Fighting the Enemy Within: Traffic Semantics-Driven Framework Against Noisy Labels for Robust Traffic Classification

Mingwei Zhan, Ruijie Zhao, Xianwen Deng, Zhi Xue, Qi Li, *Senior Member, IEEE*, Zhuotao Liu, *Senior Member, IEEE*, Guang Cheng, *Member, IEEE*, and Ke Xu, *Fellow, IEEE*

***Abstract*—Network traffic classification is essential for network management and security. In recent years, deep learning (DL) algorithms have emerged as essential tools for classifying complex traffic. However, they rely heavily on high-quality labeled training data. In practice, traffic data is often noisy due to human error or inaccurate automated labeling, which could render classification unreliable and lead to severe consequences. Although some studies have alleviated the label noise issue in specific scenarios, they are difficult to generalize to general traffic classification tasks due to the inherent semantic complexity of traffic data. In this paper, we propose FLOWREFINER$^+$, a robust and general traffic classification framework against label noise. FLOWREFINER$^+$ consists of three core components: a traffic semantics-driven noise detector, a confidence-guided label correction mechanism, and a cross-granularity robust classifier. First, the noise detector utilizes traffic semantics extracted from a pre-trained encoder to identify mislabeled flows. Next, the confidence-guided label correction module fine-tunes a label predictor to correct noisy labels and construct refined flows. Finally, the cross-granularity robust classifier learns generalized patterns of both flow-level and packet-level, improving classification robustness against noisy labels. We evaluate our method on six traffic datasets with various classification scenarios across varying noise ratios and types. Experimental results demonstrate that FLOWREFINER$^+$ mitigates the impact of label noise and consistently outperforms state-of-the-art baselines by a large margin. The code is available at https://github.com/NSSL-SJTU/FlowRefinerPlus**

***Index Terms*—Traffic classification, label noise learning, robust network traffic analysis.**

Received 15 January 2026; revised 9 June 2026; accepted 31 August 2026; approved by IEEE TRANSACTIONS ON NETWORKING Editor H. Zhu. Date of publication 6 October 2026; date of current version 16 October 2026. This work was supported in part by China National Funds for Distinguished Young Scientists under Grant 62425201; in part by the National Key Research and Development Program of China under Grant 2023YFB3107100; in part by the National Natural Science Foundation of China under Grant 62502089, Grant 62172093, Grant 62132011, and Grant U22B2025; in part by the Basic Research Program of Jiangsu under Grant BK20251353; in part by the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China under Grant JYB2025XDXM114; in part by Beijing-Tianjin-Hebei Natural Science Foundation Cooperation Project under Grant 25JJJC0003; and in part by the Big Data Computing Center of Southeast University. The preliminary version [DOI: 10.52202/085713-4270] of this work has been accepted by the 39th Conference on Neural Information Processing Systems (NeurIPS 2025). *(Corresponding author: Ruijie Zhao.)*

Mingwei Zhan, Xianwen Deng, and Zhi Xue are with the School of Computer Science, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: mw.zhan@sjtu.edu.cn; 2594306528@sjtu.edu.cn; zxue@sjtu.edu.cn).

Ruijie Zhao is with the School of Cyber Science and Engineering, Southeast University, Nanjing, Jiangsu 211189, China, and also with the State Key Laboratory of Internet Architecture, Tsinghua University, Beijing 100084, China (e-mail: ruijiezhao@seu.edu.cn).

Qi Li and Zhuotao Liu are with the Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing 100084, China (e-mail: qli01@tsinghua.edu.cn; zhuotaoliu@tsinghua.edu.cn).

Guang Cheng is with the School of Cyber Science and Engineering, Southeast University, Nanjing, Jiangsu 211189, China (e-mail: chengguang@seu.edu.cn).

Ke Xu is with the Department of Computer Science and Technology, Tsinghua University, Beijing 100190, China, and also with the Zhongguancun Laboratory, Beijing 100086, China (e-mail: xuke@tsinghua.edu.cn).

Digital Object Identifier 10.1109/TON.2026.3739909

I. INTRODUCTION

NETWORK traffic classification is a fundamental task for the management, security, and optimization of modern networks. It enables critical capabilities such as identifying malicious behaviors [2], [3], [4], enforcing quality-of-service (QoS) policies [5], [6], [7], and monitoring application usage [8]. It can also support user-centric analysis such as profiling in social network applications by fine-grained inference of user actions [9], [10]. As network applications and protocols continue to evolve, traffic data have become increasingly complex, reflecting not only protocol interactions but also diverse user behaviors and service patterns [11]. These complexities make automated traffic classification both more important and more challenging [12], [13], [14].

In recent years, deep learning (DL) algorithms have emerged as powerful tools for traffic analysis for handling such complexity [15], [16], [17], [18], [19]. By leveraging large-scale labeled raw traffic data, DL-based models can automatically extract discriminative representations and achieve accurate classification performance. However, the success of DL-based methods heavily relies on the quality of labeled training data, and their performance can degrade significantly in the presence of label noise [20]. On the other hand, label noise, caused by incorrect, ambiguous, or inconsistent annotations, is especially a critical issue in traffic analysis tasks. Erroneous labels can hinder generalization in real-world scenarios and lead to severe consequences, such as failing to detect malicious traffic as suspicious or misclassifying latency-sensitive applications in QoS enforcement. Unfortunately, label noise is especially common in realistic traffic due to the complexity and variability of real-world environments and the flaws of automated labeling techniques [21], [22]. Though labels of lab-generated traffic could be refined through controlled environments and manual annotation processes, the label quality of realistic traffic is

2998-4157 © 2026 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: BEIJING INSTITUTE OF TECHNOLOGY. Downloaded on November 15,2026 at 03:10:37 UTC from IEEE Xplore. Restrictions apply.

TABLE I
THE COMPARISON WITH THE EXISTING METHODS

Methods	Against Label Noise	General Traffic Analysis	Work Basis
ET-BERT [30]	×	✓	N.A.
YaTC [31]	×	✓	N.A.
Co-teaching [27]	✓	×	N.A.
DivideMix [26]	✓	×	N.A.
RAPIER [24]	✓	×	Benign-Malicious Boundary
MCRe [25]	✓	×	Benign-Malicious Boundary
FLOWREFINER$^+$	✓	✓	**Traffic Semantics**

still challenging since the automated labeling is prone to errors [23].

Despite its significance, the issue of label quality is rarely noticed and discussed in existing DL-based traffic classification methods. As demonstrated in Table I, only a few studies [24], [25] explicitly target label noise in the context of malicious traffic, a specific subfield of traffic analysis focused primarily on intrusion detection. However, these approaches can hardly generalize to general traffic classification tasks, as some [24] are limited to binary classification, while others [25] rely on strong benign–malicious separability that does not hold in more diverse traffic scenarios. Specifically, these malicious-specific noise methods detect label errors via sample density or distance, implicitly assuming clear inter-class separation and tight intra-class clustering, which are conditions that do not hold in general traffic, where the same application can exhibit diverse behavioral patterns and different applications may share similar network-level characteristics. In addition, though various methods [26], [27], [28], [29] have been developed to address label noise in other fields (e.g., computer vision), they still struggle in traffic classification due to the inherent structural complexity and obfuscated semantics of traffic data. Thus, dealing with label noise in general traffic data remains a crucial and unresolved issue.

In this paper, we propose FLOWREFINER$^+$, a traffic semantics-driven framework against noisy labels for robust traffic classification. Our core insight is to utilize traffic semantics captured via self-supervised pre-training [30], [31], [32] as the label-independent foundation for noise handling that generalizes across diverse traffic scenarios. Since traffic semantics reflect the genuine nature of flows independent of their assigned labels [33], they provide a reliable and noise-immune basis for identifying label inconsistencies. Driven by traffic semantics, FLOWREFINER$^+$ effectively detects, corrects, and classifies traffic with noisy labels within a unified framework, achieving high performance without relying on high-quality labeled data, which is costly and difficult to obtain in increasingly complex network environments. It also introduces adaptive designs that automatically determine optimal boundaries for noise detection and correction, thereby minimizing the necessity for manual intervention. Moreover, by accurately identifying mislabeled flows, FLOWREFINER$^+$ provides valuable feedback to assist network administrators and annotators in improving traffic data quality. Specifically, FLOWREFINER$^+$ consists of three key components: a traffic semantics-driven noise detector, a confidence-guided label correction mechanism, and a cross-granularity robust classifier. First, we deploy a pre-trained traffic encoder to extract latent traffic semantic representations for noisy flow isolation. Then, a fine-grained clustering is employed, and adaptively defined majority labels within each cluster guide the division of flows into clean flows and noisy flows. Next, the confidence-guided label correction fine-tunes a traffic label predictor on clean flows to obtain the prediction confidences of noisy flows. A Gaussian Mixture Model is utilized to model the confidences and further direct the correction of label noise, thus reintroducing the corrected noisy flows into the clean flows to form the refined flows. Finally, our cross-granularity robust classifier integrates both flow-level and packet-level classification, enabling the model to capture generalized patterns rather than focusing on noise-driven isolated features. It ensures robustness and improves classification performance even in the presence of noisy labels.

In summary, our contributions are as follows:

- We introduce FLOWREFINER$^+$, a robust method for general traffic classification that effectively detects, corrects, and classifies traffic with noisy labels. To the best of our knowledge, this is the first work to address the label noise issues across diverse traffic classification tasks.
- We propose a traffic semantics-driven noise detector. It leverages the traffic semantics extracted by the pre-trained traffic encoder to detect outlier labels as noise, where the raw flows are divided into clean flows and noisy flows.
- We fine-tune a traffic label predictor based on isolated clean flows to perform confidence-guided label correction of detected noisy flows. The correction results can assist network administrators and annotators to improve traffic label quality.
- We design a cross-granularity robust classifier that integrates both flow-level and packet-level classification tasks to improve robustness, preventing the encoder from overfitting to noisy labels.
- We evaluate FLOWREFINER$^+$ on six real-world traffic datasets with different noise ratios and types. Results show that our method can achieve robust traffic classification against label noise and significantly outperforms the state-of-the-art baselines.

II. RELATED WORK

1) Traffic Classification Methods: In traffic analysis, traditional rule-based methods initially relied on fixed attributes like port numbers and protocols to classify traffic. However, with the rise of encryption protocols such as SSL and TLS, their effectiveness has gradually decreased because these protocols conceal key traffic characteristics [8], [34]. To overcome these limitations, machine learning (ML) techniques, including support vector machine (SVM) and random forest (RF), leveraged statistical features from traffic data to automate feature selection and improve classification accuracy [8], [35]. Although these ML-based methods marked a significant advancement, they heavily depend on handcrafted features, which limits adaptability across various traffic types [14].

Authorized licensed use limited to: BEIJING INSTITUTE OF TECHNOLOGY. Downloaded on November 15,2026 at 03:10:37 UTC from IEEE Xplore. Restrictions apply.

Deep learning (DL) methods have brought a shift towards automatic feature extraction from raw traffic data, with convolutional neural networks (CNNs), recurrent neural networks (RNNs), and Transformers being widely adopted for learning hierarchical representations [15], [17]. Recent arts have also explored pre-training techniques, with PERT and ET-BERT [30], [36] applying NLP-inspired BERT models [37] to encrypted traffic classification, and Flow-MAE [38], YaTC [31] using masked autoencoders (MAE) [32] to improve robustness in feature extraction under encryption. However, they are particularly prone to label noise, as their high capacity allows them to overfit incorrect labels [20], [39]. Some works [24], [25], [40], [41] have noticed the label noise issue in deep learning-based intrusion detection systems, and propose corresponding methods for malicious traffic. However, these methods exploit structural properties specific to intrusion detection and do not generalize to broader traffic scenarios: some studies [24], [42], [43] identify noisy samples by density differences between benign and attack flows; ULDC [41] assigns a single center per class and treats outlying samples as noisy, implicitly assuming tight intra-class clusters and clear inter-class separation; MCRe [25] similarly relies on the strong benign–malicious boundary to detect and relabel mislabeled samples. These approaches fundamentally assume clear inter-class separability, which does not hold in general traffic: the same application can exhibit diverse behavioral patterns (e.g., MAIL traffic spans bulk attachment transfers and lightweight message exchanges), while different applications may share similar network-level behaviors (e.g., FTP and P2P both involve large file transfers), making density- or distance-based noise detection unreliable.

