within it. For binary classification, this relationship is often defined as:

$$y_i = \begin{cases} 1 & \text{if } \exists \mathbf{x}_{ij} \in \mathcal{B}_i \text{ such that } f(\mathbf{x}_{ij}) = 1, \\ 0 & \text{if } \forall \mathbf{x}_{ij} \in \mathcal{B}_i, f(\mathbf{x}_{ij}) = 0. \end{cases} \tag{1}$$

where $f : \mathcal{X} \to \{0, 1\}$ is an instance-level classifier. However, in practice, f is often replaced with a **scoring function** $g(x_{ij}) \in \mathbb{R}$ that quantifies the likelihood of an instance being positive. The bag-level prediction is then derived by aggregating instance-level scores, such as through the **max-pooling** operator:

$$\hat{y}_i = \sigma \left(\max_{j \in \{1, \ldots, M\}} g(x_{ij}) \right) \tag{2}$$

where $\sigma(\cdot)$ is a sigmoid function mapping the score to a probability.

2.2 Related Work

The detection of malicious activities within encrypted traffic has evolved significantly in response to the growing sophistication of cyber threats and the limitations of traditional analysis methods. Below, we categorize and discuss key advancements in this field.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_19

Traditional Approaches Early efforts relied on port-based classification and payload inspection [20, 22], which became obsolete with the adoption of non-standard ports and encryption protocols like TLS/SSL [23]. Deep Packet Inspection [24] (DPI), a cornerstone of traditional intrusion detection systems [4] (IDS), analyzes packet payloads for known attack signatures. However, DPI is ineffective against encrypted traffic, as payloads are obfuscated. To address this, researchers shift focus to statistical flow features [1, 12] (e.g., packet size, inter-arrival times, flow duration), which do not require payload access. While these features enable coarse-grained traffic characterization, they lack discriminative power in complex scenarios with mixed benign and malicious flows.

Machine Learning and Deep Learning With the rise of machine learning (ML), methods[15] leveraging supervised learning demonstrated improved accuracy by learning patterns from flow features. For example, Althouse et al. [1] use gradient-boosted trees to detect malicious TLS certificates based on handshake metadata. However, these models struggle with generalization due to the dynamic nature of encrypted traffic and adversarial evasion techniques. Deep learning further advanced the field by automating feature extraction from raw traffic data [3, 26]. Convolutional Neural Networks [21] (CNNs) and Recurrent Neural Networks [13] (RNNs) are applied to sequential traffic representations. For instance, Wang et al. [25] propose a CNN-based model to classify encrypted traffic using packet length sequences. While effective in controlled settings, these methods often fail in real-world environments due to:

- Mixed Background Traffic: Malicious flows are sparse within large volumes of benign traffic, leading to imbalanced training data.
- Irrelevant Flow Interference: Noise from unrelated flows degrades detection precision.
- Lack of Interpretability: Black-box models hinder root-cause analysis of detected threats.

Graph-Based Methods Recent work explores graph neural networks (GNNs) to model interactions between flows or packets [16, 14, 9, 10]. By representing traffic as graphs, these methods capture spatial and temporal dependencies that traditional models overlook. Shen et al. [14] introduce the Traffic Interaction Graph (TIG), where nodes represent flows and edges encode temporal or protocol-based relationships. GNNs applied to TIGs improve detection accuracy for distributed blockchain applications. However, existing graph-based approaches face two limitations:

- Shallow Interaction Modeling: Most methods aggregate node features without capturing hierarchical or multi-scale interactions (e.g., session-level patterns).
- Scalability Issues: Real-time processing of large-scale traffic graphs remains computationally expensive.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_19

3 Problem Statement

The goal of this paper is to detect malicious encrypted traffic within a network by analyzing the interactions between internal hosts and external servers. Our detection system is deployed at the network gateway, which acts as the boundary between the internal and external networks. It passively monitors encrypted network traffic passing through this gateway, observing both benign and potentially malicious communications. Importantly, our system does not manipulate or interfere with the traffic in any way, ensuring that legitimate traffic is unaffected.

