Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.077275

ARTICLE

NCRT: A Noise-Tolerant Label Recognition and Correction Framework for Network Traffic Detection

Yu Yang, Yuheng Gu*, Zhuoyun Yang and Shangjun Wu

College of Information Engineering, Chinese People's Armed Police Force Engineering University, Xi'an, China

*Corresponding Author: Yuheng Gu. Email: b15613085672@163.com

Received: 05 December 2025; Accepted: 26 February 2026; Published: 09 April 2026

ABSTRACT: The performance of network traffic detection heavily relies on high-quality annotated data, yet the widespread presence of noisy labels in real-world scenarios severely undermines model reliability and generalization. Existing methods predominantly rely on training dynamics signals and struggle to distinguish between noisy labels and valuable hard samples, leading to diminished model sensitivity to emerging threats. To address this fundamental challenge, this paper proposes a noise-tolerant label recognition and correction framework based on graph-structured neighbor consistency (NCRT). The framework leverages the inherent clustering characteristics of network traffic in feature space, constructs an adaptive K-nearest neighbor graph to model behavioral similarities among samples, and employs multi-dimensional consistency evaluation metrics to accurately differentiate between noisy labels and hard samples. Furthermore, a feature-space-based label propagation algorithm is introduced, which utilizes reliable anchor samples in the graph to guide noisy labels toward correction based on the consensus of their semantic neighbors. The core innovation lies in leveraging graph-structured neighbor consistency to fundamentally distinguish noisy labels from hard samples, and correcting them via label propagation over an adaptive K-nearest neighbor graph. Experiments on real-world datasets such as CICIDS2017 and NSL-KDD demonstrate that the proposed method significantly outperforms mainstream baseline approaches in terms of noise identification, label correction, and final classification performance. It exhibits exceptional robustness and stability, particularly under high-ratio and asymmetric noise settings, providing an effective solution for building reliable traffic detection models adapted to complex real-world network environments.

KEYWORDS: Network traffic detection; noisy labels; graph structure; neighbor consistency; label propagation

1 Introduction

As a core technology in network management and security defense, the performance of network traffic detection directly impacts the efficiency of threat detection, quality-of-service assurance, and network operation and maintenance [1–3]. In recent years, deep learning-based classification methods have been widely adopted in network traffic detection. However, their performance heavily relies on large-scale, high-quality labeled data [4]. In real-world network environments, acquiring such high-quality data is extremely challenging. Misjudgments by automated analysis tools, inconsistencies in crowd-sourced labeling, dynamic evolution of network environments, and the presence of adversarial attacks all contribute to the widespread issue of noisy labels in training data. The presence of incorrect labels not only reduces the classification accuracy of models but, more seriously, can lead models to learn erroneous feature representations, thereby introducing systemic bias in practical deployment [5,6].

Current research on the noisy label problem primarily revolves around three technical approaches (Fig. 1). The first category is loss correction methods, which modify the training objective by modeling noise transition matrices or designing noise-robust loss functions. Although theoretically sound, their effectiveness heavily depends on prior assumptions about the noise distribution, making it difficult to adapt to the complex and variable noise patterns in real network environments. The second category is sample selection strategies [7], which filter potentially clean samples based on dynamic metrics such as training loss. However, these methods suffer from an inherent flaw: they cannot effectively distinguish between genuine labeling noise and valuable hard samples. In network traffic classification tasks, hard samples of significant value, such as novel network attacks and malware variants, often also exhibit high loss values. Simply discarding these samples would prevent the model from recognizing emerging threats. The third category is meta-learning frameworks, which jointly learn model parameters and label purification through complex optimization processes. While these methods perform well in certain scenarios, their computational overhead is substantial, and their algorithmic complexity is high, making them difficult to apply in real network environments that need to handle massive traffic volumes.

