Symbolic Flow Representation Based on the First-M Packets for Early Traffic Classification

Marcelo A. C. Fernandes[1]

[1]InovAI Lab – nPITI/IMD – UFRN
Leading Advanced Technologies Center of Excellence (LANCE) – nPITI/IMD – UFRN
Department of Computer Engineering and Automation (DCA) – UFRN
Natal – RN – Brazil

mfernandes@dca.ufrn.br

Abstract. *This paper presents a symbolic flow-level representation for early network traffic classification based on the first M packets of a bidirectional flow. Each packet payload is converted into a symbolic token sequence and embedded using a transformer-based sentence embedding model, followed by flow-level aggregation. The resulting embeddings are evaluated using XGBoost for binary VPN mode classification and multiclass traffic-type classification. Experiments are conducted on the ISCX VPN-nonVPN dataset using repeated balanced hold-out validation. Results show that the proposed representation enables accurate discrimination between VPN and non-VPN traffic and supports traffic-type identification under both VPN and non-VPN settings.*

1. Introduction

Network traffic classification is a fundamental task for network management and security, supporting applications such as service identification, anomaly detection, and policy enforcement. The widespread adoption of encryption mechanisms, including VPN tunnels and application-layer encryption, has reduced the effectiveness of traditional approaches based on payload inspection and protocol-specific features. Consequently, recent research has shifted toward learning-based methods operating on flow-level representations, particularly for encrypted and VPN traffic [Azab et al. 2024, Dong et al. 2025]. However, many existing approaches rely on time-based observation windows or complex feature engineering, which introduces variability in decision latency and limits applicability to early classification scenarios [Azab et al. 2024, Nascita et al. 2024]. These limitations motivate the development of representations that avoid explicit temporal dependencies while remaining effective when only limited early traffic observations are available.

Several studies have investigated learning-based approaches for network traffic classification in encrypted communication scenarios. The work presented in [Lotfollahi et al. 2020] proposed *Deep Packet*, which applies deep neural networks directly to raw packet payloads, showing that discriminative patterns can be learned under encryption. More recently, the work presented in [Lin et al. 2022] introduced ET-BERT, leveraging transformer-based pre-training to obtain contextualized representations from encrypted datagrams. Other efforts have explored classical machine learning in flow-based settings, often combined with Software-Defined Networking, as reported in [Salau and Beyene 2024, Najm et al. 2024], where engineered statistical features are used for traffic classification. Although these approaches achieve competitive results,

they commonly depend on deep architectures with high computational cost or on feature representations derived from time-based aggregation and protocol-specific characteristics. Additional work has addressed encrypted traffic classification using broader learning paradigms, including uncertainty-aware models, mobile traffic analysis, and label-efficient strategies. The work presented in [Jorgensen et al. 2023] explored uncertainty quantification for adaptive traffic labeling in VPN environments, while the work presented in [Aceto et al. 2018] applied deep learning to mobile encrypted traffic. More recently, the work presented in [Eslami and Hamouda 2025] combined self-supervised and confident learning to mitigate label scarcity and noise. Despite addressing important challenges, these approaches generally rely on temporal aggregation, statistical flow features, or complex training pipelines, motivating alternative representations that support early classification based on limited packet observations.

Despite the progress achieved by learning-based approaches for encrypted traffic classification, several gaps remain in the literature. Many existing methods depend on time-based aggregation, flow duration, or handcrafted feature extraction, resulting in variable decision latency and reduced suitability for early classification scenarios. In addition, deep architectures and large-scale pre-training often incur high computational cost, while flow-based machine learning solutions commonly depend on statistical features that are sensitive to protocol behavior or temporal dynamics. Furthermore, most studies address VPN detection and application classification as separate problems, rather than under a unified and deterministic representation. In this context, the present work introduces a Symbolic Flow Representation based on the First-M packets (SFR-M), which operates directly on payload-derived token sequences and avoids explicit temporal metrics. By combining packet-level symbolic encoding, transformer-based embeddings, and a simple aggregation scheme, SFR-M supports early classification with deterministic latency and fixed computational cost per flow.

The methodology adopted in this work combines a symbolic flow-level representation with transformer-based embeddings and supervised learning for early traffic classification. Network flows are constructed from the first M packets of each bidirectional communication, and packet payloads are converted into symbolic token sequences that are subsequently embedded and aggregated at the flow level. The evaluation is conducted using the ISCX VPN-nonVPN dataset, originally introduced in [Gil et al. 2016], which provides labeled encrypted and VPN traffic across multiple application categories under controlled experimental conditions. Classification experiments are performed using XGBoost under a repeated balanced holdout protocol, addressing both binary VPN mode detection and multiclass traffic-type identification for Non-VPN and VPN scenarios. The reported results indicate that the proposed representation supports accurate classification from early packet observations while maintaining deterministic latency and independence from explicit temporal features. The flow-level token datasets and the corresponding embedding datasets generated in this work are publicly available in Parquet format to support reproducibility and further research. The complete datasets, including configurations for $b = 2$ and $b = 8$, are available at Mendeley Data [Fernandes 2026].

2. Proposed Method

This work introduces a network traffic characterization and classification pipeline, referred to as the SFR-M, designed for continuous operation with a focus on early clas-

sification and independence from traditional time-based metrics. The system adopts the bidirectional flow as the fundamental unit of analysis, identified by a canonical 5-tuple, treating both communication directions as a single logical entity. Accordingly, a bidirectional flow defined by a canonical 5-tuple can be expressed as $\phi = (\mathrm{IP}_c, \mathrm{IP}_s, p_c, p_s, \pi)$, where IP_c and IP_s denote the IP addresses of the communicating endpoints, p_c and p_s the transport-layer ports, and π the transport protocol. The flow is treated as a single logical entity that aggregates both directions of communication. Each packet belonging to the flow is assigned to one of the two directions, denoted as $c \rightarrow s$ and $s \rightarrow c$. This modeling reflects the concept of a dialogue between endpoints and enables the capture of relevant structural patterns from the earliest observed packets, without relying on the total flow duration or fixed time windows.