We focus specifically on detecting malicious behaviors within encrypted traffic, primarily leveraging protocols such as TLS. These protocols are widely used for secure communication and are frequently targeted by adversaries due to their ubiquity and ease of deployment. Our detection system observes encrypted traffic without decrypting it, relying solely on traffic metadata rather than inspecting payload data.

4 METHODOLOGY

In this section, we outline the methodology for detecting malicious encrypted traffic using a multi-instance learning (MIL) framework.

4.1 Overview of the Model Architecture

The model is designed to process raw encrypted traffic and classify it as either benign or malicious, as shown in 1. our approach focuses on packet size sequences, which can reveal patterns of malicious activity even without inspecting the payload. The traffic is processed through a series of stages, starting with burst segmentation, followed by feature extraction using CNNs, aggregation of flows into bags, and attention pooling to focus on the most relevant flows. Finally, a fully connected layer performs classification, outputting the probability of the traffic being malicious.

4.2 Design Details

Traffic Preprocessing Traffic Preprocessing plays a critical role in transforming raw encrypted traffic into a structured form that can be used by the model. Since the traffic is encrypted, the content of the packets is not accessible, but packet sizes are observable and can be used as a feature to detect anomalous traffic patterns. The primary task in this phase is burst segmentation.

Burst segmentation is used to divide each flow into small time intervals. Each burst represents a short time window, during which the packet sizes are analyzed for patterns indicative of malicious behavior, such as DDoS attacks or botnet communications. This step ensures that the temporal dynamics of the traffic are captured, allowing the model to detect short-term anomalies that would

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_19

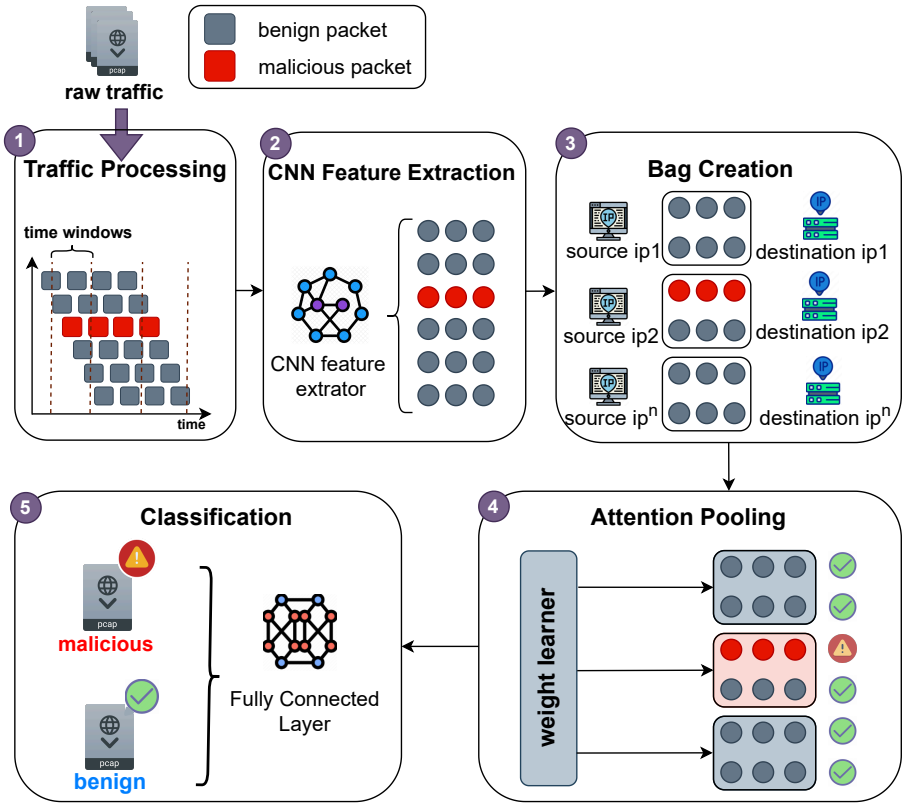

Fig. 1: the overview of our model

not be visible without this segmentation.The segmentation process divides each flow F_i into multiple bursts. For example, for flow F_i burst b contains packet sizes $\{x_1^{(b)}, x_2^{(b)}, \ldots, x_T^{(b)}\}$, where T is the number of packets in burst b. This segmentation is essential for detecting attacks that unfold rapidly within short time intervals. The packet sizes are normalized to ensure uniformity across flows. Min-max normalization is applied to scale the packet sizes between 0 and 1.