2) Label Noise Learning Methods: Various methods have been developed to address label noise in other fields such as computer vision. A series of approaches like Generalized Cross-Entropy (GCE) [28] and Symmetric Cross-Entropy (SCE) [29] modify traditional loss functions to better handle noisy labels. GCE combines cross-entropy with mean absolute error to make learning more noise-resistant, while SCE balances penalization of misclassified samples. In addition, some works [27] [26] investigate filtering noisy samples that rely on the loss value distribution. On the other hand, data augmentation techniques have been widely used against label noise. As a classic robust augmentation, Mixup [44] applies interpolation between samples to generate robust training data. But it struggles when applied to the structured nature of encrypted traffic flows, where such interpolations may not produce meaningful results. Recent methods like Manifold DivideMix [45], which incorporate contrastive learning, offer improved robustness to severe noise. However, they rely on well-defined meaningful data augmentation, which is hardly performed in structured traffic data, limiting their ability to handle label noise within traffic flows.

In summary, existing works address traffic classification and label noise handling in isolation, leaving the problem of noisy labels in general traffic unresolved. As summarized in Table I, two complementary gaps remain open. On one side, state-of-the-art traffic classifiers (e.g., ET-BERT, YaTC) achieve strong performance under clean labels but are not designed to handle label noise, leaving them vulnerable as noise levels increase. On the other side, general label noise learning methods (e.g., Co-teaching, DivideMix) fail on traffic data due to its structural complexity, and traffic-specific noise methods are confined to intrusion detection with domain-specific separability assumptions that do not transfer to general traffic scenarios. No existing method simultaneously addresses label noise and supports general traffic classification. To fill this gap, we propose FLOWREFINER$^+$, a robust framework specifically designed to detect, correct, and classify traffic with noisy labels, offering a comprehensive solution for label noise in general traffic classification. FLOWREFINER$^+$ is grounded in traffic semantics rather than domain-specific assumptions, enabling it to bridge both gaps identified in Table I and generalize across diverse traffic classification scenarios.

III. METHODOLOGY

We introduce FLOWREFINER$^+$, a robust method for general traffic classification that effectively detects, corrects, and classifies traffic with noisy labels. Figure 1 shows the overview of FLOWREFINER$^+$. Our framework starts with traffic semantics extracted by the advanced pre-trained encoder, which can reflect the similarities and diversity of flow characteristics, providing a basis for detecting label-noisy flows (detailed in Sec. III-A). After that, a predictor is fine-tuned via detected clean flows to correct noisy labels and construct refined flows with low-level label noise (detailed in Sec. III-B). Finally, a traffic classifier with a cross-granularity structure is designed to perform robust general traffic classification against label noise (detailed in Sec. III-C).

A. Traffic Semantics-Driven Noise Detector

Traffic semantics-driven noise detector aims to identify and isolate noisy labels in traffic data effectively. It achieves effective noisy flow detection by leveraging the traffic semantics extracted from the flows and combining their similarity with label distribution. Traffic semantics are defined as the intrinsic behavioral and functional characteristics of network flows encoded in raw traffic content, which remain invariant to label assignments and can be captured via self-supervised pre-training without any label supervision.

1) Traffic Semantics Extraction: We first build a traffic semantics-oriented encoder via the pre-training paradigm to extract the latent representation of traffic data. Benefiting from the learned semantics from the unlabeled traffic data in pre-training, the pre-trained traffic encoder can effectively extract traffic semantics [30], while remaining immune to interference from potential noisy labels.

We adopt the Masked Autoencoder (MAE) [32] style pre-training paradigm to enable the encoder to capture the traffic semantics by randomly masking input and reconstructing the missing portions. Figure 2 depicts the process of building the traffic semantics-oriented encoder via MAE. Initially, the network traffic data captured from Pcap files undergoes pre-processing to extract and resize the flow content, ensuring the data is in a suitable format for model training. The processed traffic data is then intentionally masked, and input into the

Authorized licensed use limited to: BEIJING INSTITUTE OF TECHNOLOGY. Downloaded on November 15,2026 at 03:10:37 UTC from IEEE Xplore. Restrictions apply.

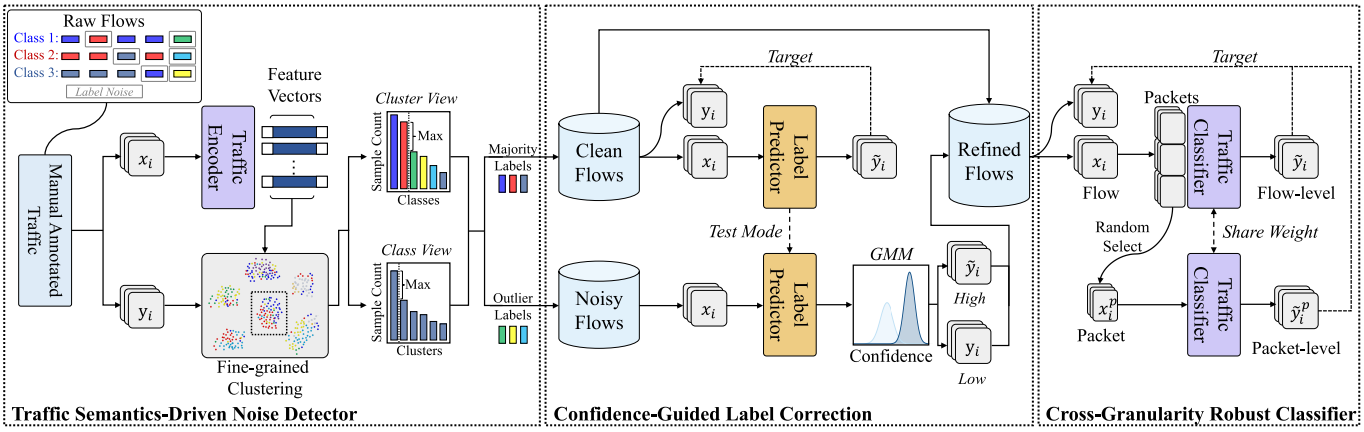


Fig. 1. The overview of FLOWREFINER+. It consists of three key components: a traffic semantics-driven noise detector, a confidence-guided label correction mechanism, and a cross-granularity robust classifier. FLOWREFINER+ effectively detects, corrects, and classifies traffic with noisy labels in a uniform framework, achieving high performance without relying on high-quality labeled data, which are costly and difficult to obtain in increasingly complex network environments.

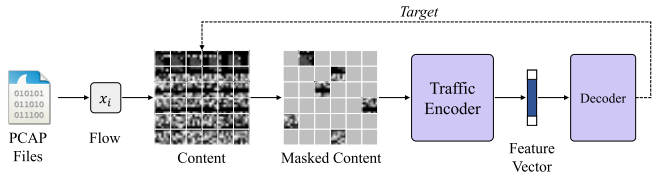


Fig. 2. The pre-training paradigm of masked autoencoder for building traffic semantics-oriented encoder.

traffic encoder, which employs the principles of an MAE by aiming to recover the original data from its corrupted form. The encoder learns to extract robust feature representations by focusing on the structure left intact by the masking process. Finally, the encoded features are passed through a decoder that attempts to reconstruct the original input, thereby enabling the model to learn critical data characteristics effectively and allowing the encoder to extract effective traffic semantics.

In addition, the encoder deploys the Transformer architecture as the backbone for feature extraction. Formally, let $X = \{x_1, x_2, \ldots, x_N\}$ represent the dataset, where x_i is an individual flow sample. The pre-trained encoder, denoted as $f_{\text{enc}}(\cdot)$, processes each sample x_i and produces a d-dimensional feature vector:

$$z_i = f_{\text{enc}}(x_i), \quad z_i \in \mathbb{R}^d. \tag{1}$$

2) Noise Detection Based on Traffic Semantics: The distances between traffic semantics embeddings can effectively reflect the similarities in flow content and function. These similarities have the potential to highlight instances of label noise, as similar flows with discrete labels are likely candidates for noisy labels. However, as we visualized via t-SNE in Figure 3, since the traffic semantics have not yet been aligned with the correct labels, there are two main challenges in detecting label noise via similarities:

- **Similar Semantics from Diverse Classes:** Traffic data from different classes may appear semantically similar, leading to groups of mixed-category flows.
- **Diverse Semantics from the Same Class:** Traffic data in the same class may not exhibit consistent behaviors, forming diverse compact clusters in the latent space.

Therefore, we designed a fine-grained clustering and defined majority labels for each cluster to address the above issues and detect noisy labels. We introduce adaptive, data-driven designs to avoid the reliance on manual tuning parameters, such as the pre-defined cluster number and fixed majority label count, which allows the model to automatically determine the optimal semantic structure and trustworthy labels based on intrinsic semantic distributions, ensuring the adaptability and generalizability of noise detection.

To capture the different behaviors inside the same class, we perform fine-grained clustering, which models the semantic similarity between traffic flows with a more fine-grained level than categories. Specifically, we perform the Agglomerative Clustering algorithm on extracted feature vectors $\{z_1, z_2, \ldots, z_N\}$ without assuming a fixed number of clusters. Unlike K-means which enforces spherical clusters and requires a pre-defined cluster count, the algorithm automatically determines the optimal number of clusters based on the local density of the traffic semantics. Two clusters C_i and C_j are merged only if their linkage distance $d(C_i, C_j)$ is less than δ. This allows the formation of natural clusters $\{C_1, C_2, \ldots, C_K\}$ where K is adaptive to the data distribution. Subsequently, flows within the same cluster share the most similar semantics, and the presence of inconsistent labels within each cluster is regarded as potential noise.

Considering the potentially similar activity of flows from different categories, we define the majority labels for each cluster to represent the inside overlapped true classes. An intuitive approach is to consider the top m most frequent labels in the cluster as the majority. However, this fixed definition still lacks the adaptability to dynamically delineate the boundary between reliable majority labels and label noise. On the other hand, classes with inherently smaller sample counts (e.g., minority classes) can hardly ever become the top m label in a cluster through simple majority voting.

Authorized licensed use limited to: BEIJING INSTITUTE OF TECHNOLOGY. Downloaded on November 15,2026 at 03:10:37 UTC from IEEE Xplore. Restrictions apply.

Thus, we introduce an adaptive mechanism to identify trustworthy labels from both the cluster and class perspectives. We define a max-gap function to identify the dominant set in a discrete distribution by locating the maximum gap between adjacent sorted sample count values. Formally, let $\mathcal{C} = \{c_1, c_2, \ldots, c_n\}$ denote the sample counts of labels within a cluster, and let $c_{(1)} \leq c_{(2)} \leq \cdots \leq c_{(n)}$ be the sorted sequence in ascending order. We compute the adjacent gaps as $g_i = c_{(i+1)} - c_{(i)}$, for $i = 1, \ldots, n-1$, and define the max-gap index as $i^* = \arg\max_i g_i$, with the corresponding cut point $\tau = c_{(i^*)}$. Specifically, this maximum gap induces a data-driven cut point that partitions the labels into two groups according to their sample counts: labels with counts greater than τ are regarded as the majority, while those with counts less than or equal to τ are treated as outliers. This formulation captures the most significant structural separation in the discrete distribution without introducing additional hyperparameters. Based on this non-parametric technique, we analyze the structural integrity of the fine-grained clusters through two complementary views:

- **Cluster View:** This perspective is designed to identify legitimate semantic overlap while excluding statistical noise. For each cluster C_k, we calculate the raw sample count $N(y|C_k)$ of each label y. We apply the max-gap function to the sorted counts to identify the majority labels L_k^{cluster}. This dynamically determines which classes are truly dominant within the specific semantic cluster, effectively distinguishing genuine semantic overlap from sporadic label noise.
- **Class View:** This perspective is designed to ensure that minority classes are not unfairly discarded. For each class label y, we examine the distribution of its samples across all generated clusters $\{C_k\}$. We calculate the raw sample count $N(C_k|y)$ of each cluster C_k within class y. We apply the max-gap function to the sorted counts $N(C_k|y)$ to identify the set of clusters that primarily represent the semantics of class y. Then, the class y is also considered as the majority label of these identified primary clusters in Class View, defined as L_y^{class}. This dynamically confirms which clusters are the primary semantic representations for class y, preserving small but valid clusters that belong to minority classes.