Unlike time-based approaches, the proposed method explicitly defines traffic observation based on a fixed number of initial packets in each communication direction ($c \rightarrow s$ and $s \rightarrow c$). For each i-th flow, denoted as ϕ_i, the first M packets in the client-to-server direction ($c \rightarrow s$) and the first M packets in the server-to-client direction ($s \rightarrow c$) are considered. Time plays only an operational role, being used to terminate inactive flow states through an idle-timeout mechanism, thus preventing unbounded growth of the internal state. In this way, the amount of evidence used for characterization is controlled by the order of packet arrivals, while the decision latency is directly associated with the progression of the dialogue rather than elapsed time. Thus, for each i-th flow ϕ_i, at most M packets are considered in each direction, resulting in the set

$$\mathcal{P}_{\phi_i} = \{p_{i,1}^{c \rightarrow s}, \ldots, p_{i,M}^{c \rightarrow s}\} \cup \{p_{i,1}^{s \rightarrow c}, \ldots, p_{i,M}^{s \rightarrow c}\} \tag{1}$$

where , $p_{i,k}^{c \rightarrow s}$ is the k-th packet in the $c \rightarrow s$ direction of the i-th flow ϕ_i, and $p_{i,k}^{s \rightarrow c}$ is the k-th packet in the $s \rightarrow c$ direction of the same i-th flow ϕ_i. In the proposed method, time does not define the content of the observation and is used only as an operational criterion for terminating inactive flows through an idle-timeout. An instance associated with the i-th flow ϕ_i is emitted as soon as both packet subsets reach cardinality M, enabling early classification and deterministic latency.

During operation, each received packet undergoes minimal parsing at the network and transport layers, sufficient only to update the flow state and determine its relative direction. There is no need for session reconstruction, segment reassembly, or deep semantic inspection of the payload. A flow instance is considered complete as soon as the first M packets in both communication directions are observed, at which point the classification decision can be taken immediately. Accordingly, for each k-th selected packet of a given direction ($c|s \rightarrow s|c$) associated with the i-th flow, denoted as $p_{i,k}^{c|s \rightarrow s|c}$, a short binary signature $\mathbf{x}_{i,k}^{c|s \rightarrow s|c}$ is extracted from the beginning of the packet payload. This signature is defined as the sequence of the first N consecutive available bits and is expressed as

$$\mathbf{x}_{i,k}^{c|s \rightarrow s|c} = \left[x_{i,k,1}^{c|s \rightarrow s|c}, \ldots, x_{i,k,N}^{c|s \rightarrow s|c}\right] \in \{0,1\}^N, \tag{2}$$

where $x_{i,k,j}^{c|s \rightarrow s|c} \in \{0,1\}$. This binary signature $\mathbf{x}_{i,k}^{c|s \rightarrow s|c}$, associated with each k-th packet of the i-th flow, is then converted into a sequence of tokens through a quantization process over blocks of b bits, resulting in

$$\mathbf{t}_{i,k}^{c|s \rightarrow s|c} = \left[t_{i,k,1}^{c|s \rightarrow s|c}, \ldots, t_{i,k,J}^{c|s \rightarrow s|c}\right], \quad J = \frac{N}{b}, \tag{3}$$

where $b < N$ and b is an even integer, i.e., $b \in \{2, 4, 6, 8, \dots\}$. Each j-th token associated with the k-th packet of the i-th flow in either direction ($c|s \rightarrow s|c$), denoted as $t_{i,k,j}^{c|s \rightarrow s|c}$, is formed by a sequence of symbols belonging to a discrete alphabet $\Sigma = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$, in which each symbol is encoded using two bits.

In the next stage, the token sequences associated with the k-th packet of the i-th flow, denoted as $\mathbf{t}_{i,k}^{c|s \rightarrow s|c}$, are transformed into dense vector representations by means of transformer-based models of the sentence embedding type. Each packet signature is treated as an independent textual unit, avoiding truncation issues and preserving the structural granularity of the initial communication. Thus, each token sequence associated with the k-th packet of the i-th flow ϕ_i is mapped to a dense vector representation through an embedding function, resulting in the dense vector

$$\mathbf{e}_{i,k}^{c|s \rightarrow s|c} = \left[e_{i,k,1}^{c|s \rightarrow s|c}, \dots, e_{i,k,D}^{c|s \rightarrow s|c} \right], \tag{4}$$

where D denotes the embedding dimensionality.

Prior to the aggregation stage, each packet-level embedding vector is normalized using the ℓ_2 norm in order to ensure comparable magnitudes across packets. Formally, the normalized embedding is defined as

$$\tilde{\mathbf{e}}_{i,k}^{c|s \rightarrow s|c} = \frac{\mathbf{e}_{i,k}^{c|s \rightarrow s|c}}{\left\| \mathbf{e}_{i,k}^{c|s \rightarrow s|c} \right\|_2} = \frac{\mathbf{e}_{i,k}^{c|s \rightarrow s|c}}{\sqrt{\sum_{j=1}^{D} \left(e_{i,k,j}^{c|s \rightarrow s|c} \right)^2}}. \tag{5}$$