CNN Feature Extraction The CNN Feature Extraction aims to learn high-level features from the packet size sequences. Convolutional neural networks (CNNs) are well-suited for this task because they can automatically learn patterns in sequential data. CNNs capture local dependencies within each flow and can detect irregular traffic patterns, which are often indicative of malicious activity.

For each burst $F_i^{(b)}$ in a flow F_i, a convolutional operation is applied to the normalized packet size sequence to extract relevant features. The convolutional

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_19

operation for flow F_i in burst b can be expressed as:

$$\mathbf{F}_i^{(b)} = \text{Conv}(X_i^{(b)}, \mathbf{W}_k) + \mathbf{b}_k \tag{3}$$

where $F_i^{(b)}$ is the feature map for burst b of flow F_i, W_k is the convolution filter, and b_k is the bias term. This operation captures important traffic patterns, such as bursty traffic or regular communication intervals, which are important indicators of potential attacks. After the convolution, we apply a ReLU activation function to introduce non-linearity into the model:

$$\text{ReLU}(x) = \max(0, x) \tag{4}$$

This enables the model to learn discriminative patterns critical for distinguishing benign and malicious traffic.To reduce the dimensionality of the feature maps and focus on the most important features, we apply max pooling. This operation selects the maximum value from each local region of the feature map, ensuring that only the most prominent features are passed to the next layer:

$$\mathbf{F}_i^{(b)\text{pool}} = \text{MaxPool}(\mathbf{F}_i^{(b)}) \tag{5}$$

Max pooling helps reduce computational complexity while retaining critical information, such as significant spikes in traffic.

Bag Creation and Attention Pooling The Bag Creation and Attention Pooling are crucial in enabling the multi-instance learning (MIL) framework to detect malicious encrypted traffic. These phases allow the model to handle multi-flow interactions and focus on the most informative flows, facilitating the detection of complex multi-flow attack patterns that span multiple flows, including when benign and malicious traffic coexist.

In the Bag Creation phase, the model aggregates CNN-extracted flow features into bags based on source-destination address. Each flow is represented by a feature vector obtained from the CNN, which encapsulates high-level patterns such as packet size distributions, flow timing, and burst behavior. These features are grouped into bags, which represent the collective behavior of flows between the same source and destination. This aggregation allows the model to capture multi-flow correlations and identify malicious activities that may span multiple flows.

The bag-level representation for a source-destination address pair i is constructed by aggregating the CNN-extracted feature vectors $\mathbf{F}_j^{(b)}$ from each flow j into a bag B_i:

$$B_i = \{\mathbf{F}_1^{(b)}, \mathbf{F}_2^{(b)}, \ldots, \mathbf{F}_n^{(b)}\} \tag{6}$$

where $\mathbf{F}_j^{(b)}$ is the CNN-extracted feature vector for flow j in bag B_i, and n is the number of flows in the bag. This aggregation allows the model to learn attack patterns that become apparent only when considering multiple flows together, such as botnet communications or DDoS attacks. Bag Creation enables the model

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_19

to capture multi-flow dependencies, which is essential for detecting attacks that span multiple flows. While individual flows may appear benign, aggregating the CNN-extracted features into a bag allows the model to capture coordinated malicious patterns that emerge when multiple flows are analyzed collectively.

After the flow features are aggregated into bags, the next phase is attention pooling, which assigns different importance weights to each flow within the bag based on its relevance to the detection task. The primary challenge in detecting malicious encrypted traffic lies in the presence of mixed benign and malicious flows. The attention pooling mechanism is designed to assign higher weights to the flows that carry critical signals of malicious behavior, while reducing the influence of irrelevant or benign flows.