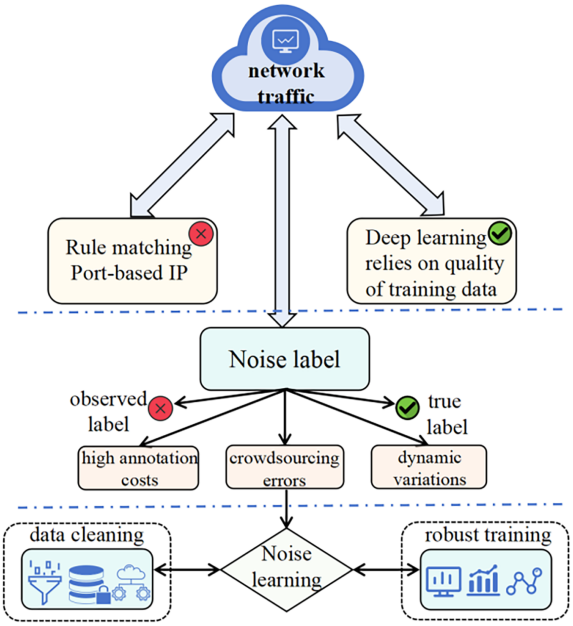

Figure 1: Technical approaches to the noisy label problem.

The common limitation of the aforementioned methods lies in their excessive reliance on dynamic signals during training, while neglecting the inherent structural information of data in the feature space. The fundamental reason is that dynamic signals such as training loss are inherently ambiguous. A high loss value may stem from an incorrect label (a noisy sample) but can equally arise from a sample that is intrinsically difficult due to its proximity to the decision boundary, ambiguous features, or representation of an emerging pattern (a hard sample). Consequently, methods that rely solely on these ambiguous signals struggle to differentiate between the two. This often leads to the erroneous filtering or down-weighting of valuable hard samples, which in turn deprives the model of the opportunity to learn from critical edge cases and novel threats, ultimately impairing its generalization and adaptive capabilities. More importantly, there is a widespread lack of a clear distinction mechanism between noisy labels and hard samples, often conflating the two. This leads to a loss of the model's ability to perceive potential emerging threat patterns during training [8]. Truly valuable hard samples—those located near decision boundaries—are mistakenly removed

or down-weighted due to their high loss values, hindering the classifier's necessary evolution and adaptation (Fig. 2). Consequently, the model becomes vulnerable when facing continuously evolving network threats.

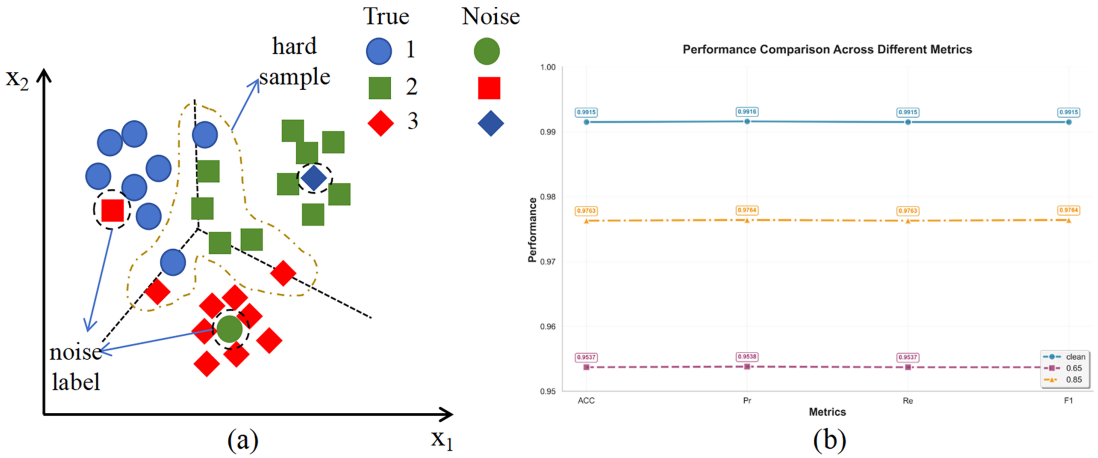

Figure 2: Different distributions of hard samples and noisy labels. Figure (**a**) illustrates the representation of hard samples and noisy labels in the feature space, while Figure (**b**) shows that when using loss values for filtering, adopting either a higher or lower loss threshold will lead to a certain degree of degradation in classification performance.