As, for each i-th flow ϕ_i, M packets are used (see Equation 1), the embedding matrices are formed as

$$\mathbf{E}_i^{c \rightarrow s} = \begin{bmatrix} \tilde{\mathbf{e}}_{i,1}^{c \rightarrow s} \\ \vdots \\ \tilde{\mathbf{e}}_{i,M}^{c \rightarrow s} \end{bmatrix} = \begin{bmatrix} \tilde{e}_{i,1,1}^{c \rightarrow s} & \dots & \tilde{e}_{i,1,D}^{c \rightarrow s} \\ \vdots & \ddots & \vdots \\ \tilde{e}_{i,M,1}^{c \rightarrow s} & \dots & \tilde{e}_{i,M,D}^{c \rightarrow s} \end{bmatrix} \tag{6}$$

and

$$\mathbf{E}_i^{s \rightarrow c} = \begin{bmatrix} \tilde{\mathbf{e}}_{i,1}^{s \rightarrow c} \\ \vdots \\ \tilde{\mathbf{e}}_{i,M}^{s \rightarrow c} \end{bmatrix} = \begin{bmatrix} \tilde{e}_{i,1,1}^{s \rightarrow c} & \dots & \tilde{e}_{i,1,D}^{s \rightarrow c} \\ \vdots & \ddots & \vdots \\ \tilde{e}_{i,M,1}^{s \rightarrow c} & \dots & \tilde{e}_{i,M,D}^{s \rightarrow c} \end{bmatrix}, \tag{7}$$

where $\mathbf{E}_i^{c \rightarrow s}$ stores the embeddings of the first M packets of the i-th flow in the client-to-server direction ($c \rightarrow s$), and $\mathbf{E}_i^{s \rightarrow c}$ stores the embeddings of the first M packets in the server-to-client direction ($s \rightarrow c$). After constructing the embedding matrices, an operation is applied to obtain a single embedding per direction, which can be expressed as

$$\mathbf{z}_i^{c \rightarrow s} = \mathbf{w}\, \mathbf{E}_i^{c \rightarrow s} = \left[z_{i,1}^{c \rightarrow s}, \dots, z_{i,D}^{c \rightarrow s} \right] \tag{8}$$

and

$$\mathbf{z}_i^{s \rightarrow c} = \mathbf{w}\, \mathbf{E}_i^{s \rightarrow c} = \left[z_{i,1}^{s \rightarrow c}, \dots, z_{i,D}^{s \rightarrow c} \right], \tag{9}$$

where $\mathbf{z}_i^{c \rightarrow s}$ and $\mathbf{z}_i^{s \rightarrow c}$ are D-dimensional vectors representing the directional embeddings of the i-th flow ϕ_i, and $\mathbf{w}$ is a weight vector that aggregates the embeddings of the M packets associated with each direction, given by

$$\mathbf{w} = [w_1, \dots, w_M]. \tag{10}$$

To obtain a single representation for each i-th flow ϕ_i, a max-based aggregation operator is applied between the directional embeddings $\mathbf{z}_i^{c\to s}$ and $\mathbf{z}_i^{s\to c}$, that is,

$$\mathbf{z}_i = \max\left(\mathbf{z}_i^{c\to s}, \mathbf{z}_i^{s\to c}\right) = \left[\max\left(z_{i,1}^{c\to s}, z_{i,1}^{s\to c}\right), \ldots, \max\left(z_{i,D}^{c\to s}, z_{i,D}^{s\to c}\right)\right], \tag{11}$$

where $\mathbf{z}_i$ is a vector representing the final signature of the i-th flow ϕ_i. It can be concluded that $\mathbf{z}_i$ captures both the weighted average behavior (see Equations 8 and 9) and the most salient patterns observed in the first M packets of the flow (see Equation 11). Finally, the final vector representation $\mathbf{z}_i$ produced by the proposed SFR-M can be used as input to any supervised machine learning classifier. In this work, XGBoost is adopted as the classification model for all experimental evaluations. This explicit separation between the semantic representation stage and the decision stage allows different classification techniques to be evaluated under the same representational basis, favoring fair and reproducible comparisons. The resulting vector representation has deterministic length and does not depend on identifier fields, semantic payload content, or explicit temporal metrics. Directional separation between client and server is preserved throughout the entire pipeline, enabling asymmetric dialogue patterns to be retained. Moreover, by operating on abstract symbolic sequences, the SFR-M approach reduces the risk of bias associated with specific endpoints and promotes generalization to environments and applications not observed during training.

3. Methodology

3.1. Flow-Based Token Dataset

The dataset used in this work is derived from the ISCX VPN-nonVPN collection [Gil et al. 2016], which contains raw network traffic captured in PCAP format under controlled experimental conditions. The dataset comprises approximately 25 GB of packet traces distributed across 140 PCAP files, including 109 non-VPN files and 31 VPN files, covering multiple application categories such as VoIP, file transfer, chat, email, streaming, and peer-to-peer services. All PCAP files contain full packet captures with payload information, enabling direct extraction of packet-level content without relying on precomputed flow features.

Raw PCAP files are processed by a dedicated pipeline that constructs bidirectional flows and extracts symbolic signatures directly from packet payloads. For each packet considered, the first 64 bytes of the payload are selected, corresponding to $N = 512$ consecutive bits. These bits are converted into a binary representation and subsequently mapped to symbolic tokens using a fixed alphabet $\Sigma = \{\sigma_1 = A, \sigma_2 = B, \sigma_3 = C, \sigma_4 = D\}$. No artificial padding is applied during this process; only bytes effectively observed in the payload are used, preventing the introduction of synthetic patterns. Tokenization is evaluated under three configurations, with $b \in \{2, 8\}$ bits per token, resulting in $J = 256$, and 64 tokens per packet, respectively. Each row in the final dataset corresponds to a bidirectional flow, for which up to $M = 4$ packets are retained in each communication direction, preserving the order of packet observation within the flow. This value was selected as a compromise between capturing sufficient discriminative information and maintaining low computational cost, as the initial packets of a flow often contain characteristic patterns relevant for service identification. Only flows that reach this maximum number of packets in both directions are included in the dataset, ensuring a uniform structure across all samples. Packets without payload are discarded, and all explicitly identifiable information,

such as IP addresses and port numbers, is removed to reduce the risk of endpoint-related bias.

After processing all PCAP files, the resulting dataset contains a total of 1,242,608 flow instances, of which 954,252 correspond to non-VPN traffic and 288,356 correspond to VPN traffic. Each flow instance contains the token sequences extracted from the M packets in both directions, forming a compact and deterministic symbolic representation at the flow level. Table 1 summarizes the class distribution of the dataset after flow construction and tokenization. The resulting dataset is therefore suitable for evaluating early traffic classification methods, as it emphasizes structural patterns present in the initial packets of each flow rather than time-dependent statistics or protocol-specific features.