Example of Botnet Communication and Normal Traffic Consider a botnet-hosted machine that is communicating with a command-and-control (C&C) server, but at the same time, it is also running normal services (e.g., hosting a legitimate website or providing file-sharing services). In this scenario, the source-destination address pair involved in botnet communication may also carry legitimate traffic, such as user browsing requests or data transfers. Botnet-related flows might show distinct packet size patterns (e.g., periodic bursts or large data transfers) that signal malicious activity.

Normal flows, on the other hand, might follow patterns that resemble legitimate traffic (e.g., steady packet sizes and regular intervals), making it challenging to distinguish the malicious flows by looking at individual instances alone. When aggregated into a single bag, these flows present a challenge for the model to identify which flows indicate malicious behavior and which are benign. This is where attention pooling becomes crucial. The attention mechanism assigns higher weights to the flows exhibiting suspicious patterns, such as burst traffic or irregular packet sizes, and assigns lower weights to the benign flows, like normal web browsing traffic, ensuring that the model focuses on the critical malicious signals.

The attention weight α_j for the j-th flow in bag B_i is computed using a learnable attention weight vector W_a:

$$\alpha_j = \frac{\exp(\mathbf{F}_j^{(b)\top}\mathbf{W}_a)}{\sum_{l=1}^{n}\exp(\mathbf{F}_l^{(b)\top}\mathbf{W}_a)} \tag{7}$$

where $\mathbf{F}_j^{(b)}$ is the feature vector for flow j in bag B_i and W_a is the attention weight vector. The softmax function ensures that the attention weights α_j sum to 1, allowing the model to prioritize the most relevant flows.

The final bag-level representation B_i' is obtained by performing a weighted sum of the flow features:

$$\mathbf{B'}_i = \sum_{j=1}^{n}\alpha_j\mathbf{F}_j^{(b)} \tag{8}$$

This process ensures that the model emphasizes the most informative flows in each bag, improving its ability to detect malicious encrypted traffic while reducing the impact of irrelevant flows.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_19

Classification After the attention pooling phase, the final bag-level representation B'_i is passed through a fully connected layer to make the final classification decision. This step enables the model to combine the relevant features from all flows in the bag and classify the traffic as either benign or malicious.

The model is trained using binary cross-entropy as the loss function, appropriate for binary classification tasks. The binary cross-entropy loss measures the error between the true labels and predicted probabilities, guiding the model to adjust its weights to reduce the misclassification rate.

The loss for each bag B'_i is calculated as:

$$\mathcal{L}_i = -\left[y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)\right] \tag{9}$$

where y_i is the true label for bag B_i (1 for malicious, 0 for benign), and $\hat{y}_i$ is the predicted probability of the traffic being malicious. The total loss function is the average loss across all bags:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_i \tag{10}$$

where N is the number of bags in the training set. The goal during training is to minimize this loss function by adjusting the model's parameters to better predict the classification labels.

5 EXPERIMENTAL EVALUATION

In this section, we present a comprehensive evaluation of our proposed model for detecting malicious encrypted traffic, including an assessment of the model's effectiveness, robustness, and computational efficiency.

5.1 Dataset

For our evaluation, we select three publicly available datasets: AndMal2017 [18], CICMalDroid 2020 [19], and Malware Traffic Analysis [7] (MTA). These datasets are chosen because they contain a diverse range of malicious traffic types and are commonly used in the literature for evaluating malware traffic detection methods. They also provide a realistic environment for testing our model's ability to detect malicious encrypted traffic, which is central to our research. Below, we provide a detailed overview of each dataset and explain which subset of data we utilized for this study.