In reality, network traffic data inherently exhibits "behavioral consistency": traffic of the same type shows high similarity in behavioral patterns. When such traffic is projected into a high-dimensional feature space via deep learning models, it naturally forms tight clustering structures. This provides a theoretical foundation for assessing label reliability from a geometric perspective.

Based on this insight, this paper proposes a noise-tolerant label processing framework based on graph-structured neighbor consistency. The core innovation lies in leveraging the intrinsic structural properties of network traffic in the feature space. By constructing an adaptive K-nearest neighbor graph, we model the behavioral similarity relationships among traffic samples. On this basis, a multi-dimensional consistency evaluation metric is designed to precisely quantify the consistency between each sample's label and its neighbor group, thereby effectively distinguishing between "noisy labels" and "hard samples". Furthermore, a label propagation algorithm is introduced to correct noisy labels. Simulating the "threat intelligence sharing" mechanism in cybersecurity, behaviorally similar traffic nodes can transmit label information through the graph structure. By using high-confidence samples as reliable label anchors, the proposed method guides mislabeled samples' tags to converge toward the consensus of their semantic neighbors, achieving automatic label purification. In essence, this work introduces a graph-based consistency-perception framework that, for the first time, integrates adaptive neighbor graph construction with label propagation to achieve precise noise-hard sample separation and reliable label correction in network traffic detection. The main research objectives are as follows: (1) To propose a noise and hard sample identification mechanism that is independent of training dynamics and instead based on the geometric structure of the feature space, in order to overcome the fundamental limitation of traditional methods in distinguishing between the two; (2) To design an integrated label correction pipeline that organically combines the identification and correction stages, leveraging the intrinsic clustering characteristics of the data and graph propagation mechanisms to achieve reliable correction of identified noisy labels; (3) To validate the effectiveness and robustness of the proposed framework on real and complex network traffic data, particularly under challenging scenarios such as high-ratio noise and asymmetric noise, and to evaluate its practical value in improving the final classification performance of the model.

In summary, the main contributions of this paper are as follows:

(1) Proposes a novel noise identification paradigm based on the geometric structure of the feature space. Unlike traditional methods that heavily rely on training dynamics, this work innovatively leverages the inherent clustering characteristics of network traffic in deep feature spaces to establish an evaluation mechanism based on graph-structured neighbor consistency. This paradigm shift provides a more stable and reliable theoretical foundation for noisy label identification and reduces dependency on prior assumptions about noise distribution.

(2) Designs a refined noise-hard sample differentiation system and an end-to-end correction framework. At the differentiation system level, we designed multi-dimensional consistency evaluation metrics (e.g., neighbor label entropy, self-label support) to achieve precise distinction between simple labeling noise and valuable hard samples located near decision boundaries. This effectively addresses the model performance degradation caused by the conflation of the two in existing methods. At the correction framework level, we organically integrate adaptive K-nearest neighbor graph construction with a label propagation algorithm to form a complete pipeline from noise identification to automatic correction. Simulating a threat intelligence sharing mechanism, this framework utilizes high-confidence samples in the graph as reliable anchors to guide mislabeled samples toward the consensus of their semantic neighbors, achieving data-level self-purification.

(3) Conducts comprehensive validation in real-world complex network scenarios, demonstrating the advancement and practicality of the method. Systematic experiments were performed on real-world datasets such as CICIDS2017 and NSL-KDD, with noise ratios ranging from low to high (10%–90%) and covering both symmetric and asymmetric noise types. The results show that the proposed method significantly outperforms mainstream baselines in terms of noise identification accuracy, label correction purity, and the generalization performance of the final classification model. Particularly under the severe challenges of high-ratio and asymmetric noise, our method exhibits exceptional robustness and stability, providing an effective solution for building reliable traffic detection models adapted to complex real-world network environments.