Finally, we detect and divide the raw flows into a clean flow set and a noisy flow set by combining both views. A sample is considered clean if its label is dominant in its cluster or if its cluster is identified as a main representation of its label:

$$\mathcal{D}_{\text{clean}} = \{(x_i, y_i) \mid x_i \in C_k, y_i \in \ L_k^{\text{cluster}} \cup L_y^{\text{class}}\}, \quad (2)$$

$$\mathcal{D}_{\text{noisy}} = \{(x_i, y_i) \mid x_i \in C_k, y_i \notin L_k^{\text{cluster}} \cup L_y^{\text{class}}\}, \quad (3)$$

where y_i is the raw label for sample x_i, and L_k^{cluster} and L_y^{class} denote the majority labels of cluster C_k and class y, respectively. The clean flows $\mathcal{D}_{\text{clean}}$ can be used to support further training, while the noisy flows $\mathcal{D}_{\text{noisy}}$ can be flagged and reported to the manager or annotator for review.

B. Confidence-Guided Label Correction

Through the semantic-driven noise detector, most noisy flows are identified and isolated from the clean flows according to their label-independent traffic semantics. Furthermore, this clean flow set can provide valid supervision to further correct the label distribution of the noisy flows and expand the available training data. For this purpose, we design the confidence-guided label correction module to correct flow labels in the noisy flow set based on prediction confidence.

1) Traffic Label Predictor Fine-Tuning: In this stage, we aim to fine-tune the latent representations of flows to align with the labels of selected clean flows $\mathcal{D}_{\text{clean}}$, thereby obtaining a confidence-aware traffic label predictor that can assess the category of selected noisy flows $\mathcal{D}_{\text{noisy}}$ based on confidence scores. Formally, let $X_{\text{clean}} = \{x_1, x_2, \ldots, x_{N_c}\}$ be the set of clean flow samples in $\mathcal{D}_{\text{clean}}$, and $y_{\text{clean}} = \{y_1, y_2, \ldots, y_{N_c}\}$ be their corresponding labels. Then, we leverage the supervised learning paradigm to fine-tune our label predictor $f_{\text{prd}}(x)$, which consists of the pre-trained encoder and an added classification head, aiming to minimize the cross-entropy loss:

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{N_c} y_i \log(f_{\text{prd}}(x_i)), \quad (4)$$

Our well-trained label predictor provides confidence scores, $p_i = \max(f_{\text{prd}}(x_i))$, which represent the predictor's certainty regarding the predicted class. These confidence scores serve as the basis for further label correction of noisy flows.

2) Adaptive Label Correction: To refine the available flows with label noise and avoid losing valuable diversified information, we correct the original labels of noisy flows based on the predicted confidence scores. Specifically, for each noisy flow sample and its label $(x_i, y_i) \in \mathcal{D}_{\text{noisy}}$, the predictor outputs a predicted label $\hat{y}_i$ along with its confidence score $p_i = \max(f_{\text{prd}}(x_i))$, which is the highest softmax probability from the label predictor. Based on these confidence scores, we refine the labels of noisy flows into two categories:

1) **High-Confidence Flow Labels**: These samples have high confidence in their predicted labels and are thus assumed to share a similar distribution with the clean set. Thus, we correct their labels with the predicted labels $\hat{y}_i$ and merge them into the refined flows.
2) **Low-Confidence Flow Labels**: These samples have low confidence and likely lie outside the semantic distribution of the clean set. Since they are difficult for the classifier to categorize, we retain their original labels and introduce them to the refined flows to maintain diversity and expand the semantic distribution of samples.

These two categories of predicted labels for noisy flows could be directly divided by setting fixed thresholds of confidence. However, fixed thresholds fail to capture the complex dynamics of training confidence, which would evolve across different scenarios. Furthermore, a static boundary may not optimally separate reliable predictions from unreliable ones as the model converges. To address this, we propose a distribution-aware adaptive correction strategy using a Gaussian Mixture Model (GMM) to model the latent structure of confidence scores. This GMM is employed to dynamically

Authorized licensed use limited to: BEIJING INSTITUTE OF TECHNOLOGY. Downloaded on November 15,2026 at 03:10:37 UTC from IEEE Xplore. Restrictions apply.

delineate the boundary that separates the high-confidence flow labels and the low-confidence flow labels defined above. We fit a two-component GMM to the set of confidence scores $\mathcal{P} = \{p_i\}_{i=1}^{N}$. The probability density function is defined as:

$$p(p_i) = \sum_{k=1}^{2} \pi_k \mathcal{N}(p_i \mid \mu_k, \sigma_k^2), \tag{5}$$

where π_k, μ_k, and σ_k^2 represent the mixture weight, mean, and variance of the k-th component, respectively. We estimate these parameters via the Expectation-Maximization (EM) algorithm.

Based on the estimated means μ_k, we order the components such that $\mu_{low} < \mu_{high}$. The GMM then automatically assigns each sample x_i to either the lower-mean component, which is denoted as $\mathcal{R}_{low}$, or the higher-mean component, which is denoted as $\mathcal{R}_{high}$. This adaptive assignment directly realizes the two refinement categories: samples assigned to the higher-mean component $\mathcal{R}_{high}$ are categorized as high-confidence flow labels and their labels are corrected to the predicted label $\hat{y}_i$; conversely, samples assigned to the lower-mean component $\mathcal{R}_{low}$ are categorized as low-confidence flow labels, and their original labels y_i are retained.

Combined with the clean flows, the obtained refined flows $\mathcal{D}_{\text{refined}}$ preserve a more representative semantic distribution. The final refined flows are formulated as:

$$\begin{aligned}\mathcal{D}_{\text{refined}} = \mathcal{D}_{\text{clean}} &\cup \{(x_i, \hat{y}_i) \mid x_i \in \mathcal{R}_{high}\} \\ &\cup \{(x_i, y_i) \mid x_i \in \mathcal{R}_{low}\}.\end{aligned} \tag{6}$$

It is worth noting that the confidence-guided correction mechanism exhibits inherent resilience to residual noise in $\mathcal{D}_{\text{clean}}$. Since $\mathcal{D}_{\text{clean}}$ is predominantly composed of correctly labeled samples, the fine-tuned predictor learns a decision boundary well-aligned with the majority clean distribution. Any predictions influenced by the small residual noisy minority are contradicted by this dominant evidence and thus receive lower confidence scores. The GMM boundary then filters these unreliable predictions from active correction, preventing residual noise in $\mathcal{D}_{\text{clean}}$ from cascading into $\mathcal{D}_{\text{refined}}$.

C. Cross-Granularity Robust Classifier

Based on the refined flows with low-level label noise, we perform traffic encoder fine-tuning for robust traffic classification. Due to strong fitting capabilities, deep learning models are prone to memorize mislabeled samples themselves, rather than capturing generalizable patterns [46], [47]. To avoid the encoder memorizing specific flows with false labels, we propose a cross-granularity robust classifier that integrates both flow-level and packet-level traffic classification tasks. Thus, our classifier can capture generalized patterns that are valid on both tasks, rather than focusing on isolated invalid features that are misdirected by noisy labels.

The robust traffic classifier contains two parallel shared weight encoders and classification heads, which could be formulaically treated as a single encoder and denoted as f_{enc} and f_{head}. The structure of the shared-weight classifier that can handle both flow and packet samples is shown in Figure 4.

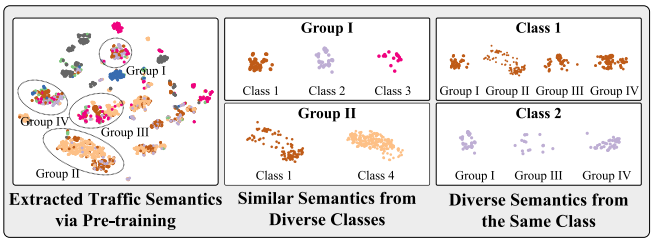


Fig. 3. The t-SNE visualization of the traffic semantics.

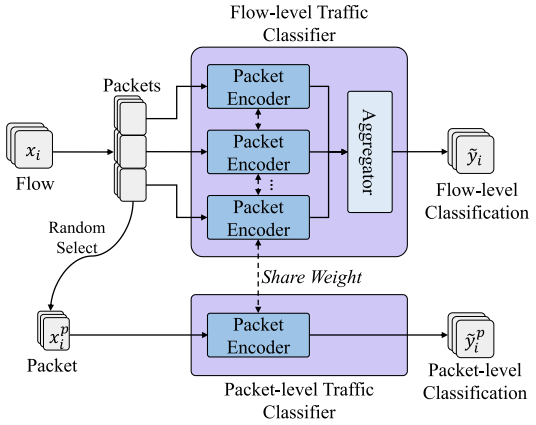


Fig. 4. The structure detail of the cross-granularity robust classifier. The flow-level classifier utilizes multiple parallel, share-weight packet encoders followed by an aggregator. The packet-level classifier uses the same shared packet encoder to classify a single packet (x_i^p) randomly selected from the flow (x_i). This dual constraint enhances robustness against label noise.

The encoder is also based on the Transformer and loads the parameters pre-trained by MAE. It first processes each flow x_i by dividing it into sequential packets then selects one packet x_i^p randomly to serve as the other input sample:

$$x_i^p = RandomSelect(\{pac_i^1, pac_i^2, \ldots \mid x_i\}). \tag{7}$$

Next, the flow sample x_i is inputted to the encoder and linear heads to generate $\hat{y}_i$, a flow-level prediction of the label. Following the above process, the flow classification task is performed with the flow sample x_i and label y_i from the refined flows by minimizing the cross-entropy loss. Then, the packet classification task is trained subsequently during each epoch. Similarly, the randomly selected packet sample is inputted to the encoder and classification head, and obtains the packet level prediction $\hat{y}_i^p$. By associating packet sample x_i^p with the flow label y_i via the cross-entropy loss, the classifier performs packet classification and learns traffic patterns on another granularity.

By the cross-granularity classification tasks, the classifier can utilize both global (flow-level) and local (packet-level) features within the refined flows, providing more varied input against label noise. Specifically, by introducing randomness in the packet selection, the classifier obtains different sample variants in each epoch, thus avoiding the memorization of specific flows with false labels. In addition, the trained cross-granularity robust classifier can serve for both flow-level or packet-level traffic classification under the label noise

Authorized licensed use limited to: BEIJING INSTITUTE OF TECHNOLOGY. Downloaded on November 15,2026 at 03:10:37 UTC from IEEE Xplore. Restrictions apply.

TABLE II
DATASET STATISTICS WITH PER-CLASS SAMPLE DISTRIBUTION

Dataset	Year	#Classes	Total	Min	Max	Std
ISCXVPN	2016	7	2,844	138	864	283.5
CrossPlatform	2020	30	6,801	31	790	207.1
USTC-TFC	2016	20	2,397	50	200	58.3
Malware	2023	10	3,678	128	556	136.7
CICIDS2017	2017	7	261,215	478	158,715	53,881.9
RealNetwork	2025	7	3,500	500	500	0.0

according to the practical needs, bringing not only robustness but also flexibility.

IV. EXPERIMENTS

A. Experiment Settings

1) Datasets: We conduct our experiments on six traffic datasets: ISCXVPN [48], CrossPlatform [49], USTC-TFC [17], Malware [50], CICIDS2017 [51], and RealNetwork [52], covering different traffic scenarios. Each dataset sample in our experiments corresponds to a network flow, which is obtained via session-aware splitting according to the standard 5-tuple (source IP, destination IP, source port, destination port, and protocol). In FLOWREFINER+, we process each flow into a formatted matrix via the MFR algorithm [31]. To comprehensively evaluate the robustness of our methods against label noise, we generate noisy datasets for each scenario with different noise ratios ($5\%, 10\%, 20\%, 40\%, 60\%$). The details of these datasets are shown as follows:

- **ISCXVPN** [48] contains 7 categories of encrypted traffic, collected by the Canadian Institute for Cybersecurity via OpenVPN. The dataset consists of 2,275 training samples and 569 test samples.
- **CrossPlatform** [49] includes encrypted traffic data from various platforms. In our experiments, we use iOS application traffic in the China region, comprising 30 different categories. The dataset consists of 5,429 training samples and 1,372 test samples.
- **USTC-TFC** [17] is a traffic dataset with 10 categories of benign application traffic and 10 categories of malware traffic. The dataset consists of 1,914 training samples and 483 test samples.
- **Malware** [50] is a recently published dataset featuring traffic from 10 malware families. The dataset consists of 2,938 training samples and 740 test samples.
- **CICIDS2017** [51] used in Section IV-C3 contains 7 real-world benign and attack traffic classes, which are proven to have existing real-world label noise by [21] and [22]. The train set contains 208,970 samples, while the 52,245 test samples are relabeled following [21] for evaluation.
- **RealNetwork** [52] is a newly released (2025) real-world dataset used in Section IV-E1, comprising 7 categories of encrypted application traffic. The dataset contains 2,800 training and 700 test samples.