Table 1. Distribution of flow instances in the constructed dataset.

VPN Mode	Total	VoIP	File Transfer	Chat	Email	Streaming	P2P
Non-VPN	954,252	827,883	119,439	4,969	1,030	931	–
VPN	288,356	280,213	3,245	2,956	740	597	605

3.2. Embedding Dataset

Following the symbolic representation defined in the proposed method, each token sequence $\mathbf{t}_{i,k}^{c|s\to s|c}$ associated with the k-th packet of the i-th flow ϕ_i is mapped to a dense vector representation using a transformer-based sentence embedding model. In this work, we employ the *all-MiniLM-L6-v2* architecture from the Sentence-Transformers framework, which encodes each packet-level token sequence independently into a fixed-dimensional embedding vector $\mathbf{e}_{i,k}^{c|s\to s|c}$. This model supports a maximum input length of 256 tokens, which is sufficient to accommodate all token sequences generated in our experiments for the considered configurations, namely $b = 2$ and $b = 8$ with $N = 512$ bits, resulting in $J = 256$ and $J = 64$ tokens per packet, respectively. Treating each packet as an independent textual unit avoids truncation effects and preserves the structural granularity of the early communication, as all token sequences extracted from the first M packets are fully processed by the transformer encoder. Prior to aggregation, each embedding vector is ℓ_2-normalized to improve numerical stability and ensure comparable vector magnitudes across packets.

In the aggregation stage, the weight vector $\mathbf{w}$ used in Equations 8 and 9 is defined as a uniform weighting scheme, given by $\mathbf{w} = \left[\frac{1}{M}, \frac{1}{M}, \ldots, \frac{1}{M}\right]$, that is, $w_k = \frac{1}{M}$ for all $k = 1, \ldots, M$. With this definition, the aggregation of packet-level embeddings within each communication direction corresponds to the arithmetic mean of the M normalized embeddings associated with that direction. This choice ensures that all packets contribute equally to the directional representation, while preserving a fixed-dimensional and order-independent summary of the early flow behavior.

For each flow ϕ_i, the packet-level embeddings are aggregated according to the formulation introduced in Section 2. Specifically, embeddings from the M packets in each direction are combined using a mean–maximum aggregation strategy, producing the directional vectors $\mathbf{z}_i^{c\to s}$ and $\mathbf{z}_i^{s\to c}$. The final flow-level representation $\mathbf{z}_i$ is then obtained by applying an element-wise maximum between the two directional vectors, resulting in a single embedding of dimension D per flow. The resulting embedding dataset there-

fore consists of compact, deterministic, and direction-aware flow representations, directly aligned with the symbolic formulation of the proposed method.

3.3. ML Classification

This subsection describes the supervised learning protocol adopted to evaluate the proposed flow-level representation $\mathbf{z}_i$ under different tokenization configurations. The experiments consider two tokenization settings, $b \in \{2, 8\}$, with fixed parameters $N = 512$ bits and $M = 4$ packets per direction. Classification is performed exclusively using XGBoost, which is employed as the supervised learning model for all evaluation scenarios. For each tokenization setting, the binary classification task of distinguishing non-VPN from VPN traffic is first addressed using the flow-level embeddings as input features.

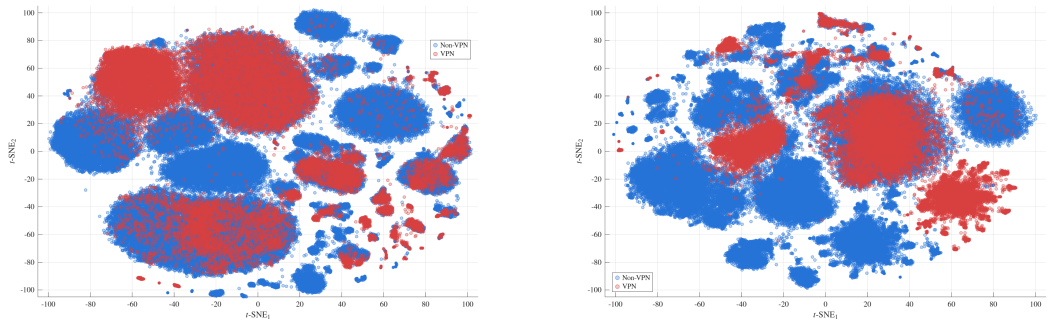
To mitigate the effects of class imbalance, a minimum-class balancing strategy based on random under-sampling is adopted. At each repetition, the training set is constructed by sampling the same number of instances from each class, corresponding to the size of the minority class. After balancing, a stratified holdout split is applied, with 80% of the balanced instances used for training and 20% for testing. This procedure is repeated 10 times using independent re-sampling and re-splitting, and performance metrics are reported as the aggregation of results across the 10 repetitions. This evaluation protocol ensures consistent and reproducible assessment of the proposed representation under controlled conditions. In addition to VPN mode detection, multiclass traffic-type classification is evaluated separately for non-VPN and VPN traffic. In this case, models are trained and tested using only the subset of flows corresponding to each VPN mode. The same minimum-class under-sampling strategy is applied across all traffic-type classes, followed by a stratified $80/20$ holdout split, repeated 10 times. This design results in three evaluation scenarios for each tokenization setting: (i) binary VPN mode classification on the full dataset, (ii) traffic-type classification restricted to non-VPN flows, and (iii) traffic-type classification restricted to VPN flows, all evaluated under identical balancing and repeated holdout conditions.

4. Results and Analysis

4.1. Data Analysis

The t-distributed Stochastic Neighbor Embedding (t-SNE) technique is applied to project the high-dimensional flow embeddings $\mathbf{z}_i$ into a two-dimensional space for exploratory data analysis. Figures 1a and 1b present the resulting projections for the binary classification task between non-VPN and VPN traffic, considering the two tokenization configurations evaluated in this study, namely $b = 2$ and $b = 8$. In both figures, each point represents a flow-level embedding, and colors indicate the corresponding VPN mode. The projections are generated using the same embedding dataset employed in the classification experiments, ensuring consistency between qualitative visualization and quantitative evaluation.