2 Related Work

The noisy label problem has become a significant challenge in the field of network traffic analysis that cannot be ignored. In real-world network environments, annotating large-scale traffic data requires substantial time and human resources, and label misassignments are often unavoidable. Research by Anderson and McGrew [9] and Shi et al. [10] indicates that in automated annotation systems, ambiguity in port numbers, protocol types, or traffic encryption features can cause some traffic samples to be misclassified, thereby introducing systematic label noise. This misleads the learning direction during model training, resulting in degraded generalization performance of classifiers. Especially in complex models like deep neural networks, due to their strong memorization capacity of training data, they are more prone to overfitting noisy labels, ultimately reducing the stability and reliability of network traffic recognition [11,12]. Therefore, how to effectively identify and correct noisy labels has become a key issue in current intelligent network traffic analysis research.

In recent years, scholars both domestically and internationally have conducted extensive research on the noisy label problem and proposed various identification and correction strategies. Overall, existing methods can be divided into two major categories: data cleaning methods and robust training methods. The former improves training sample purity and credibility through data-level optimization by using clustering analysis, feature consistency detection, or self-supervised learning to identify and correct incorrect labels. The latter focuses on robustness design at the model level, enabling the model to maintain stable learning

performance even in the presence of noisy labels through mechanisms such as improved loss functions, sample re-weighting, contrastive learning, or co-training.

In addition, network traffic detection faces the fundamental challenges of the diverse and evolving nature of security attacks. Modern attacks are increasingly sophisticated and stealthy, placing higher demands on the generalization capability and robustness of detection models. To address this, researchers often adopt strategies such as ensemble learning to enhance the system's coverage of diverse attacks. For example, Laila et al. conducted benchmark tests of multiple ensemble classifiers on diverse network attacks, demonstrating the significant advantages of ensemble methods in terms of detection accuracy, stability, and cross-dataset generalization capability. Their study indicates that by combining feature selection and class balancing strategies, ensemble learning frameworks can effectively address the heterogeneity and stealthiness of modern attacks, providing an empirical foundation for building detection systems adapted to complex threat environments [13]. However, the performance of both basic models and ensemble systems fundamentally depends on the quality of the training data. Noisy labels can contaminate the training process of each base learner, thereby affecting the reliability of the overall system. Therefore, while improving the robustness of model architectures, addressing label quality issues at the data source is crucial for constructing highly reliable network traffic detection systems. The noise-tolerant framework proposed in this paper is designed precisely to address this fundamental problem.

2.1 Data Cleaning Methods for Noisy Label Handling

Data cleaning techniques achieve dataset purification through noise detection and sample filtering, and several mature studies have been established in academia. Wang et al. [14] proposed a lightweight and efficient label error reduction system, MalWhiteout, specifically targeting label noise issues in Android malware detection. It innovatively introduces Confident Learning for noise estimation and combines ensemble learning and adjustments based on application relationships to enhance the robustness and accuracy of noise detection. However, this method heavily relies on external metadata such as developer information, limiting its applicability in network environments where such information is lacking. Medina-Arco et al. [15] proposed a method to detect contaminated training datasets in machine learning-based network intrusion detection systems, determining the optimal training window size and selecting the best-performing data subset through a two-stage process. However, this method relies on fixed time window divisions, making it difficult to adapt to the non-stationary and dynamic characteristics of network traffic. Li et al. [16] proposed the DSDIR method to address the coexistence of noisy labels and long-tailed distributions in malicious traffic detection. It significantly improves model robustness through distribution-aware clean sample selection and dynamic instance re-labeling techniques. However, its two-stage training process is relatively complex and highly sensitive to the quality of the initial clean samples.