The key statistics of these datasets, i.e., per-class sample distribution statistics (total size, Min, Max, and Std across classes), are summarized in Table II.

2) Baselines: To evaluate the performance of FLOWREFINER+, we compare it with six state-of-the-art traffic classification methods. These include two traditional ML- and DL-based methods, Appscanner [8] and FS-Net [15]; three advanced pre-training methods, ET-BERT [30], MAE [38], and YaTC [31]; and the malicious traffic label noise method, MCRe [25]:

- **Appscanner** [8]: A traditional machine learning tool that classifies traffic using handcrafted flow-based features, often less resilient to noisy or encrypted data.
- **FS-Net** [15]: A flow-based model incorporating both classification and reconstruction tasks, enhancing its robustness to noise and improving semantic feature extraction.
- **ET-BERT** [30]: A BERT-like Transformer pre-trained model, leveraging self-supervised learning to capture flow-level semantic patterns, offering strong resilience to label noise.
- **MAE** [38]: A self-supervised traffic classifier based on Masked Autoencoder (MAE), masking parts of traffic flows and reconstructing them, which makes it robust in noisy conditions.
- **YaTC** [31]: A hybrid traffic classification model combining supervised and self-supervised learning, balancing noise handling and effective traffic feature extraction.
- **MCRe** [25]: A state-of-the-art traffic analysis method focuses on malicious traffic label noise learning.
- **FlowRefiner** [1]: The variant of our approach without adaptive designs. Specifically, it utilizes K-means clustering with a fixed cluster count and applies a static majority label count, i.e., top m, for noise detection, and employs fixed confidence thresholds for label correction.

In addition, we introduce the following label noise learning baselines, all of which use the same encoder structure for fair comparison:

- **CE**: A standard cross-entropy loss used in classification tasks, which can be sensitive to label noise as it directly penalizes incorrect predictions.
- **LSR** [53]: Label Smoothing Regularization reduces model overconfidence by distributing probability mass to incorrect labels, making the model more robust to noise.
- **Mixup** [44]: A data augmentation technique that generates synthetic samples by interpolating between two samples and their labels, helping the model generalize better under noisy conditions.
- **GCE** [28]: Generalized Cross-Entropy blends the advantages of MAE and CE to address label noise, making it more robust by adjusting the loss function dynamically.
- **SCE** [29]: Symmetric Cross-Entropy introduces a correction term to balance the standard CE loss, aiming to better handle noisy labels by mitigating over-penalization of incorrect predictions.
- **Co-teaching** [27]: Trains two networks simultaneously and lets them teach each other by selecting small-loss samples, assuming that clean samples have lower loss values. This helps filter out noisy labels during training.

Authorized licensed use limited to: BEIJING INSTITUTE OF TECHNOLOGY. Downloaded on November 15,2026 at 03:10:37 UTC from IEEE Xplore. Restrictions apply.

TABLE III
PERFORMANCE COMPARISON WITH TRAFFIC CLASSIFICATION BASELINES UNDER DIFFERENT NOISE RATIOS

Dataset	Noise	Appscanner		FS-Net		ET-BERT		MAE		YaTC		MCRe		FlowRefiner		Ours	
		Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
ISCXVPN	5%	81.49	82.12	87.97	88.05	86.12	86.11	91.04	91.08	92.44	92.35	83.01	83.63	93.67	93.34	**93.85**	**93.61**
	10%	79.22	79.63	85.47	85.49	85.24	85.18	89.81	89.80	89.28	89.03	81.05	81.63	91.04	90.48	**92.27**	**92.04**
	20%	77.27	77.95	82.57	82.57	78.91	79.07	79.26	79.29	82.60	82.49	79.88	80.64	90.86	90.15	**90.86**	**90.32**
	40%	73.70	73.85	74.37	75.08	59.92	61.78	63.62	64.21	66.43	67.15	76.56	76.62	85.06	84.19	**86.47**	**86.24**
	60%	57.14	57.89	56.40	58.24	34.79	36.21	42.00	43.19	43.23	45.37	69.92	70.55	73.29	72.90	**75.40**	**75.40**
CrossPlatform	5%	68.00	67.34	61.88	61.04	98.97	98.99	97.67	97.69	98.91	98.91	67.66	67.59	99.71	99.71	**99.93**	**99.93**
	10%	66.40	65.92	59.61	59.62	98.39	98.41	95.19	95.24	98.32	98.32	63.67	63.45	99.56	99.55	**99.85**	**99.85**
	20%	62.53	62.39	57.89	58.07	94.46	94.49	90.67	90.75	93.51	93.58	61.48	61.14	98.76	98.75	**99.13**	**99.11**
	40%	55.32	55.51	50.54	50.49	78.42	79.30	73.98	75.12	80.47	80.92	53.91	52.44	95.34	95.20	**98.32**	**98.31**
	60%	43.29	44.76	39.76	41.21	56.26	59.04	48.54	50.62	57.22	58.94	43.44	40.76	84.33	84.29	**95.04**	**94.87**
USTC-TFC	5%	62.40	57.64	87.50	87.61	95.03	94.98	94.82	94.82	94.62	94.58	93.75	94.91	96.07	96.02	**96.69**	**96.68**
	10%	63.94	60.83	86.95	86.89	93.79	93.77	94.00	93.95	93.79	93.99	93.36	94.71	95.65	95.60	**95.86**	**95.79**
	20%	62.65	59.95	84.76	84.70	92.33	92.34	86.13	86.46	89.86	89.75	91.02	92.91	94.20	93.93	**95.45**	**95.41**
	40%	56.26	50.56	77.89	77.52	81.98	81.69	74.12	73.82	80.54	80.11	89.06	91.41	91.72	91.69	**92.55**	**92.40**
	60%	51.73	50.98	70.62	70.22	58.59	58.09	47.62	48.11	56.31	56.57	75.78	75.02	78.88	78.71	**87.58**	**87.47**
Malware	5%	78.16	77.86	77.81	77.90	85.95	86.01	92.57	92.59	93.11	93.11	65.62	66.77	93.38	93.33	**96.22**	**96.22**
	10%	76.63	76.36	76.71	76.67	84.46	84.34	90.54	90.56	90.95	90.94	63.28	64.85	91.35	91.28	**95.27**	**95.27**
	20%	76.43	76.14	75.78	75.83	77.83	78.03	81.49	81.43	83.51	83.50	62.50	64.19	88.24	88.03	**92.03**	**91.97**
	40%	71.26	70.67	71.40	71.08	61.35	61.49	65.27	65.39	67.97	68.49	57.23	58.83	83.51	83.33	**85.14**	**84.87**
	60%	63.02	62.71	57.81	57.13	42.43	43.83	46.08	47.16	46.35	46.45	55.47	56.21	73.11	72.61	**74.86**	**74.81**

- **DivideMix** [26]: Uses a Gaussian mixture model to divide samples into clean and noisy sets based on loss values, then applies semi-supervised learning to train the model using both labeled and unlabeled data.

3) Implementation Details: We use Transformer [54] as the backbone of FLOWREFINER$^+$, including the traffic encoder, label predictor, and traffic classifier of our framework. It contains an embedding module and 4 Transformer layers with 192 dimensions and 16 attention heads. In the training stage, we set the batch size as 64, the epochs as 15, and the learning rate as 2×10^{-3} with the AdamW optimizer [55]. All experiments are implemented in four NVIDIA GeForce RTX3090 GPUs with PyTorch 1.9.0.

B. Comparison With Baselines

1) Traffic Classification Baselines: The performance of our method and other traffic analysis methods on the four traffic datasets under different noise ratios is shown in Table III. Overall, the performance of all baselines degrades significantly as the label noise rate increases, which highlights the substantial impact of label noise on DL-based methods. Traditional models such as Appscanner and FS-Net are particularly vulnerable, showing rapid performance drops even under moderate noise. Although advanced pre-training methods such as ET-BERT, MAE, and YaTC perform well under low-noise settings such as 5% and 10%, their robustness deteriorates noticeably as the noise level increases. MCRe, which is designed for handling label noise in malicious traffic, shows strong robustness on the USTC-TFC that includes multiple benign and malicious traffic categories. However, its performance degrades significantly on other general classification tasks, such as VPN or mobile app traffic, as well as on the Malware dataset, which contains only recent malware families, due to its reliance on distinct benign-malicious separability.

FlowRefiner, the variant of our method without adaptive designs, while generally superior to other baselines, still suffers from reduced performance at high noise levels due to the limitations of its reliance on fixed clustering and confidence thresholds. We can observe that FLOWREFINER$^+$ consistently outperforms all baselines across datasets and noise levels. Compared to the variant, FLOWREFINER$^+$'s adaptive design proves critical, consistently yielding superior performance and substantially increased robustness across high noise ratios. Our traffic semantics-driven noise detector and confidence-guided label correction can validly isolate noisy labels and refine the raw flows. Then, the cross-granularity robust classifier can capture meaningful hierarchical semantics and avoid remembering specific noisy flows. As a result, our method consistently achieves accuracy and F1 scores exceeding 70% even at a 60% label noise ratio, where the majority of training samples are incorrectly labeled. It can be concluded that our method provides a robust traffic classification framework against label noise to achieve superior performance on various traffic datasets under noisy label conditions.

2) Label Noise Learning Baselines: We further reproduce six general label noise learning methods from other fields on traffic data with consistent encoders for fair comparison, as illustrated in Table IV. At noise ratios above 5%, LSR and SCE provide limited overall improvements over CE (i.e., the baseline), while Mixup and GCE can handle label noise relatively effectively due to the interpolating and adaptive loss design. More advanced methods, such as Co-teaching and DivideMix, achieve competitive performance. However, Co-teaching relies on a predefined forgetting-rate schedule to retain small-loss samples as likely clean data, which may not adapt well to different noise conditions. DivideMix, on the other hand, uses a Gaussian mixture model based on loss value distribution to distinguish noise, but it fails when clean or noisy samples are too few to support reliable modeling. In contrast, our method detects noisy flows by modeling traffic semantics rather than relying on loss value distribution, making it more aligned with the nature of traffic data. Additionally, the GMM

Authorized licensed use limited to: BEIJING INSTITUTE OF TECHNOLOGY. Downloaded on November 15,2026 at 03:10:37 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
PERFORMANCE COMPARISON WITH LABEL NOISE BASELINES UNDER DIFFERENT NOISE RATIOS