The projections show different spatial distributions of non-VPN and VPN flows across the two tokenization settings. For $b = 2$, the embedding space contains regions where flows from both classes are interspersed, indicating partial overlap between the two categories. For $b = 8$, the projections display regions with higher concentration of flows associated with the same VPN mode, particularly for VPN traffic. Overlapping regions are



(a) $b = 2$ **(b)** $b = 8$

Figure 1. t-SNE projections of the flow-level embeddings $\mathbf{z}_i$ for VPN mode classification (Non-VPN vs VPN) under two tokenization settings.

still present in both configurations, reflecting the heterogeneity of traffic types included in the dataset. These observations are aligned with the differences in classification results obtained for the two tokenization settings.

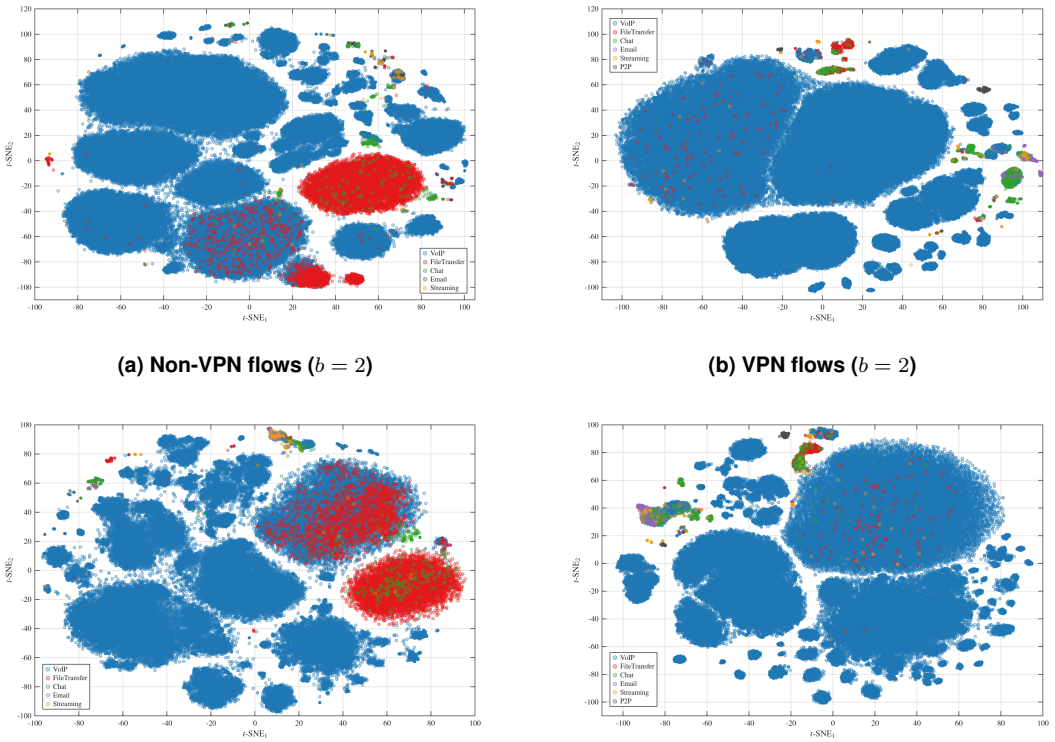
Figures 2a, 2b, 2c, and 2d present two-dimensional t-SNE projections of the flow-level embeddings $\mathbf{z}_i$, generated separately for the Non-VPN and VPN classes under both values of b. In each figure, every point corresponds to a flow-level embedding, and colors indicate the associated traffic type (see Table 1). The projections are computed from the same embedding dataset used in the classification experiments, ensuring consistency between the exploratory visualization and the quantitative evaluation.

The t-SNE projections reveal differences in the spatial organization of flows within each VPN mode across the two tokenization settings. For $b = 2$, both Non-VPN and VPN projections exhibit regions where multiple traffic types are interspersed, indicating overlap in the embedding space. In contrast, the $b = 8$ projections show more localized clusters, particularly within the VPN class, where flows of the same traffic type tend to concentrate in distinct regions. Nevertheless, overlapping regions remain present in both configurations, reflecting the diversity of applications and communication patterns included in the dataset. These observations are consistent with the classification results and indicate that the symbolic representation preserves structural information whose discriminative power varies with the granularity of the tokenization parameter b.

4.2. XGBoost Classification Results

Table 2 reports the XGBoost results for binary VPN mode classification under both values of b. The results are presented as mean and standard deviation over $10\times$ repeated holdout experiments with an $80/20$ train–test split and minimum-class balancing. The evaluated metrics include Accuracy, Precision, Recall, F1-score, ROC-AUC, PR-AUC, and Log-loss. Figure 3 complements these results by illustrating the corresponding confusion matrices, allowing an inspection of class-wise prediction behavior.

The results indicate that both tokenization settings yield high recall for the VPN class, with values above 0.99, reflecting a low rate of false negatives in VPN detection. The configuration with $b = 8$ presents slightly higher Accuracy, F1-score, ROC-AUC, and PR-AUC, as well as lower Log-loss, compared to $b = 2$, suggesting a more confi-



(a) Non-VPN flows ($b = 2$) **(b) VPN flows ($b = 2$)**

(c) Non-VPN flows ($b = 8$) **(d) VPN flows ($b = 8$)**

Figure 2. t-SNE projections of flow-level embeddings $\mathbf{z}_i$ generated separately for Non-VPN and VPN traffic under two tokenization configurations. Each point corresponds to a flow-level embedding, and colors indicate traffic types within the same VPN mode.