Although the aforementioned methods have achieved notable success in their respective scenarios, they either depend on domain-specific prior knowledge, are constrained by fixed data partitioning strategies, or face stability challenges due to complex processes. In contrast, the graph-structured neighbor consistency-based method proposed in this paper offers the following unique advantages: First, it does not rely on any external meta-information, utilizing only the intrinsic structure of the traffic data itself in the feature space. Second, through the construction of an adaptive K-nearest neighbor graph, it naturally adapts to the dynamic changes in network environments. Finally, the integrated identification-correction framework maintains method simplicity while avoiding the potential error accumulation issues associated with complex multi-stage processes. These characteristics enable the proposed method to demonstrate stronger applicability and robustness in addressing complex and variable noisy label problems in network traffic.

2.2 Robust Training Methods for Handling Noisy Labels

Robust training mitigates the interference of erroneous labeling during model training by designing error-tolerant network layers and loss functions, enabling the model to accurately identify malicious traffic on in-distribution data even when trained on low-quality annotated datasets [17,18]. In terms of technical approaches, robust training methods that reduce the impact of mislabeling can be categorized into sample selection-based methods and robust architecture-based methods.

Sample selection methods refer to dynamically selecting clean samples from the dataset during training to adjust network weights. Co-teaching [19] and Co-teaching+ [20] construct parallel training networks consisting of two isomorphic models that mutually select clean samples during training. However, these methods heavily rely on loss values as selection criteria, making it difficult to effectively distinguish between true noisy samples and hard samples with learning value. JoCoR [21] improves upon Co-teaching by regularizing the differences between networks, bringing the predictions of parallel networks closer together. However, the strong correlation between networks may reduce the diversity of selection, leading to the joint memorization of certain noisy samples. In noisy label identification for network traffic, Xu et al. [22] proposed the Differential Training framework, which trains two deep learning models simultaneously and applies multiple outlier detection algorithms for comprehensive judgment. However, this method requires training multiple models and applying various detection algorithms, resulting in high computational costs and limited efficiency for large-scale deployment.

Robust architecture-based methods reduce the impact of erroneous labeling on the training process by designing network architectures with high tolerance. Jiang et al. [23] applied the concept of curriculum learning to noisy learning, guiding the student network's training using a pre-trained teacher network. However, the effectiveness of this method largely depends on the quality of the "clean data" used during the pre-training of the teacher network, a precondition that is often difficult to guarantee in practice. In noisy label identification for network traffic, Liang et al. [24] proposed the FARE framework, which achieves fine-grained attack classification by integrating multiple unsupervised learning algorithms and metric learning. However, the performance of this method heavily depends on the choice of clustering algorithms and is sensitive to hyperparameter settings. Research by Qing et al. [25] found that feature engineering plays a significant role in handling noisy labels in encrypted malicious traffic classification. However, this method requires extensive domain expertise to construct feature sets, limiting its transferability across different network environments.

Although existing sample selection methods have made some progress in handling noise, their core selection mechanisms mostly rely on training dynamics or preset distribution assumptions, making it difficult to fundamentally resolve the confusion between noisy labels and hard samples (Table 1). These methods often misclassify hard samples with high loss values as noise, causing the model to lose its ability to discriminate key boundary samples during training. To address this fundamental challenge, this paper proposes a noise and hard sample identification framework based on graph-structured neighbor consistency. This method breaks through the limitations of traditional reliance on training dynamics and instead leverages the geometric structure of the feature space to characterize semantic relationships among samples by constructing an adaptive K-nearest neighbor graph. Building on this, we design multi-dimensional consistency evaluation metrics that comprehensively consider structural features such as the entropy of neighbor label distributions, self-label support, and neighbor dominant labels. This establishes an evaluation system capable of simultaneously achieving high accuracy in noise identification and preserving the learning value of hard samples.

Table 1: Relevant studies conducted on noise label processing.