Dataset	Noise	CE		LSR		Mixup		GCE		SCE		Co-teaching		DivideMix		Ours	
		Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
ISCXVPN	5%	78.56	78.38	78.56	78.22	79.61	79.33	76.80	75.39	77.33	77.04	81.90	78.24	81.37	81.93	**93.85**	**93.61**
	10%	75.57	75.19	75.92	76.29	78.56	77.79	75.57	74.59	73.99	73.80	85.41	84.14	83.48	83.79	**92.27**	**92.04**
	20%	74.34	74.65	75.22	75.65	79.26	79.10	73.11	71.80	73.46	73.44	88.24	87.65	80.84	81.24	**90.86**	**90.32**
	40%	55.89	56.42	56.94	58.03	67.84	66.46	66.61	65.08	60.46	59.49	77.33	76.69	76.44	76.61	**86.47**	**86.24**
	60%	41.12	39.95	42.00	41.43	50.09	48.31	51.14	49.05	40.60	40.68	54.83	55.89	67.14	68.88	**75.40**	**75.40**
CrossPlatform	5%	93.37	93.29	94.75	94.56	90.96	90.83	94.02	93.29	94.17	94.04	86.01	82.29	83.67	84.48	**99.93**	**99.93**
	10%	92.20	92.20	90.38	90.45	89.14	89.05	93.59	92.81	93.44	93.41	90.08	87.83	86.29	86.72	**99.85**	**99.85**
	20%	85.71	85.75	87.53	87.45	86.66	86.58	92.71	92.19	87.10	87.12	93.00	92.04	86.15	86.63	**99.13**	**99.11**
	40%	70.26	70.86	70.41	70.81	78.28	78.48	90.67	90.19	76.68	76.82	84.03	83.93	80.68	81.62	**98.32**	**98.31**
	60%	46.64	48.07	46.43	48.41	62.54	61.59	80.39	79.69	60.13	60.33	61.81	62.61	75.15	75.34	**95.04**	**94.87**
USTC-TFC	5%	86.75	86.67	88.82	88.73	91.93	91.89	83.22	82.28	86.75	86.74	86.54	86.24	89.23	89.16	**96.69**	**96.68**
	10%	84.06	83.58	85.30	84.93	90.48	90.29	81.78	80.68	85.30	85.10	89.02	88.16	87.16	87.77	**95.86**	**95.79**
	20%	80.95	80.56	79.08	78.36	87.58	87.40	79.30	78.42	85.30	85.04	93.58	93.50	91.10	91.11	**95.45**	**95.41**
	40%	66.25	66.05	67.91	67.52	74.74	73.77	78.26	77.00	74.12	73.84	81.57	81.21	85.30	84.84	**92.55**	**92.40**
	60%	51.14	50.76	55.49	55.27	65.01	64.12	70.60	68.01	57.35	56.54	56.73	57.22	78.47	78.43	**87.58**	**87.47**
Malware	5%	78.38	78.60	87.84	87.91	89.73	89.77	86.22	86.22	88.92	88.87	86.76	86.85	85.81	85.91	**96.22**	**96.22**
	10%	78.11	78.01	83.78	83.74	88.11	88.12	81.89	81.76	83.78	83.90	88.11	88.20	85.95	86.15	**95.27**	**95.27**
	20%	78.78	78.78	82.16	81.93	82.43	82.22	80.00	80.15	78.10	77.93	84.56	84.20	83.10	83.47	**92.03**	**91.97**
	40%	65.00	64.93	64.86	65.58	73.38	73.33	75.14	74.59	68.11	67.79	80.95	80.80	82.30	82.41	**85.14**	**84.87**
	60%	52.43	52.94	54.73	55.03	62.30	62.63	64.05	64.46	56.22	56.61	59.86	60.29	70.81	70.34	**74.86**	**74.81**

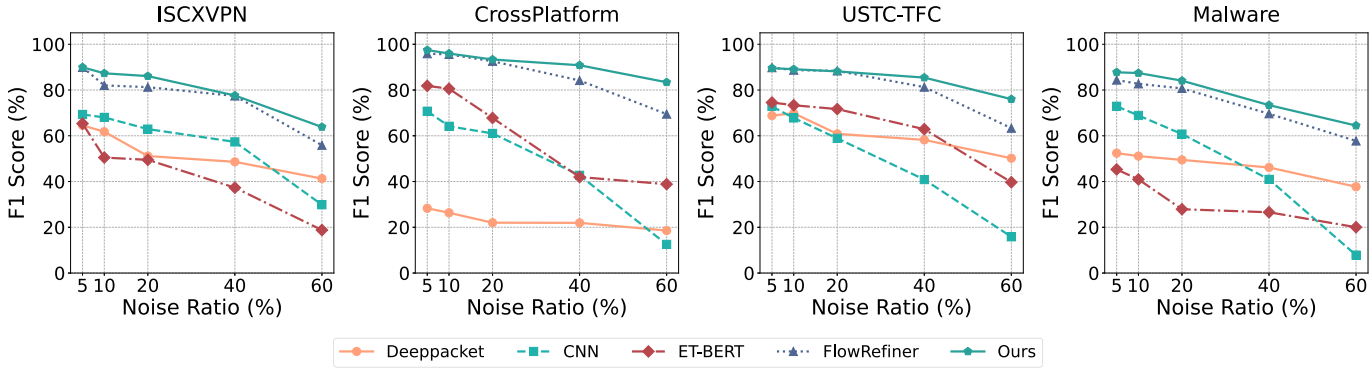


Fig. 5. Comparison of packet classification performance with packet-level baselines.

in FLOWREFINER+ is used to divide high-confidence labels and low-confidence labels based on prediction confidence, which is more achievable compared to DivideMix's noisy label identification. The results show that FLOWREFINER+ achieves high performance under different noise ratios and leads the noisy label learning baseline by a large margin.

3) Packet Classification Ability: The proposed cross-granularity robust classifier integrates both flow-level and packet-level traffic classification tasks against label noise, bringing the ability to serve as a robust packet classifier. We compare the packet-level classification performance with advanced packet classifiers, including Deeppacket [56], CNN [17], and ET-BERT [30]. As shown in Figure 5, in packet-level traffic analysis tasks, where each sample contains less information, the classification methods suffer from label noise more. Only our method can achieve F1 scores higher than 80% under noise ratios of 20% on all four datasets. Furthermore, the adaptive designs in FLOWREFINER+ significantly improve the stability over the static FlowRefiner, particularly under high noise rates. It demonstrates that FLOWREFINER+ could obtain two levels of robust encrypted traffic classifier at once and adapt to different requirements.

4) Class-Dependent Noise Evaluation: Class-dependent noise occurs when the probability of label corruption is not uniform across classes, meaning certain classes are more prone to having mislabeled samples than others. We perform the evaluation on ISCXVPN and USTC-TFC with different ratios of class-dependent noise, where categories with similar patterns are more likely to be mislabeled as each other. Specifically, on ISCXVPN, traffic is confused between similar types like VOIP/Streaming and FTP/MAIL, while on USTC-TFC, confusion occurs among protocols sharing file transfer characteristics (FTP/SMB) or similar real-time features (Skype/Facetime, WoW/Skype).

We comprehensively performed a comparison with both traffic classification methods and label noise methods under class-dependent noise, which are shown in Table V and Table VI. It can be observed that almost all methods perform worse compared to the complete random noise setting, since the class-dependent noise leads to more severe category confusion and shifted decision boundaries. In particular,

Authorized licensed use limited to: BEIJING INSTITUTE OF TECHNOLOGY. Downloaded on November 15,2026 at 03:10:37 UTC from IEEE Xplore. Restrictions apply.

TABLE V
COMPARISON WITH TRAFFIC CLASSIFICATION BASELINES ON CLASS-DEPENDENT NOISE SCENARIOS

Dataset	Noise	Appscanner		FS-Net		ET-BERT		MAE		YaTC		MCRe		FlowRefiner		Ours	
		Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
ISCXVPN	5%	81.49	82.07	87.42	87.58	85.76	85.77	91.39	91.36	93.15	92.98	80.86	81.72	93.32	92.98	**93.50**	**93.29**
	10%	81.16	81.70	85.78	86.21	81.90	81.69	90.51	90.45	91.21	91.13	78.52	79.29	92.62	92.28	**93.15**	**92.79**
	20%	80.51	80.36	83.04	83.33	80.49	80.52	84.35	84.44	88.75	88.64	77.54	77.38	89.45	88.90	**93.32**	**93.11**
	40%	67.20	68.15	69.14	70.29	55.18	55.99	68.71	69.50	67.31	68.28	74.61	74.75	78.03	77.29	**86.12**	**85.37**
	60%	49.67	51.14	47.10	47.95	46.04	48.23	46.74	48.35	42.00	44.03	45.51	46.29	52.72	53.38	**61.51**	**62.30**
USTC-TFC	5%	62.14	62.15	86.33	86.53	94.20	94.18	93.58	93.52	95.24	94.20	94.12	94.25	94.62	94.50	**97.10**	**97.12**
	10%	61.38	61.38	85.78	85.98	92.96	93.07	93.79	93.75	93.17	93.18	90.23	92.40	94.20	94.16	**95.86**	**95.80**
	20%	60.86	55.61	80.39	79.97	91.51	91.52	88.61	88.62	90.68	90.65	89.45	91.80	92.13	92.15	**94.20**	**94.18**
	40%	54.98	50.17	76.56	77.01	75.98	76.41	78.88	79.99	83.43	83.73	82.03	83.87	84.47	84.90	**89.44**	**89.23**
	60%	42.19	41.75	62.34	62.52	54.65	57.11	56.10	57.27	66.87	68.28	35.55	40.02	75.36	76.30	**76.40**	**77.24**

TABLE VI
COMPARISON WITH LABEL NOISE LEARNING BASELINES ON CLASS-DEPENDENT NOISE SCENARIOS

Dataset	Noise	CE		LSR		Mixup		GCE		SCE		Co-teaching		DivideMix		Ours	
		Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
ISCXVPN	5%	78.91	79.06	79.61	79.66	82.95	82.74	76.09	74.58	79.09	79.06	85.41	84.44	85.24	85.37	**93.50**	**93.29**
	10%	76.09	75.31	77.85	77.34	78.91	78.76	75.04	73.88	77.50	76.47	86.46	85.74	80.84	81.37	**93.15**	**92.79**
	20%	76.44	76.12	74.34	74.02	76.80	76.67	72.93	71.89	74.34	74.77	87.69	87.37	79.61	80.15	**93.32**	**93.11**
	40%	58.52	59.16	59.75	59.91	67.48	67.39	68.18	67.23	64.49	64.37	75.57	75.50	76.63	77.00	**86.12**	**85.37**
	60%	38.84	40.05	34.62	36.01	43.76	45.73	42.88	42.03	41.82	44.02	51.85	50.94	50.44	53.44	**61.51**	**62.30**
USTC-TFC	5%	87.16	86.73	86.33	85.67	90.68	90.45	80.12	78.79	86.75	86.45	76.81	73.84	85.92	85.72	**97.10**	**97.12**
	10%	85.92	86.06	83.85	83.31	89.23	88.51	79.50	78.66	85.92	85.86	77.64	74.63	90.48	90.41	**95.86**	**95.80**
	20%	81.78	81.27	82.81	82.27	83.22	82.49	78.67	76.84	80.12	79.90	83.44	80.90	89.23	88.98	**94.20**	**94.18**
	40%	65.21	65.18	66.04	66.20	74.12	74.41	71.42	70.16	71.22	70.62	82.40	81.75	81.78	82.25	**89.44**	**89.23**
	60%	58.17	58.86	51.55	51.65	55.69	56.93	64.80	65.05	48.44	49.47	68.12	69.73	65.22	66.35	**76.40**	**77.24**

comparing FLOWREFINER+ with its non-adaptive variant, FlowRefiner, the performance gap widens significantly as the noise ratio increases. For instance, at 60% class-dependent noise on ISCXVPN, FLOWREFINER+ achieves 62.30% F1, a notable improvement over FlowRefiner's 53.38% F1. This performance divergence confirms that static thresholds and fixed cluster sizes are inadequate for delineating boundaries between highly correlated and confusing classes. Furthermore, as demonstrated in Table VI, even against strong label noise learning baselines like DivideMix and Co-teaching, FLOWREFINER+'s adaptive approach provides superior robustness, achieving up to 93.29% F1 on ISCXVPN at 5% noise and maintaining 77.24% F1 on USTC-TFC at 60% noise. Note that our method can still achieve significant performance advantages over the optimal baseline, demonstrating better robustness under different noise types.

C. Label Noise Evaluation

1) Noise Detection Performance: We evaluate the performance of the traffic semantics-driven noise detector in the left part of Figure 6. The detector consistently identifies a large portion of noisy-labeled flows across all datasets, with the detection ratio, i.e., the proportion of detected noise among all noisy samples, remaining above 80% in most datasets and noise settings, even at 60% noise ratio. Furthermore, we also evaluate the noise detection performance under the class-dependent label noise settings in Figure 7. As shown in the left part of Figure 7, the detection ratio under class-dependent noise is generally lower compared to general label noise, due to the semantic proximity of confused classes. Nevertheless, the detector still maintains high effectiveness, confirming its

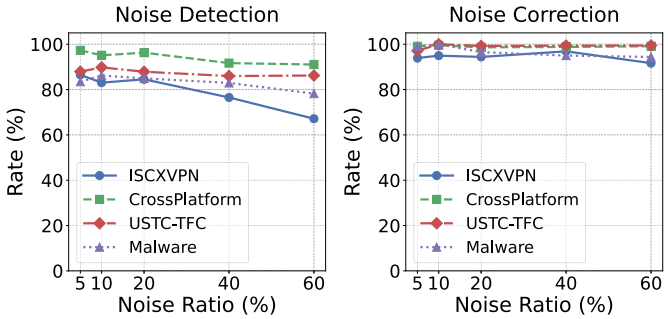


Fig. 6. Performance of noise detection and noise correction in four datasets under general label noise.