– **AndMal2017/CICMalDroid 2020 [18, 19]:** These datasets contain labeled Android malware traffic from over ten malware families (e.g., Anubis, Cerberus). We extract 10,124 TLS flows from AndMal2017 and 15,402 TLS flows from AndMal2020, ensuring temporal separation for model generalization testing.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_19

- **Malware Traffic Analysis [7] (MTA):** This dataset includes traffic captures from various malware infections. We select 3,121 PCAP files from 2018–2024 due to its frequent updates and diverse range of malware behaviors. The dataset contains both malicious and benign encrypted traffic, presenting a challenge in detecting malware amidst normal encrypted flows.
- **Benign Traffic:** We collected benign traffic by continuously accessing the top 80,000 Alexa-ranked websites using a sandbox and automated scripts. Specifically, Our scripts emulated diverse user activities: (1) dynamic web browsing with Selenium, (2) adaptive-bitrate video streaming, and (3) file transfers of common types. This dataset captures typical user interactions and network behaviors, serving as a baseline for distinguishing malicious from benign encrypted traffic. From this dataset, we selected 500,000 TLS flows, ensuring a diverse mix of traffic types, including web browsing, video streaming, and other common user activities.

5.2 Detection Performance

Baselines We evaluate our model against three representative detection approaches: 1) a deep learning-based method–FS-Net [17], 2) a traditional machine learning approach–ETA [2], and 3) a graph-based learning technique–ST-Graph [10]. We select these baselines to cover diverse technical paradigms: deep learning for end-to-end sequence modeling, traditional ML for hand-crafted feature validation, and graph-based methods for explicit flow relationship analysis, ensuring comprehensive performance comparison. The details of the baselines are as follows.

- **FS-Net [17]:** An end-to-end deep learning model using a multi-layer encoder-decoder architecture to extract sequential features directly from raw network flows. It captures temporal patterns in encrypted traffic by learning hierarchical representations from packet size sequences and classifying flows using fully connected layers, making it effective for detecting subtle anomalies.
- **ST-Graph [10]:** A graph-based approach that models network traffic by representing relationships between endpoints and flows as a graph. Using graph convolutional networks, it captures structural dependencies and coordinated patterns, enabling the detection of complex attack behaviors such as coordinated malware communications.
- **ETA [2]:** A traditional machine learning method that uses a random forest classifier. It extracts TLS handshake metadata, DNS contextual streams, and HTTP header information from surrounding traffic to detect malware, leveraging both encrypted communication and contextual data to differentiate malicious from benign activity.

We configure the baseline hyperparameters based on either the values recommended by their respective authors or the default settings. For instance, for FS-Net, we employ a hidden state dimension of 128, utilized 2 layers, and set the length embedding dimension to 16. For other models, such as the random forest classifier, we use the default parameters provided by scikit-learn.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_19

Environment All experiments are conducted on a workstation running Ubuntu 18.04 LTS, equipped with an Intel i7-12700 CPU, 32 GB of RAM, and an NVIDIA GeForce RTX 3060Ti GPU. Our implementation is based on Python using the PyTorch framework.

Table 1: Hyper-parameters used in the model

Model Layer	**Parameter Name**	**Default Setting**
Learning Rate	Initial Learning Rate	1×10^{-4} (with step decay schedule)
CNN Layer	Kernel Size	3×3, 5×5, 7×7
	Filters	32, 64, 128
Attention Mechanism	Dropout Rate	0.5
Fully Connected Layer	Hidden Layer Units	256
Other Parameters	Batch Size	128
	Epochs	50

Hyper-parameter Key hyper-parameters are determined via grid search combined with 10-fold cross-validation to achieve an optimal balance between accuracy and efficiency,as shown in table 1.

Detection Results Our proposed model consistently outperforms the baseline methods across all three datasets—as shown in Table 2.

For instance, on AndMal2017, our model improves precision from 86.90% (as achieved by ST-Graph) to 88.50% while reducing the false positive rate (FPR) from 0.015% to 0.012%. Similar enhancements are observed on CICMalDroid 2020, where our model attains a precision of 90.50% and a recall of 89.20%, compared to ST-Graph's 88.30% and 87.50%, respectively. Even on the more challenging MTA dataset, our method leads with 84.20% precision and 82.60% recall, outperforming the competitors by approximately 3–4 percentage points. These consistent gains indicate that our approach is not only adept at accurately detecting malicious traffic but also maintains a lower rate of false alarms—a critical balance in scenarios where both detection accuracy and operational efficiency are paramount.