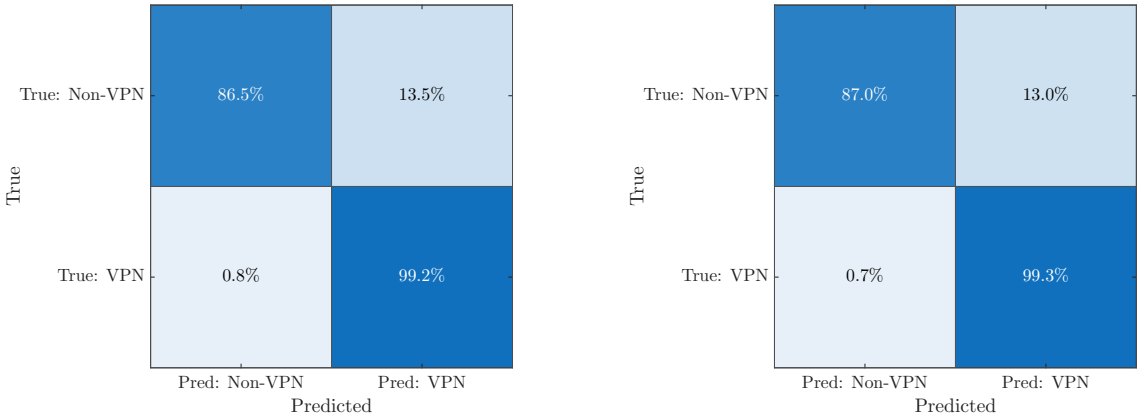
dent separation between the two classes. The confusion matrices in Figure 3 show that most misclassifications occur in the Non-VPN class, while the VPN class is consistently identified with high correctness across both configurations. These results indicate that the proposed flow-level symbolic representation supports reliable VPN mode discrimination using early packet observations, with performance variations that are influenced by the tokenization granularity.

A comparison between the quantitative results in Table 2, the confusion matrices in Figure 3, and the t-SNE visualization in Figure 2b indicates a consistent relationship between the embedding space structure and the observed classification performance. For $b = 2$, the t-SNE projection shows partial overlap between Non-VPN and VPN flows, which is reflected in the confusion matrices by misclassifications concentrated in the Non-VPN class, while the VPN class exhibits a low false-negative rate. Despite this overlap, the quantitative results report high Recall and F1-score values for both tokenization settings, indicating that the learned flow-level representation retains sufficient discriminative information for reliable VPN mode classification. Overall, the qualitative patterns observed in the t-SNE projection are aligned with the class-wise performance trends obtained with XGBoost, highlighting the influence of tokenization granularity on embedding separability and classification outcomes.

Table 3 presents the XGBoost results for multiclass traffic-type classification un-

Table 2. XGBoost results for Non-VPN vs. VPN classification (mean $\pm$ std over $10\times$ repeated holdout with $80/20$ split and minimum-class balancing).

Metric	$b = 2$	$b = 8$
Accuracy	0.9283 ± 0.0007	0.9319 ± 0.0015
Precision	0.8802 ± 0.0014	0.8846 ± 0.0034
Recall	0.9915 ± 0.0010	0.9933 ± 0.0018
F1-score	0.9326 ± 0.0006	0.9358 ± 0.0012
ROC-AUC	0.9827 ± 0.0003	0.9858 ± 0.0004
PR-AUC	0.9821 ± 0.0003	0.9857 ± 0.0004
Log-loss	0.1640 ± 0.0015	0.1528 ± 0.0030



(a) $b = 2$ **(b)** $b = 8$

Figure 3. Confusion matrices for XGBoost-based VPN mode classification under two tokenization configurations ($b = 2$ and $b = 8$).

der Non-VPN and VPN settings, for both values of b. The same balanced holdout protocol described above is adopted, and the evaluated metrics include Accuracy, Macro F1, and Log-loss. Figure 4 complements these results by showing the confusion matrices for each VPN mode and tokenization setting, allowing a detailed inspection of class-wise prediction behavior across traffic types.

The results indicate that traffic-type classification performance varies with both the VPN mode and the tokenization granularity. For Non-VPN traffic, the configuration with $b = 2$ yields higher Accuracy and Macro F1 compared to $b = 8$, while also presenting lower Log-loss. A similar trend is observed for VPN traffic, where $b = 2$ outperforms $b = 8$ across all reported metrics. The confusion matrices in Figure 4 show that most classification errors occur among semantically related traffic types, such as file transfer and streaming, whereas classes such as VoIP and email tend to exhibit higher correct classification rates. These results suggest that coarser symbolic quantization preserves discriminative patterns that are more stable for multiclass traffic-type identification under both VPN and non-VPN scenarios.

A comparison between the quantitative results in Table 3, the confusion matrices in Figure 4, and the t-SNE projections in Figure 2 shows a consistent relationship between the embedding space organization and traffic-type classification performance. Under $b = 2$, the t-SNE projections indicate partial overlap among application classes for both Non-

Table 3. XGBoost results for traffic-type classification under Non-VPN and VPN settings (mean $\pm$ std over $10\times$ repeated holdout with $80/20$ split and minimum-class balancing).

VPN Mode	b	Accuracy	Macro F1	Log-loss
Non-VPN	$b = 2$	0.9356 ± 0.0065	0.9355 ± 0.0065	0.2186 ± 0.0318
Non-VPN	$b = 8$	0.8772 ± 0.0099	0.8760 ± 0.0099	0.3707 ± 0.0429
VPN	$b = 2$	0.9252 ± 0.0042	0.9251 ± 0.0041	0.2699 ± 0.0407
VPN	$b = 8$	0.8877 ± 0.0091	0.8867 ± 0.0093	0.3704 ± 0.0394