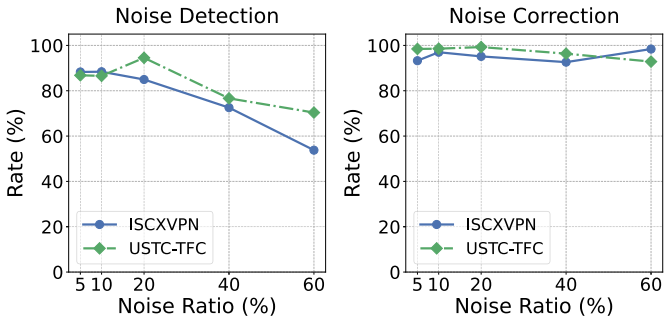


Fig. 7. Performance of noise detection and noise correction on ISCXVPN and USTC-TFC under class-dependent label noise.

Authorized licensed use limited to: BEIJING INSTITUTE OF TECHNOLOGY. Downloaded on November 15,2026 at 03:10:37 UTC from IEEE Xplore. Restrictions apply.

TABLE VII
REAL-WORLD NOISY DATASET EVALUATION ON CICIDS2017 [51]

Method	Classification	Noise Detection	Noise Correction
MCRe	94.51	47.31	59.23
FlowRefiner	96.19	84.61	81.92
FLOWREFINER$^+$	**99.10**	**86.27**	**99.58**

robustness across different noise types. This robustness is particularly valuable in security-sensitive tasks such as malware detection, where undetected noise may lead to critical failures. These results confirm the detector's effectiveness in isolating noisy flows and supporting reliable traffic annotation and management.

2) Noise Correction Performance: The confidence-guided label correction is responsible for correcting the labels of the previously selected noisy flows. The correction ratio, i.e. accuracy of the noisy flow correction, is shown in the right part of Figure 6. It can be seen that our method can accurately select and assign the correct labels to noisy flows based on prediction confidence, consistently achieving a correction accuracy of over 90% in all scenarios, and exceeding 95% in the majority of settings. Particularly on the CrossPlatform and USTC-TFC datasets, labels are almost all correctly assigned to the selected noisy flows, demonstrating the notable performance of the label predictor in this module. Furthermore, we analyze the correction performance under class-dependent noise in the right part of Figure 7. Despite the difficulty posed by semantically confused samples, the correction accuracy remains robustly high, consistently staying above 90% across both datasets. This superior correction performance highlights the critical role of our adaptive GMM mechanism in accurately delineating the confidence boundary between clean and noisy samples, thereby ensuring that refined flows are highly trustworthy. Notably, this robustness holds even though the noise detector leaves more residual noise in $\mathcal{D}_{\text{clean}}$ under class-dependent settings, where the detection ratio is lower. In such cases, the fine-tuned predictor still assigns high confidence to predictions consistent with the dominant clean distribution in $\mathcal{D}_{\text{clean}}$, while predictions influenced by the residual noisy minority conflict with the dominant clean distribution and receive lower confidence. The GMM-based separation thus naturally suppresses error cascading by excluding low-confidence corrections from $\mathcal{D}_{\text{refined}}$, which explains why correction accuracy remains high even when detection is imperfect.

3) Real-World Noisy Dataset Evaluation: Previous studies [21], [22] have revealed significant label noise in the CICIDS2017 dataset [51], and provide a corrected version with revised flow labels. Based on this dataset, we conduct a real-world evaluation where all models are trained on the original training set with its original labels, which are known to be noisy. For classification performance, the test set is evaluated against the manually corrected ground truth from [21]. For noise detection and correction metrics, the manually corrected labels of the training samples serve as the reference, which are introduced only during post-hoc metric computation and play no role in the training pipeline, ensuring strict quality control and eliminating any risk of data leakage. MCRe, the other traffic label noise method and the only baseline capable of both detecting and correcting label noise, is introduced to the comparison. As shown in Table VII, the comparison demonstrates the effectiveness of our framework and the necessity of our adaptive design in handling complex real-world data distributions. The non-adaptive FlowRefiner variant successfully identified 84.61% of the noisy flows and achieved a correction rate of 81.92%. In contrast, FLOWREFINER$^+$ substantially advances the performance, achieving 86.27% in noise detection and an impressive 99.58% in noise correction. This good correction demonstrates the robustness of our GMM-based adaptive boundary setting in real-world scenarios. Furthermore, regarding overall classification accuracy, FLOWREFINER$^+$ achieves 99.10%, which is a notable improvement over FlowRefiner's 96.19% and MCRe's 94.51%. The significant leap in classification performance from FlowRefiner to FLOWREFINER$^+$ confirms the crucial impact of our adaptive mechanism on refining the training data quality and improving generalization in practical applications.

D. Ablation Study

The ablation study is conducted to evaluate the contribution of each component in FLOWREFINER$^+$. Ablating a component also disables the subsequent components that depend on its output. As shown in Table VIII and Table IX, the performance declines consistently on all traffic datasets when any of the key components are removed. In particular, the removal of the traffic semantics-driven noise detector (TSND) results in the most severe performance degradation under high noise ratios. For example, on the ISCXVPN dataset with 60% noise, removing TSND leads to a significant drop in F1 score from 75.40% to 48.60%. Similarly, on the CrossPlatform dataset, F1 score drops from 94.87% to 46.85%. This highlights the critical role of the TSND in accurately identifying and isolating noisy flows, which ensures that the subsequent components operate with more reliable data. When the confidence-guided label correction (CLC) is ablated, the performance also degrades, suggesting that this component can effectively mitigate label noise by handling difficult cases and expanding the clean set. Finally, the removal of the structure of cross-granularity robust classifier (CRC) results in stable performance reductions under all datasets and noise ratios. It proves that the joint task of both flow and packet classification could improve the robustness under different noise ratios. Ablating CRC removes the concurrent packet-level classification task and reverts the model to flow-level classification only, without the random single-packet sampling. Crucially, the performance gap grows substantially with increasing noise ratio, e.g., on CrossPlatform, removing CRC causes only a 0.89% F1 drop at 5% noise but a 16.47% drop at 60% noise, demonstrating that CRC functions as a genuine anti-noise mechanism rather than mere data augmentation. The reason is that a randomly selected single packet carries only partial flow information, making it resistant to the flow-specific pattern memorization induced by noisy labels, yet sufficient for learning generalizable class-level semantics. Via shared encoder parameters,

Authorized licensed use limited to: BEIJING INSTITUTE OF TECHNOLOGY. Downloaded on November 15,2026 at 03:10:37 UTC from IEEE Xplore. Restrictions apply.

TABLE VIII
ABLATION STUDY ON ISCXVPN AND CROSSPLATFORM

Components			ISCXVPN										CrossPlatform									
TSND	CLC	CRC	5%		10%		20%		40%		60%		5%		10%		20%		40%		60%	
			Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
✓	✓	✓	**93.85**	**93.61**	**92.27**	**92.04**	**90.86**	**90.32**	**86.47**	**86.24**	**75.40**	**75.40**	**99.93**	**99.93**	**99.85**	**99.85**	**99.13**	**99.11**	**98.32**	**98.31**	**95.04**	**94.87**
×	×	×	93.14	93.00	89.28	88.99	84.18	84.10	63.26	64.27	47.62	48.60	97.67	97.69	95.92	95.93	88.41	88.52	68.00	68.84	44.89	46.85
✓	×	×	88.40	88.01	88.40	87.94	88.05	87.72	84.36	83.74	71.70	71.60	91.76	91.44	91.55	91.15	91.40	90.97	86.95	86.61	78.64	78.04
✓	✓	×	91.39	91.13	90.16	89.95	89.98	89.61	82.95	82.54	70.47	69.15	99.05	99.04	98.47	98.46	96.72	96.72	89.65	89.72	79.23	78.40
w/o TS	✓	✓	82.60	80.84	73.99	72.51	65.03	60.91	50.09	44.79	29.70	18.17	96.21	96.09	93.51	92.83	92.20	91.28	86.81	84.33	69.17	63.11

TABLE IX
ABLATION STUDY ON USTC-TFC AND MALWARE

Components			USTC-TFC										Malware									
TSND	CLC	CRC	5%		10%		20%		40%		60%		5%		10%		20%		40%		60%	
			Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
✓	✓	✓	**96.69**	**96.68**	**95.86**	**95.79**	**95.45**	**95.41**	**92.55**	**92.40**	**87.58**	**87.47**	**96.22**	**96.22**	**95.27**	**95.27**	**92.03**	**91.97**	**85.14**	**84.87**	**74.86**	**74.81**
×	×	×	96.07	95.99	93.99	93.87	91.51	91.43	75.56	75.66	48.44	48.95	93.11	93.14	92.30	92.27	84.05	84.14	67.70	67.65	47.02	47.42
✓	×	×	94.82	94.84	95.65	95.57	94.41	94.35	91.72	91.47	86.54	86.08	89.19	89.23	90.95	91.00	82.97	82.78	80.14	79.81	69.59	69.75
✓	✓	×	95.86	95.82	95.44	95.34	94.82	94.78	91.92	91.97	87.16	86.95	92.16	92.20	92.03	92.00	88.65	88.62	81.89	81.87	71.22	70.67
w/o TS	✓	✓	81.16	79.41	78.05	75.86	75.16	72.33	67.08	62.27	48.65	40.17	81.49	81.41	80.27	79.48	76.49	74.43	67.03	65.12	60.54	56.62

this noise-resistant packet-level supervision regularizes the flow-level classifier against label noise memorization across both granularities. Note that our method has less performance degradation on the USTC-TFC dataset compared to other datasets under the ablation of CLC and CRC. This is because it contains both benign and malicious traffic with obvious separability, which is less difficult to classify after the noise detection.

We further ablate the pre-trained traffic semantics (w/o TS in TSND) by replacing the MAE pre-trained encoder with a randomly initialized counterpart, while keeping all three components intact. Unlike conventional handcrafted statistical features, traffic semantics are label-independent latent representations learned via self-supervised pre-training, capturing intrinsic structural patterns of flows regardless of their labels. This label-independence is what enables the noise detector to identify label-inconsistent flows through semantic clustering. The results show that removing pre-trained semantics causes catastrophic degradation: on ISCXVPN at 60% noise, F1 collapses to 18.17%, which is even lower than removing TSND entirely (48.60%), since random features corrupt the clustering and actively mislabel clean samples as noisy. This confirms that pre-trained traffic semantics are the indispensable foundation of the entire framework.

Overall, FLOWREFINER+ effectively detects, corrects, and classifies traffic with noisy labels in a uniform framework, where each component shows a certain contribution to the performance.

E. Discussions

1) Timeliness: Traffic patterns have evolved with new protocols and obfuscation techniques. To assess whether FLOWREFINER+ generalizes to recently collected traffic, we further evaluate it on the newly released RealNetwork dataset [52] (2025), which reflects modern encrypted application traffic patterns. As shown in Table X, FLOWREFINER+ consistently outperforms all baselines across all noise ratios on this newly collected data. The performance gap widens under higher noise, reaching over 10% F1 improvement against the best-performing baseline at 60% noise. These results confirm that FLOWREFINER+ maintains strong noise robustness on recently collected data, demonstrating that its noise detection, correction, and classification pipeline is applicable beyond legacy benchmarks.

TABLE X
PERFORMANCE COMPARISON ON THE NEWLY RELEASED DATASET UNDER DIFFERENT NOISE RATIOS

Method	5%		10%		20%		40%		60%	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
AppScanner	56.92	54.08	57.09	54.76	55.69	52.64	48.86	45.91	43.43	41.36
FS-Net	68.05	67.65	65.16	64.53	56.48	56.14	47.81	47.12	33.67	33.62
ET-BERT	80.71	80.45	77.86	77.78	70.29	69.66	60.43	60.52	42.14	41.62
MAE	80.86	80.82	77.00	76.88	69.43	69.46	57.14	56.96	37.29	37.35
YaTC	84.00	83.94	83.43	83.36	76.57	76.41	60.00	59.94	39.00	39.00
MCRe	72.07	73.97	72.27	74.38	65.04	67.50	57.62	59.85	46.48	46.73
CE	67.57	67.60	65.43	65.28	57.00	56.14	43.29	42.54	32.86	32.47
LSR	69.43	69.24	65.71	65.51	53.86	53.45	44.14	43.79	30.57	29.89
Mixup	70.00	69.64	66.57	66.68	62.00	61.38	46.57	45.76	33.71	32.62
GCE	62.43	62.87	61.29	61.44	58.29	58.28	46.29	46.07	36.00	35.88
SCE	68.43	68.53	66.43	66.45	59.14	58.67	47.57	47.40	35.14	34.03
Co-teaching	73.00	72.86	71.00	70.34	71.57	71.35	67.57	67.76	55.00	55.44
DivideMix	77.57	77.69	75.71	76.30	74.29	74.37	65.29	65.30	54.71	55.18
FlowRefiner	87.14	87.08	85.00	84.85	80.57	80.48	73.57	73.57	54.00	53.99
Ours	**87.71**	**87.60**	**88.14**	**87.27**	**87.31**	**84.33**	**76.00**	**75.69**	**65.14**	**65.30**

2) Parameter Sensitivity: The threshold δ in the Agglomerative Clustering algorithm controls cluster granularity: two clusters are merged only if their linkage distance falls below δ, directly determining how finely the semantic space is

Authorized licensed use limited to: BEIJING INSTITUTE OF TECHNOLOGY. Downloaded on November 15,2026 at 03:10:37 UTC from IEEE Xplore. Restrictions apply.