In practical deployment, such improvements translate into tangible benefits. Consider an enterprise network handling tens of thousands of connections daily: a reduction in FPR from 0.015% to 0.012% might seem marginal, yet it can result in significantly fewer false alerts, thereby enabling security analysts to focus on genuine threats. Likewise, on CICMalDroid 2020, the 2-percentage-point boost in recall means that our model is more capable of identifying subtle malware communications that other methods might overlook. Even in environments with high background noise, as in the MTA dataset, our model's superior precision and recall ensure reliable detection of encrypted malicious traffic. These results underscore the robustness, scalability, and practical advantages of our approach in real-world, diverse network environments.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_19

Table 2: Detection performance comparison across datasets.

Dataset	Method	Precision (%)	Recall (%)	FPR (%)
And2017	FS-Net	83.20	82.00	0.180
	ST-Graph	86.90	87.10	0.015
	ETA	85.10	83.80	0.160
	Our Model	**88.50**	**87.90**	**0.012**
CIC2020	FS-Net	85.40	83.70	0.170
	ST-Graph	88.30	87.50	0.014
	ETA	86.10	84.60	0.150
	Our Model	**90.50**	**89.20**	**0.010**
MTA	FS-Net	78.30	75.40	0.210
	ST-Graph	81.50	80.10	0.032
	ETA	79.40	76.80	0.190
	Our Model	**84.20**	**82.60**	**0.028**

5.3 Ablation Study

To validate the effectiveness of key components in our proposed model, we conduct ablation studies by removing or replacing individual modules and evaluating the impact on detection performance-as shown in figure 2.

Burst Segmentation (Precision: 76.4 vs. 84.2) Taking MTA as an example, Removing temporal segmentation causes a significant precision drop (7.8%) and increased FPR (0.042 vs. 0.028). This validates its necessity in capturing short-term attack patterns (e.g., 50ms C&C heartbeat bursts), where global sequence analysis misses time-localized anomalies.

Attention Pooling (Recall: 78.3 vs. 82.6) Taking AndMal2017 as an example, Replacing attention with average pooling reduces recall by 4.3%, primarily due to false negatives in mixed-traffic scenarios. Attention weights (α) prioritize malicious flows (84% of instances with $\alpha > 0.8$ are ground-truth malicious), demonstrating its ability to suppress benign noise.

Bag Aggregation (FPR: 0.046 vs. 0.028) Random-flow grouping (disregarding source-destination address) increases FPR by 64%, indicating our endpoint-oriented method effectively identifies coordinated attacks (e.g., cross-flow DDoS synchronization) that random aggregation obscures.

Learned vs. Hand-crafted Features Hand-crafted TLS metadata (ciphers, certificates) performs worst (Precision: 69.5), as adversarial manipulation of metadata (e.g., spoofed SNI) bypasses static rules. By contrast, CNN-learned packet size dynamics model persistent statistical patterns (e.g., bimodal size distributions in beaconing).

5.4 Robustness

To ensure the practical applicability of our model in real-world environments, we conduct a comprehensive robustness evaluation across multiple challenging sce-

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_19

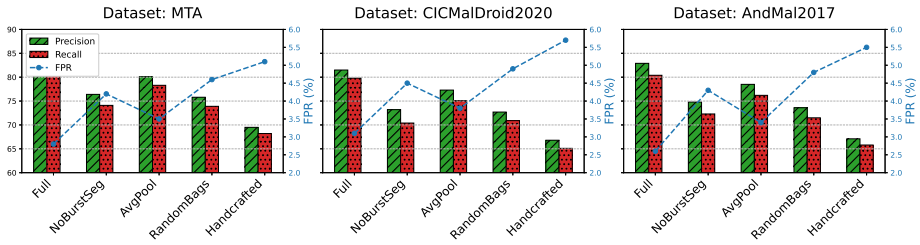

Fig. 2: Cross-Dataset Ablation Study: Precision, Recall, and FPR Comparison

narios, as shown in figure 3. Robustness is assessed in terms of the model's ability to maintain detection performance when faced with noisy environments, temporal distribution shifts, and adversarial evasion attempts. This section presents three core aspects of robustness: resilience to background noise, generalization across temporal variations, and resistance to adversarial manipulation. The results highlight the advantages of our attention-based multi-instance learning (MIL) framework in identifying malicious behaviors even under adverse conditions.