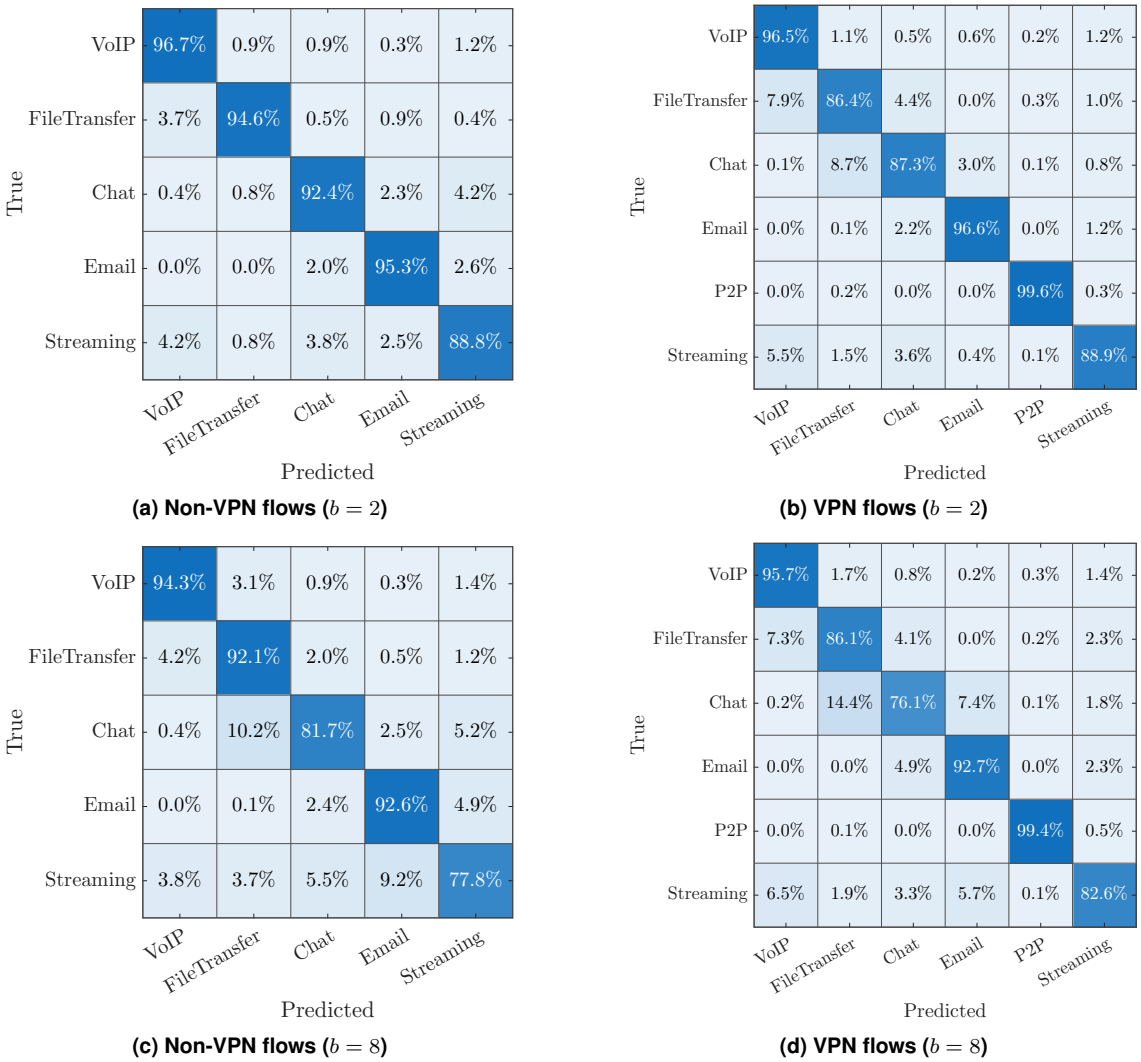
VPN and VPN settings, which is reflected in the confusion matrices by misclassifications concentrated among semantically related traffic types (e.g., file transfer and streaming). Under $b = 8$, the projections exhibit tighter per-application clusters, especially within the VPN subset; however, Table 3 indicates that this increased structural separation does not yield improved overall metrics, as both Accuracy and Macro F1 are higher for $b = 2$ in both scenarios. Overall, the qualitative patterns observed in the t-SNE projections align with the class-wise error patterns and aggregate metrics obtained with XGBoost, highlighting the influence of tokenization granularity on the trade-off between separability and generalization in traffic-type classification.

4.3. Comparison with Related Work

Tables 4 and 5 present a comparison between the proposed approach and representative state-of-the-art methods for encrypted traffic classification. Table 4 focuses on methodological aspects, including input representation, learning model, feature design, and the level at which traffic is modeled, while Table 5 summarizes operational characteristics related to time dependence, early classification capability, computational cost, and suitability for real-time deployment.

From a methodological perspective, Table 4 shows that several state-of-the-art approaches rely either on raw packet bytes or on engineered statistical flow features. Deep learning-based methods, such as Deep Packet and ET-BERT, learn implicit representations directly from packet-level data, but typically operate at the packet or mixed packet/flow level and depend on complex architectures or large-scale pre-training. In contrast, flow-based machine learning approaches rely on manually engineered features aggregated over time windows or full flows. The proposed method differs by introducing a symbolic representation derived from payload tokens and by constructing a flow-level embedding based on the first M packets of each bidirectional flow. This design combines implicit feature learning through a pre-trained transformer with a lightweight supervised classifier, while maintaining a strictly flow-level representation that does not depend on handcrafted features or protocol-specific statistics.

From an operational standpoint, Table 5 highlights that most existing methods are time-dependent, as they rely on observation windows, flow duration, or temporal aggregation to build their representations. As a consequence, early classification is either not supported or only partially addressed, and computational costs can be high, particularly for transformer-based approaches with extensive pre-training. The proposed method explicitly avoids time-based dependencies by limiting observation to a fixed number of initial packets, enabling deterministic decision latency and a bounded computational cost



(a) Non-VPN flows ($b = 2$) **(b) VPN flows ($b = 2$)**

(c) Non-VPN flows ($b = 8$) **(d) VPN flows ($b = 8$)**

Figure 4. Confusion matrices for XGBoost-based traffic-type classification under Non-VPN and VPN settings for two tokenization configurations ($b = 2$ and $b = 8$).

per flow. This characteristic supports early classification while preserving compatibility with real-time network monitoring scenarios. Overall, the comparison indicates that the proposed approach addresses methodological and operational gaps in the literature by enabling early, flow-level traffic classification with moderate computational requirements and without reliance on explicit temporal features.