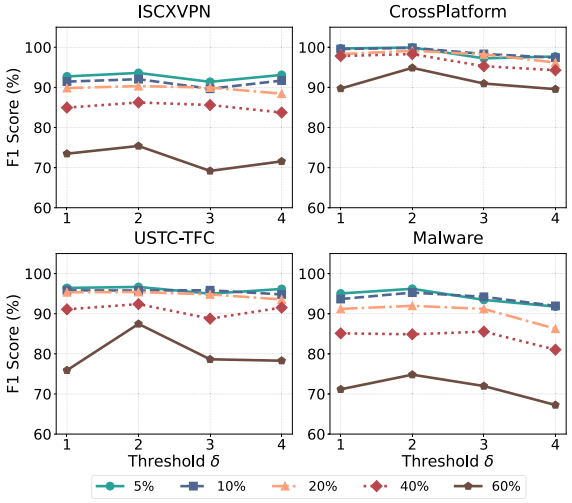


Fig. 8. Parameter sensitivity of δ in agglomerative clustering.

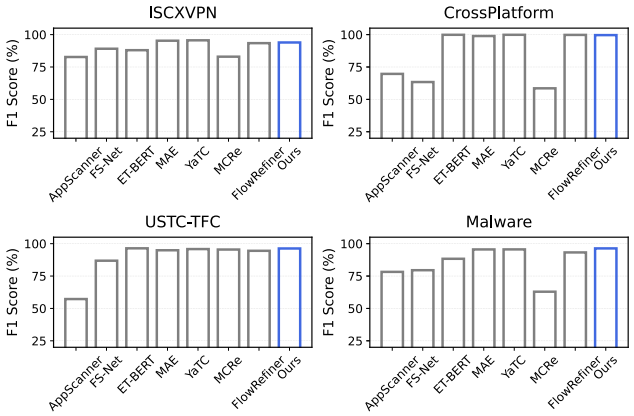


Fig. 9. Performance under zero label noise.

partitioned for noise detection. We evaluate classification performance under $\delta \in \{1, 2, 3, 4\}$ across four datasets and five noise ratios, as shown in Figure 8. The value $\delta = 2$ consistently achieves the best or near-best performance, with the advantage most pronounced at high noise ratios (e.g., 75.40% vs. 73.47%/69.18% F1 on ISCXVPN at 60% noise; 94.87% vs. 89.71%/90.96% on CrossPlatform at 60% noise). A smaller δ over-fragments the semantic space, causing semantically similar flows to be split into isolated clusters and increasing false positives in noise detection; a larger δ merges semantically distinct flows and reduces the detector's discriminative power. Overall, the model is not overly sensitive to δ: under low noise ratios ($\leq 20\%$), performance remains stable across different δ values, while the advantage of $\delta = 2$ becomes more pronounced as the noise level increases.

3) Zero-Noise Condition: Although FLOWREFINER$^+$ is designed for noisy-label settings, it is important to verify that the framework does not degrade performance when labels are clean. Figure 9 compares the F1 scores of all methods under the 0% noise condition on four datasets. FLOWREFINER$^+$ remains competitive with the best-performing traffic classification methods (e.g., YaTC, MAE)

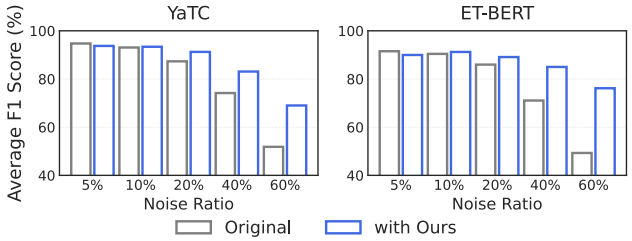


Fig. 10. The applicability of FLOWREFINER$^+$ for enhancing the robustness of existing traffic classification architectures under label noise.

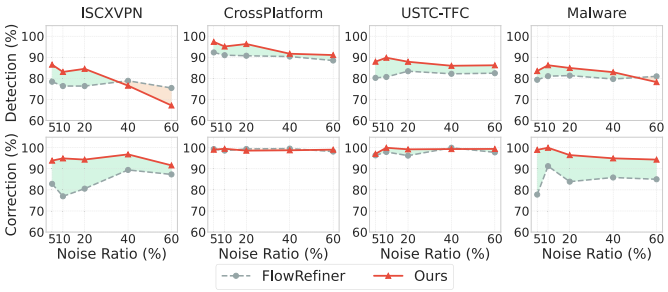


Fig. 11. Noise detection and noise correction comparison of FLOWREFINER$^+$ with the variant FlowRefiner without adaptive designs.

across all datasets, despite the additional noise-handling components. This demonstrates that the semantic-driven noise detection and adaptive correction introduce no significant degradation in classification accuracy when clean labels are available, confirming the framework's robustness in both noisy and noise-free deployment environments.

4) Adaptiveness: To validate the criticality of the proposed data-driven designs, we compare FLOWREFINER$^+$ against its static variant FlowRefiner, which employs the same semantic structure but relies on fixed hyperparameters throughout. FlowRefiner requires four manually set hyperparameters: cluster count (K) and majority label count (m) for noise detection, alongside fixed confidence thresholds (τ_h, τ_l) for noise correction. FLOWREFINER$^+$ reduces this to a single parameter, the linkage distance δ: the max-gap function eliminates m by adaptively inferring the majority label boundary from the data distribution; the GMM eliminates τ_h and τ_l by automatically modeling the confidence distribution; and hierarchical clustering replaces the rigid K with the more interpretable δ. As shown in the Parameter Sensitivity analysis, δ is stable across a wide range of values, placing minimal tuning burden on practitioners. As shown in Figure 11, the adaptive designs yield overall gains, with improvements most pronounced in label correction. On ISCXVPN and Malware, FLOWREFINER$^+$ achieves notably higher correction accuracy than FlowRefiner across all noise ratios, while detection improvements are more moderate. This confirms that the adaptive GMM boundary is especially effective at separating high- and low-confidence predictions in complex traffic distributions.

5) Applicability to Other Architectures: Beyond its standalone use, FLOWREFINER$^+$ is designed to be compatible with existing pre-trained traffic classification architectures.

Authorized licensed use limited to: BEIJING INSTITUTE OF TECHNOLOGY. Downloaded on November 15,2026 at 03:10:37 UTC from IEEE Xplore. Restrictions apply.

TABLE XI
INFERENCE EFFICIENCY OF FLOWREFINER+ ON DIFFERENT HARDWARE PLATFORMS

	GPU	CPU	Edge Device
Inference Time (ms/sample)	0.24	5.31	145.07
Throughput (samples/s)	4,226.88	188.26	6.89

Specifically, any pre-trained model can serve as the traffic encoder, label predictor, and traffic classifier within our framework, enabling it to benefit from our noise detection, correction, and robust classification pipeline. To validate this, we integrate FLOWREFINER+ with two representative pre-trained baselines, YaTC and ET-BERT, and evaluate the average F1 scores across four datasets, as shown in Figure 10. The results show that incorporating FLOWREFINER+ consistently and substantially improves robustness under high noise ratios: at 60% noise, YaTC improves from 51.83% to 69.01% F1, and ET-BERT improves from 49.29% to 76.16% F1. At lower noise ratios, the improvement is more moderate, as both architectures already perform reasonably well in relatively clean settings. These results demonstrate that FLOWREFINER+ can serve as a general noise-handling plugin for diverse traffic classification architectures, broadening its applicability as a unified solution for label noise in various network environments.

6) Inference Efficiency and Deployment: We evaluate the inference efficiency of FLOWREFINER+ across three representative hardware platforms, i.e., a GPU (NVIDIA RTX 3090), a standard CPU, and a simulated edge device (single-core CPU, 50% CPU quota, 1 GB memory, AVX2 instruction set), to assess its practical deployability. As shown in Table XI, the GPU setting achieves an inference latency of 0.24 ms per sample and a throughput of 4,226.88 samples/s, which is well-suited to high-performance network monitoring systems. On a standard CPU, the throughput of 188.26 samples/s supports moderate-scale offline analysis workflows. Even on the resource-constrained edge device, FLOWREFINER+ maintains a throughput of 6.89 samples/s, demonstrating its feasibility for lightweight deployment at the network edge. These results confirm that FLOWREFINER+ can be flexibly deployed across a wide range of network environments, from high-throughput data centers to resource-limited edge nodes.

7) Complexity of Clustering: The Agglomerative Clustering algorithm in the noise detector has a time complexity of $\mathcal{O}(n^2 \log n)$ and a space complexity of $\mathcal{O}(n^2)$, where n is the number of labeled training samples. Since noise detection is performed offline as a one-time preprocessing step, the clustering cost is entirely decoupled from real-time traffic classification and does not affect inference efficiency. Moreover, labeled datasets in traffic classification scenarios are typically of moderate scale, making the overhead practically manageable. Should the labeled corpus be exceptionally large, the data can be partitioned into multiple batches with noise detection applied to each batch independently, preserving both detection quality and computational feasibility.

8) Limitations: As a deep learning-based method, FLOWREFINER+ requires substantial computational resources for efficient training, preferably with GPU acceleration. Our framework currently takes about 3 minutes on an RTX 3090 GPU with 2.79 GB of memory for training. In future work, we will continue to optimize the training pipeline to reduce computational overhead and improve accessibility, while also leveraging ongoing progress in lightweight inference and dedicated hardware accelerators, which are expected to further enhance the practicality and scalability of FLOWREFINER+ in real-world deployments.

V. CONCLUSION

In this paper, we proposed FLOWREFINER+, a robust framework for general traffic classification under noisy label conditions. Our method addresses the significant challenges of label noise, including its detection, correction, and robust classification, thus reducing dependency on high-quality labeled data. Through the integration of a semantic-driven noise detector, confidence-guided label correction, and a cross-granularity robust classifier, FLOWREFINER+ effectively leverages noisy traffic to improve classification performance. Our experimental results across various datasets and scenarios demonstrate the ability of FLOWREFINER+ to outperform state-of-the-art methods in both accuracy and resilience to noise. This framework provides a valuable solution for enhancing general traffic analysis, particularly in environments where high-quality labels are difficult to obtain, and sets the stage for future advancements in traffic classification under noisy conditions.