Context-Aware Noise Resistance To evaluate robustness against background traffic, we inject increasing ratios of benign flows into malicious bags Our model maintains 81% recall even with 70% added noise (vs. ST-Graph's 63% at same noise level), thanks to attention-based suppression of irrelevant flows. Case analysis shows 92% of benign injections receive attention weights $\alpha_j < 0.1$.

Temporal Generalization We test temporal cross-dataset generalization by training on AndMal2017 and testing on CICMalDroid 2020. Our model achieves 23% higher F1-score than FS-Net, demonstrating better adaptation to evolving malware tactics. The performance gap primarily stems from previously unseen TLS 1.3 traffic patterns, where our burst-level CNN features generalize better than flow-level sequence models.

Adversarial Scenario Analysis We simulate evasion attacks where adversaries perturb packet sizes by $\pm 10\%$ to mimic benign patterns. Despite this manipulation, our model detects 79% of attacks (vs. 43% for ST-Graph), as shown in figure 3. The MIL architecture's multi-flow perspective increases resilience – while individual flows may disguise themselves, coordinated malicious flows reveal anomalies through their collective burst timing and size correlations.

6 Conclusion

This paper introduces a multi-instance learning framework for detecting encrypted malicious traffic, using bag aggregation and attention mechanisms to address mixed-flow challenges. Experiments demonstrate improved precision and robustness over state-of-the-art methods. While effective for TLS, its protocol scope and attention interpretability could be enhanced. Future work will broaden

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_19

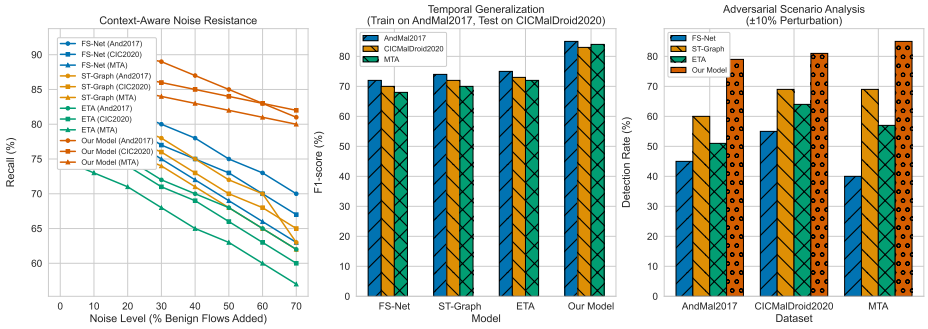

Fig. 3: Results of Robustness Testing

protocol support (e.g., QUIC) and pursue lightweight designs for real-time deployment, advancing its practical utility in evolving cybersecurity landscapes.

Acknowledgments. This work is supported by the Scaling Program of Institute of Information Engineering,CAS (Grant No. E3Z0041101).

References

1. Althouse, J.: Tls fingerprinting with ja3 and ja3s (2019), https://engineering.salesforce.com/tls-fingerprinting-with-ja3-and-ja3s-247362855967, accessed: 2025-02-28
2. Anderson, B., McGrew, D.: Identifying encrypted malware traffic with contextual flow data. In: Proceedings of the 2016 ACM workshop on artificial intelligence and security. pp. 35–46 (2016)
3. Bahramali, A., Soltani, R., Houmansadr, A., Goeckel, D., Towsley, D.: Practical traffic analysis attacks on secure messaging applications. arXiv preprint arXiv:2005.00508 (2020)
4. Bro, P.V.: A system for detecting network intruders in real-time. In: Proc. 7th USENIX security symposium (1998)
5. Carbonneau, M.A., Cheplygina, V., Granger, E., Gagnon, G.: Multiple instance learning: A survey of problem characteristics and applications. Pattern recognition **77**, 329–353 (2018)
6. Cimpanu, C.: Nopen is the equation group's backdoor for unix systems (2022)
7. Duncan, B.: Malware-traffic-analysis.net (2024), https://malware-traffic-analysis.net, accessed: 2025-02-28
8. Fu, C., Li, Q., Shen, M., Xu, K.: Detecting tunneled flooding traffic via deep semantic analysis of packet length patterns. In: Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security. pp. 3659–3673 (2024)
9. Fu, C., Li, Q., Xu, K.: Detecting unknown encrypted malicious traffic in real time via flow interaction graph analysis. arXiv preprint arXiv:2301.13686 (2023)
10. Fu, Z., Liu, M., Qin, Y., Zhang, J., Zou, Y., Yin, Q., Li, Q., Duan, H.: Encrypted malware traffic detection via graph-based network analysis. In: Proceedings of the