5. Conclusions

This paper introduced a Symbolic Flow Representation based on the First-M packets (SFR-M) for early network traffic classification under encrypted and VPN scenarios. The proposed method constructs a flow-level representation from symbolic payload tokens extracted from the initial packets of a bidirectional flow, avoiding explicit temporal features and protocol-specific statistics. By combining packet-level symbolic encoding, transformer-based embeddings, and a simple aggregation strategy, SFR-M produces deterministic and compact flow representations suitable for supervised learning. Experi-

Table 4. Methodological comparison between the proposed approach and representative state-of-the-art methods.

Reference	Method	Input Representation	Learning Model	Feature Design	Flow-base
[Lotfollahi et al. 2020]	Deep Packet	Raw packet bytes	CNN/SAE	Implicit (learned)	Packet-level
[Lin et al. 2022]	ET-BERT	Datagram sequences	Pre-trained Transformer	Implicit (pre-trained)	Packet/Flow
[Salau and Beyene 2024]	SDN + ML	Engineered flow features	Classical ML	Manual	Flow-level
[Najm et al. 2024]	Enhanced ML	Statistical flow features	Classical ML	Manual	Flow-level
Proposed Method	SFR-M	Symbolic payload tokens	Pre-trained Transformer + XGBoost	Implicit (symbolic)	Flow-level

Table 5. Operational comparison between the proposed approach and state-of-the-art methods.

Reference	Time-dependent	Early Classification	Computational Cost	Real-time Suitability
[Lotfollahi et al. 2020]	Yes	No	High	Limited
[Lin et al. 2022]	Yes	Limited	Very High	Limited
[Salau and Beyene 2024]	Yes	No	Moderate	Yes
[Najm et al. 2024]	Yes	No	Moderate	Yes
Proposed Method	No	Yes	Moderate	Yes

mental evaluation on the ISCX VPN-nonVPN dataset demonstrated that SFR-M supports accurate VPN mode detection and traffic-type classification using XGBoost, with consistent performance across different tokenization granularities. The results and comparative analysis with state-of-the-art approaches indicate that SFR-M addresses key methodological and operational limitations of existing methods by enabling early classification with bounded computational cost and compatibility with real-time deployment. Future work includes extending the evaluation to additional datasets, investigating alternative aggregation strategies, and exploring the integration of SFR-M with other learning paradigms and online classification settings.

Acknowledgments

The authors thank the National Council for Scientific and Technological Development (CNPq) and the Coordination for the Improvement of Higher Education Personnel (CAPES) for their support and funding. This research is part of the INCT of Communication Networks and Intelligent Internet of Things (ICoNIoT), funded by CNPq (grant 405940/2022-0) and CAPES (Funding Code 88887.954253/2024-00).

References

Aceto, G., Ciuonzo, D., Montieri, A., and Pescapé, A. (2018). Mobile encrypted traffic classification using deep learning. In *2018 Network traffic measurement and analysis conference (TMA)*, pages 1–8. IEEE.

Azab, A., Khasawneh, M., Alrabaee, S., Choo, K.-K. R., and Sarsour, M. (2024). Network traffic classification: Techniques, datasets, and challenges. *Digital Communications and Networks*, 10(3):676–692.

Dong, W., Yu, J., Lin, X., Gou, G., and Xiong, G. (2025). Deep learning and pre-training technology for encrypted traffic classification: A comprehensive review. *Neurocomputing*, 617:128444.

Eslami, E. and Hamouda, W. (2025). Network traffic classification using self-supervised learning and confident learning. *IEEE Open Journal of the Communications Society*.

Fernandes, M. (2026). SFR-M flow token dataset and embeddings (ISCX VPN-nonVPN). Mendeley Data, V1. doi: 10.17632/wc48j3hn7w.1.

Gil, G. D., Lashkari, A. H., Mamun, M., and Ghorbani, A. A. (2016). Characterization of encrypted and vpn traffic using time-related features. In *Proceedings of the 2nd international conference on information systems security and privacy (ICISSP 2016)*, pages 407–414. SciTePress Setúbal, Portugal.

Jorgensen, S., Holodnak, J., Dempsey, J., de Souza, K., Raghunath, A., Rivet, V., DeMoes, N., Alejos, A., and Wollaber, A. (2023). Extensible machine learning for encrypted network traffic application labeling via uncertainty quantification. *IEEE Transactions on Artificial Intelligence*, 5(1):420–433.

Lin, X., Xiong, G., Gou, G., Li, Z., Shi, J., and Yu, J. (2022). Et-bert: A contextualized datagram representation with pre-training transformers for encrypted traffic classification. In *Proceedings of the ACM Web Conference 2022*, pages 633–642.

Lotfollahi, M., Jafari Siavoshani, M., Shirali Hossein Zade, R., and Saberian, M. (2020). Deep packet: A novel approach for encrypted traffic classification using deep learning. *Soft Computing*, 24(3):1999–2012.

Najm, I. A., Saeed, A. H., Ahmad, B., Ahmed, S. R., Sekhar, R., Shah, P., and Veena, B. (2024). Enhanced network traffic classification with machine learning algorithms. In *Proceedings of the cognitive models and artificial intelligence conference*, pages 322–327.

Nascita, A., Aceto, G., Ciuonzo, D., Montieri, A., Persico, V., and Pescapé, A. (2024). A survey on explainable artificial intelligence for internet traffic classification and prediction, and intrusion detection. *IEEE Communications Surveys & Tutorials*.

Salau, A. O. and Beyene, M. M. (2024). Software defined networking based network traffic classification using machine learning techniques. *Scientific Reports*, 14(1):20060.