REFERENCES

[1] M. Zhan et al., "FlowRefiner: A robust traffic classification framework against label noise," in *Proc. Adv. Neural Inf. Process. Syst. 38*, 2025, pp. 142207–142230.
[2] C. Fu, Q. Li, M. Shen, and K. Xu, "Frequency domain feature based robust malicious traffic detection," *IEEE/ACM Trans. Netw.*, vol. 31, no. 1, pp. 452–467, Feb. 2023.
[3] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An ensemble of autoencoders for online network intrusion detection," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2018, pp. 274–288.
[4] C. Fu, Q. Li, and K. Xu, "Detecting unknown encrypted malicious traffic in real time via flow interaction graph analysis," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2023, pp. 1–18.
[5] E. Papadogiannaki and S. Ioannidis, "A survey on encrypted network traffic analysis applications, techniques, and countermeasures," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 1–35, Jul. 2022.
[6] G. Zhou, Z. Liu, C. Fu, Q. Li, and K. Xu, "An efficient design of intelligent network data plane," in *Proc. 32nd USENIX Security Symp. (USENIX Security)*, 2023, pp. 6203–6220.
[7] R. Xie et al., "Rosetta: Enabling robust TLS encrypted traffic classification in diverse network environments with TCP-aware traffic augmentation," in *Proc. 32nd USENIX Secur. Symp. (USENIX Secur.)*, 2023, pp. 625–642.
[8] V. F. Taylor, R. Spolaor, M. Conti, and I. Martinovic, "AppScanner: Automatic fingerprinting of smartphone apps from encrypted network traffic," in *Proc. IEEE Eur. Symp. Secur. Privacy*, Mar. 2016, pp. 439–454.
[9] F. Al-Obaidy, S. Momtahen, M. F. Hossain, and F. Mohammadi, "Encrypted traffic classification based ML for identifying different social media applications," in *Proc. IEEE Can. Conf. Electr. Comput. Eng. (CCECE)*, May 2019, pp. 1–5.
[10] M. Conti, L. V. Mancini, R. Spolaor, and N. V. Verde, "Analyzing Android encrypted network traffic to identify user actions," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 1, pp. 114–125, Jan. 2016.
[11] P. Velan, M. Čermák, P. Čeleda, and M. Drašar, "A survey of methods for encrypted traffic classification and analysis," *Int. J. Netw. Manage.*, vol. 25, no. 5, pp. 355–374, Sep. 2015.
[12] Z. Cao, G. Xiong, Y. T. Zhao, Z. Li, and L. Guo, "A survey on encrypted traffic classification," in *Proc. Int. Conf. Appl. Techn. Inf. Secur.*, 2014, pp. 73–81.

Authorized licensed use limited to: BEIJING INSTITUTE OF TECHNOLOGY. Downloaded on November 15,2026 at 03:10:37 UTC from IEEE Xplore. Restrictions apply.

[13] M. Shen et al., "Machine learning-powered encrypted network traffic analysis: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 1, pp. 791–824, 1st Quart., 2023.
[14] M. Shen, Y. Liu, L. Zhu, K. Xu, X. Du, and N. Guizani, "Optimizing feature selection for efficient encrypted traffic classification: A systematic approach," *IEEE Netw.*, vol. 34, no. 4, pp. 20–27, Jul. 2020.
[15] C. Liu, L. He, G. Xiong, Z. Cao, and Z. Li, "FS-Net: A flow sequence network for encrypted traffic classification," in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, Apr. 2019, pp. 1171–1179.
[16] Y. Zeng, H. Gu, W. Wei, and Y. Guo, "Deep-full-range: A deep learning based network encrypted traffic classification and intrusion detection framework," *IEEE Access*, vol. 7, pp. 45182–45190, 2019.
[17] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Da Nang, Vietnam, Jan. 2017, pp. 712–717.
[18] J. Zhang, F. Li, F. Ye, and H. Wu, "Autonomous unknown-application filtering and labeling for DL-based traffic classifier update," in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, Jul. 2020, pp. 397–405.
[19] K. Lin, X. Xu, and H. Gao, "TSCRNN: A novel classification scheme of encrypted traffic based on flow spatiotemporal features for efficient management of IIoT," *Comput. Netw.*, vol. 190, May 2021, Art. no. 107974.
[20] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, "Learning from noisy labels with deep neural networks: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 11, pp. 8135–8153, Nov. 2023.
[21] G. Engelen, V. Rimmer, and W. Joosen, "Troubleshooting an intrusion detection dataset: The CICIDS2017 case study," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2021, pp. 7–12.
[22] L. Liu, G. Engelen, T. Lynar, D. Essam, and W. Joosen, "Error prevalence in NIDS datasets: A case study on CIC-IDS-2017 and CSE-CIC-IDS-2018," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Oct. 2022, pp. 254–262.
[23] J. L. Guerra, C. Catania, and E. Veas, "Datasets are not enough: Challenges in labeling network traffic," *Comput. Secur.*, vol. 120, Sep. 2022, Art. no. 102810.
[24] Y. Qing et al., "Low-quality training data only? A robust framework for detecting encrypted malicious network traffic," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2024, pp. 1–18.
[25] Q. Yuan, G. Gou, Y. Zhu, Y. Zhu, G. Xiong, and Y. Wang, "MCRe: A unified framework for handling malicious traffic with noise labels based on multidimensional constraint representation," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 133–147, 2024.
[26] J. Li, R. Socher, and S. C. H. Hoi, "DivideMix: Learning with noisy labels as semi-supervised learning," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2020, pp. 8923–8936.
[27] B. Han et al., "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 8527–8537.
[28] Z. Zhang and M. R. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 8792–8802.
[29] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. Bailey, "Symmetric cross entropy for robust learning with noisy labels," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 322–330.
[30] X. Lin, G. Xiong, G. Gou, Z. Li, J. Shi, and J. Yu, "ET-BERT: A contextualized datagram representation with pre-training transformers for encrypted traffic classification," in *Proc. ACM Web Conf.*, Apr. 2022, pp. 633–642.
[31] R. Zhao et al., "Yet another traffic classifier: A masked autoencoder based traffic transformer with multi-level flow representation," in *Proc. 37th AAAI Conf. Artif. Intell. (AAAI)*, vol. 37, 2023, pp. 5420–5427.
[32] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 16000–16009.
[33] R. Zhao et al., "Learning flow semantics for encrypted traffic analysis: A contrastive pre-training approach," *IEEE Trans. Dependable Secure Comput.*, vol. 23, no. 4, pp. 7864–7877, Jul. 2026.
[34] T. T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Commun. Surveys Tuts.*, vol. 10, no. 4, pp. 56–76, 4th Quart., 2008.
[35] A. Panchenko et al., "Website fingerprinting at internet scale," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2016, pp. 584–598.
[36] H. Y. He, Z. Guo Yang, and X. N. Chen, "PERT: Payload encoding representation from transformer for encrypted traffic classification," in *Proc. ITU Kaleidoscope, Ind.-Driven Digit. Transformation (ITU K)*, Dec. 2020, pp. 1–8.
[37] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.* Minneapolis, MI, USA: Association for Computational Linguistics, vol. 1, Jun. 2019, pp. 4171–4186.
[38] Z. Hang, Y. Lu, Y. Wang, and Y. Xie, "Flow-MAE: Leveraging masked AutoEncoder for accurate, efficient and robust malicious traffic classification," in *Proc. 26th Int. Symp. Res. Attacks, Intrusions Defenses*, Oct. 2023, pp. 297–314.
[39] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning (still) requires rethinking generalization," *Commun. ACM*, vol. 64, no. 3, pp. 107–115, Mar. 2021.
[40] Q. Yuan et al., "BoAu: Malicious traffic detection with noise labels based on boundary augmentation," *Comput. Secur.*, vol. 131, Aug. 2023, Art. no. 103300.
[41] Q. Yuan et al., "ULDC: Unsupervised learning-based data cleaning for malicious traffic with high noise," *Comput. J.*, vol. 67, no. 3, pp. 976–987, Apr. 2024.
[42] Q. Meng et al., "When unknown threat meets label noise: A self-correcting framework," *IEEE Trans. Dependable Secure Comput.*, vol. 23, no. 1, pp. 905–922, Jan. 2026.
[43] Y. Qing et al., "Toward robust detection of malicious encrypted traffic using only low-quality training data," *IEEE Trans. Netw.*, vol. 34, pp. 5223–5238, 2026.
[44] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 2866–2878.
[45] F. Fooladgar, M. N. N. To, P. Mousavi, and P. Abolmaesumi, "Manifold DivideMix: A semi-supervised contrastive learning framework for severe label noise," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2024, pp. 4012–4021.
[46] D. Arpit et al., "A closer look at memorization in deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 233–242.
[47] S. Liu, J. Niles-Weed, N. Razavian, and C. Fernandez-Granda, "Early-learning regularization prevents memorization of noisy labels," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 33, 2020, pp. 20331–20342.
[48] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of encrypted and VPN traffic using time-related features," in *Proc. 2nd Int. Conf. Inf. Syst. Secur. Privacy*, 2016, pp. 407–414.
[49] T. Van Ede et al., "FlowPrint: Semi-supervised mobile-app fingerprinting on encrypted network traffic," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, 2020, pp. 369–386.
[50] Malware2023 Dataset. (2023). *A Source for Packet Capture (PCAP) Files and Malware Samples*. [Online]. Available: https://malware-traffic-analysis.net/
[51] I. Sharafaldin, A. Gharib, A. H. Lashkari, and A. A. Ghorbani, "Towards a reliable intrusion detection benchmark dataset," *Softw. Netw.*, vol. 2017, no. 1, pp. 177–200, 2017.
[52] W. Jiang, B. Zhang, Q. Zhu, C. Liao, and W. Wang, "A real network environment dataset for traffic analysis," *Scientific Data*, vol. 12, no. 1, p. 756, May 2025.
[53] M. Lukasik, S. Bhojanapalli, A. Menon, and S. Kumar, "Does label smoothing mitigate label noise?," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 6448–6458.
[54] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.
[55] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 4061–4078.
[56] M. Lotfollahi, M. J. Siavoshani, R. S. H. Zade, and M. Saberian, "Deep packet: A novel approach for encrypted traffic classification using deep learning," *Soft Comput.*, vol. 24, no. 3, pp. 1999–2012, Feb. 2020.



Mingwei Zhan received the M.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 2024, where he is currently pursuing the Ph.D. degree with the School of Computer Science. His research interests include deep learning, network traffic analysis, and network security.

Authorized licensed use limited to: BEIJING INSTITUTE OF TECHNOLOGY. Downloaded on November 15,2026 at 03:10:37 UTC from IEEE Xplore. Restrictions apply.



Ruijie Zhao received the M.S. and Ph.D. degrees from Shanghai Jiao Tong University, Shanghai, China, in 2021 and 2024, respectively. He is currently an Associate Professor with the School of Cyber Science and Engineering, Southeast University, Nanjing, China. His research interests include network security, traffic analysis, and AI-driven networking.



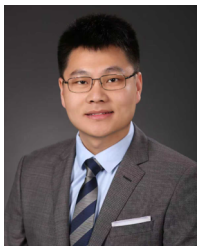
Xianwen Deng received the B.Eng. degree from Shanghai Jiao Tong University, Shanghai, China, in 2021, where he is currently pursuing the Ph.D. degree with the School of Computer Science. His current research interests include captcha systems, network traffic analysis, and deep learning.



Zhi Xue received the B.S. and Ph.D. degrees from Shanghai Jiao Tong University, Shanghai, China, in 1992 and 2001, respectively. He is currently a Professor with the School of Computer Science, Shanghai Jiao Tong University. He was the leader of more than 30 state/province projects of China, and authored or co-authored more than 60 articles. His research interests include machine learning, network security, and data science.



Qi Li (Senior Member, IEEE) received the Ph.D. degree from Tsinghua University, Beijing, China. He is currently an Associate Professor with the Institute for Network Sciences and Cyberspace, Tsinghua University. His research interests include internet security and machine learning security. He currently serves on the Editorial Board for IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, ACM TOPS, and ACM DTRAP.



Zhuotao Liu (Senior Member, IEEE) received the Ph.D. degree from the University of Illinois at Urbana–Champaign, USA. He is currently an Associate Professor with Tsinghua University. Before joining Tsinghua, he was a Technical Lead with Google, managing one of world's largest software-defined datacenter networks. His research interests include data/AI security and privacy, blockchain and applied cryptography, and secure internet architecture. His research received multiple Distinguished Paper Awards from USENIX Security and ACM CCS.



Guang Cheng (Member, IEEE) received the Ph.D. degree in computer science from Southeast University, Nanjing, China, in 2003. From 2006 to 2007, he was a Post-Doctoral Researcher with the School of Electrical and Computer Engineering, Georgia Institute of Technology. He is currently a Full Professor with the School of Cyber Science and Engineering, Southeast University. His research interests include network security, network measurement, and encrypted traffic analysis.



Ke Xu (Fellow, IEEE) received the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China. He is currently a Full Professor with the Department of Computer Science and Technology, Tsinghua University. He has published more than 200 technical articles and holds more than ten U.S. patents in the research areas of next-generation internet architecture, network security, and AI-driven networking. He is a member of ACM. He has received numerous accolades for his work, including multiple Distinguished Paper Awards at top-tier venues, such as USENIX Security in 2023 and 2024, respectively; ACM CCS 2025; and NDSS 2025.

Authorized licensed use limited to: BEIJING INSTITUTE OF TECHNOLOGY. Downloaded on November 15,2026 at 03:10:37 UTC from IEEE Xplore. Restrictions apply.