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_19

25th International Symposium on Research in Attacks, Intrusions and Defenses. pp. 495–509 (2022)
11. Gallagher, S.: Nearly half of malware now use tls to conceal communications. Sophos news (2021)
12. Gezer, A., Warner, G., Wilson, C., Shrestha, P.: A flow-based approach for trickbot banking trojan detection. Computers & Security **84**, 179–192 (2019)
13. He, Y., Huang, P., Hong, W., Luo, Q., Li, L., Tsui, K.L.: In-depth insights into the application of recurrent neural networks (rnns) in traffic prediction: A comprehensive review. Algorithms **17**(9), 398 (2024)
14. Hong, Y., Li, Q., Yang, Y., Shen, M.: Graph based encrypted malicious traffic detection with hybrid analysis of multi-view features. Information Sciences **644**, 119229 (2023)
15. Jiang, X., Zhang, H.R., Zhou, Y.: Multi-granularity abnormal traffic detection based on multi-instance learning. IEEE Transactions on Network and Service Management **21**(2), 1467–1477 (2023)
16. Li, W., Zhang, X.Y., Bao, H., Shi, H., Wang, Q.: Prograph: Robust network traffic identification with graph propagation. IEEE/ACM Transactions on Networking **31**(3), 1385–1399 (2022)
17. Liu, C., He, L., Xiong, G., Cao, Z., Li, Z.: Fs-net: A flow sequence network for encrypted traffic classification. In: IEEE INFOCOM 2019-IEEE Conference On Computer Communications. pp. 1171–1179. IEEE (2019)
18. Mahdavifar, S., Alhadidi, D., Ghorbani, A.A.: Effective and efficient hybrid android malware classification using pseudo-label stacked auto-encoder. Journal of network and systems management **30**(1), 22 (2022)
19. Mahdavifar, S., Kadir, A.F.A., Fatemi, R., Alhadidi, D., Ghorbani, A.A.: Dynamic android malware category classification using semi-supervised deep learning. In: 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech). pp. 515–522. IEEE (2020)
20. Mimura, M., Otsubo, Y., Tanaka, H., Tanaka, H.: A practical experiment of the http-based rat detection method in proxy server logs. In: 2017 12th Asia Joint Conference on Information Security (AsiaJCIS). pp. 31–37. IEEE (2017)
21. Nasr, M., Bahramali, A., Houmansadr, A.: Deepcorr: Strong flow correlation attacks on tor using deep learning. In: Proceedings of the 2018 ACM SIGSAC conference on computer and communications security. pp. 1962–1976 (2018)
22. Nelms, T., Perdisci, R., Ahamad, M.: {ExecScent}: Mining for new {C&C} domains in live networks with adaptive control protocol templates. In: 22nd USENIX Security Symposium (USENIX Security 13). pp. 589–604 (2013)
23. Rescorla, E.: The transport layer security (tls) protocol version 1.3. Tech. rep., Internet Engineering Task Force (IETF) (2018)
24. Roesch, M., et al.: Snort: Lightweight intrusion detection for networks. In: Lisa. pp. 229–238 (1999)
25. Wang, G., Gu, Y.: Multi-task scenario encrypted traffic classification and parameter analysis. Sensors **24**(10), 3078 (2024)
26. Zhang, Y., Chen, X., Jin, L., Wang, X., Guo, D.: Network intrusion detection: Based on deep hierarchical network and original flow data. IEEE Access **7**, 37004–37016 (2019)

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97629-2_19