Method Name	Type	No External Metadata Dependency	Adaptive to Dynamic Changes	Integrated Framework	Distinguishes Noise from Hard Samples	Utilizes Feature Space Structure
MalWhiteout [14]	Data Cleaning	×	×	×	×	×
Medina-Arco et al. [15]	Data Cleaning	√	×	×	×	×
DSDIR [16]	Data Cleaning	√	×	×	×	×
Co-teaching [19]	Sample Selection	√	×	×	×	×
Co-teaching+ [20]	Sample Selection	√	×	×	×	×
JoCoR [21]	Sample Selection	√	×	×	×	×
Differential Training [22]	Sample Selection	√	×	×	×	×
Curriculum Learning [23]	Robust Architecture	×	×	×	×	×
FARE [24]	Robust Architecture	√	×	×	×	×
NCRT (Our Work)	Graph-Structured Consistency	√	√	√	√	√

3 Methodology

The method proposed in this paper aims to systematically address the label noise problem in network traffic detection. As shown in Fig. 3, the constructed framework consists of two core stages: first, fine-grained identification of easy, noisy, and hard samples based on graph-structured neighbor consistency; then, effective correction of the identified noisy samples using a label propagation algorithm. The entire process does not rely on any clean reference dataset and is capable of autonomously completing the full procedure from noise identification to correction, providing reliable support for building robust traffic detection models.

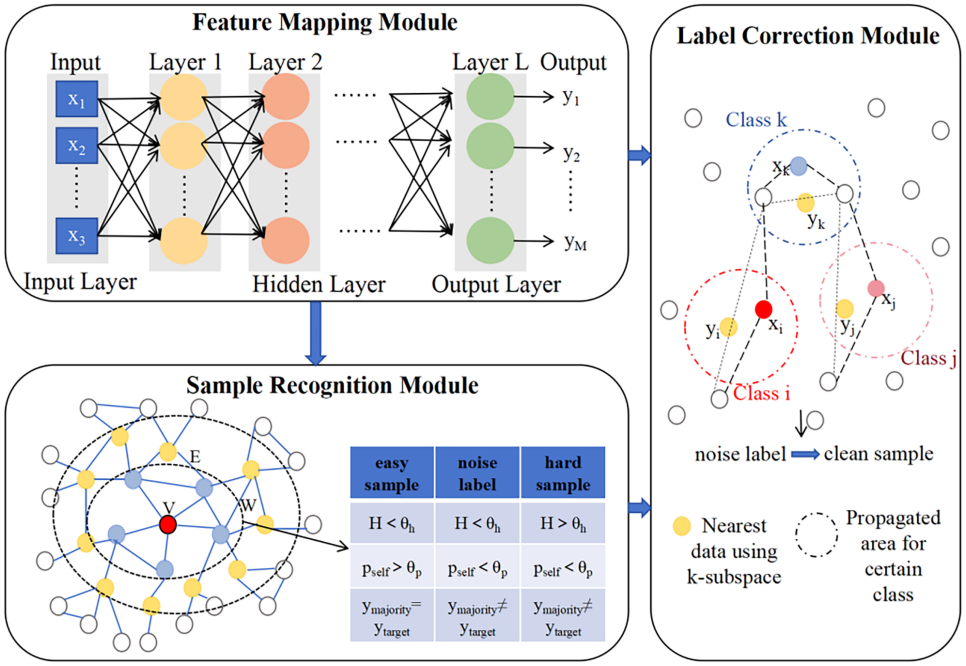

Figure 3: Noise-tolerant label recognition and correction framework.

3.1 Problem Definition

In the task of network traffic detection, we face a typical supervised learning scenario: given a training dataset $D = \{x_i, y_i\}_{i=1}^{N}$ containing N samples, where x_i represents raw network traffic data and $y_i \in \{1, 2, \ldots, C\}$ is its corresponding class label. However, in practical applications, due to factors such as misjudgments by automated analysis tools, inconsistencies in crowd sourced labeling, and the dynamic evolution of network environments, the dataset contains an unknown proportion of noisy labels. That is, the observed label y_i of some samples is inconsistent with their true semantic label $\widetilde{y}_i$.

The core task of this method can be formally defined as follows: under the conditions that the true labels $\widetilde{y}_i$ are unknown and the noise ratio is unknown, by analyzing the intrinsic structure of dataset D in the feature space, achieve the following two objectives:

(1) Identify samples in the training set that may be mislabeled and perform fine-grained classification on them, particularly accurately distinguishing between "noisy samples" and "hard samples".
(2) Perform label correction on the identified noisy samples by assigning appropriate new labels $\widetilde{y}_i$, thereby obtaining a purified training set $\hat{D}$ for training a more robust traffic classification model.

The main challenges of this problem lie in the fact that network traffic data often exhibits complex characteristics such as non-uniform distribution, class imbalance, and dynamic evolution, making traditional noise handling methods based on loss values or training dynamics difficult to adapt. Therefore, there is an urgent need for a solution that can fully leverage the semantic structure of traffic data itself, does not rely on strong assumptions, and can adapt to the data distribution characteristics.

3.2 Noise and Hard Sample Identification Based on Neighbor Consistency

The objective of this stage is to move beyond traditional methods that rely on training loss, and instead, precisely distinguish between "noisy samples" and "hard samples" based on the intrinsic structure of the feature space.

3.2.1 Traffic Representation and Feature Space Construction

To achieve effective noisy label identification, it is first necessary to transform the raw network traffic data into discriminative feature representations. This study employs a feature extractor based on a Multilayer Perceptron (MLP). The network architecture consists of three fully connected layers, with the middle layer using the ReLU activation function, and the final layer outputting a 128-dimensional feature vector.

Specifically, each raw network traffic sample $\mathrm{x_i}$ is mapped to a feature vector $v_i = f_\theta(x_i) \in \mathrm{R}^{128}$ through this feature extractor $f_\theta(\cdot)$. This simple network structure not only ensures computational efficiency but also effectively learns nonlinear relationships within the traffic data, mapping semantically similar traffic samples to adjacent positions in the feature space.

The set of feature vectors from all samples $\{v_1, v_2, \ldots, v_N\}$ forms the feature space $V \in \mathrm{R}^{128}$. Within this space, cosine similarity is adopted as the similarity metric, calculated as follows:

$$sim(v_i, v_j) = \frac{v_i \cdot v_j}{||v_i||||v_j||} \tag{1}$$

This metric is insensitive to scale variations in feature vectors and focuses on measuring the directional similarity of traffic behavior patterns, providing a reliable similarity calculation foundation for the subsequent graph structure construction.

3.2.2 Adaptive K-Nearest Neighbor Graph Construction

To accurately model the local similarity structure of network traffic samples in the feature space, an adaptive K-nearest neighbor graph is constructed. Considering that network traffic data typically exhibits an uneven distribution—where sample counts for different malware families or application types may vary significantly—traditional methods with a fixed K value struggle to adapt to such complex data distributions. Therefore, an adaptive K-value determination strategy based on local density is proposed, which dynamically calculates the optimal number of neighbors for each sample.

For a sample x_i, its adaptive K-value K_i is computed as follows: First, the distance d_i^m from x_i to its m-th nearest neighbor is calculated, where m is a preset initial value. Next, the number of samples within the hyper-sphere centered at x_i with radius d_i^m is counted, yielding the local density estimate ρ_i. Finally, the adaptive K-value is obtained using the formula $K_i = [K_{base} \cdot \frac{\rho_i}{\rho_{avg}}]$, where K_{base} is the base neighbor number and ρ_{avg} is the average local density of all samples. This design ensures that smaller K-values are used in sparse sample regions to avoid introducing irrelevant distant neighbors, while larger K-values are employed in dense sample regions to fully leverage rich local contextual information. Assume the preset value is $m = 10$. The distance from sample x_i to its 10th nearest neighbor is calculated as $d_i10 = 0.5$. Within the hypersphere centered at x_i with a radius of 0.5, a total of 18 samples (including x_i itself) are counted, giving a local density estimate $\rho_i = 18$. If the average local density of all samples is $\rho_{avg} = 12$ and the base neighbor number is $K_{base} = 15$, the adaptive K-value is computed as $K_i = 15 \times (18/12) = 22.5 = 22$. This value is greater than K_{base}, reflecting that x_i is in a relatively dense region and thus requires more neighbors to characterize its context.

After determining the adaptive K-value for each sample, an undirected weighted graph G = (V, E, W) is constructed to formally represent the similarity relationships among samples. The node set V corresponds to all traffic samples, with each node carrying its corresponding feature vector v_i. The edge set E is constructed following the mutual neighbor principle: an edge e_{ij} exists between nodes i and j if and only if j belongs to the K_i-nearest neighbor set of i, or i belongs to the K_j-nearest neighbor set of j. This symmetric design ensures the connectivity of the graph and prevents information propagation interruptions caused by unidirectional neighbor relationships.

The edge weight matrix W is computed using a Gaussian kernel function, specifically defined as $w_{ij} = \exp\left(-\frac{\|v_i - v_j\|^2}{2\sigma^2}\right)$, where σ is a scale parameter controlling the decay rate of similarity. This calculation is based on the Euclidean distance in the feature space, effectively capturing the local similarity between samples: when the feature vectors of two samples are closer, the corresponding edge weight approaches 1, indicating high similarity; conversely, as the distance between samples increases, the edge weight decays exponentially toward 0, indicating low similarity. Through this soft-threshold design, the primary similarity relationships are preserved while reducing the influence of irrelevant samples, providing a precise similarity measurement foundation for subsequent label propagation.

3.2.3 Sample Type Determination Based on Neighbor Consistency Evaluation Metrics

For each target sample x_{target} (with label y_{target}), the label distribution of its K nearest neighbors $\mathcal{N}_K$ is analyzed, and the following three core metrics are computed:

- Neighbor Label Consistency (Normalized Entropy): This metric measures the confusion level of the label distribution among neighbor nodes. Let n_j denote the number of neighbor samples belonging to class j. The label distribution probability is $p_j = n_j/K$. The normalized entropy H_{norm} is calculated as follows:

$$H = -\sum_{j=1}^{C} p_j \log p_j, H_{norm} = \frac{H}{\log C} \quad (2)$$

where C is the total number of classes. $H_{norm} \in [0, 1]$. A value closer to 0 indicates higher consistency among neighbor labels, while a value closer to 1 indicates greater confusion in neighbor labels.

- Consistency with the Central Node Label (p_{self}): This metric measures the degree of support the target sample's own label receives from its neighbors.

$$p_{self} = \frac{n_{y_{target}}}{K} \tag{3}$$

- $p_{self} \in [0, 1]$. A higher value indicates that the target sample's label aligns better with its behavioral neighbors.
- Neighbor Majority Label ($y_{majority}$): This metric identifies the most frequent label among the neighbors.

$$y_{majority} = arg \max_{j} n_j \tag{4}$$

Its corresponding proportion $y_{majority}$ reflects the strength of the dominant opinion within the neighbor group.

Based on the above metrics, samples are classified into three categories using preset thresholds θ_h (for H_{norm}) and θ_p (for p_{self}) (Table 2):

- Clean Sample: Neighbor labels are consistent and support the sample's own label, representing high-quality samples.
- Noisy Label: Neighbor labels are consistent, but the consensus label differs from the sample's own label, indicating a high likelihood of mislabeling.
- Hard Sample: Neighbor labels are confused, and the sample's own label lacks support. These samples may lie near class boundaries and represent valuable instances of novel or rare attack patterns, which should be preserved.

Table 2: Evaluation metrics for distinguishing sample types.

Sample Category	Neighbor Label Consistency	Consistency with Central Node Label	Neighbor Majority Label
Clean Sample	$H < \theta_h$	$p_{self} > \theta_p$	$y_{majority} = y_{target}$
Noisy Label	$H < \theta_h$	$p_{self} < \theta_p$	$y_{majority} \neq y_{target}$
Hard Sample	$H > \theta_h$	$p_{self} < \theta_p$	$y_{majority} \neq y_{target}$

The Algorithm 1 for identifying noisy and hard samples is as follows:

Algorithm 1: Neighbor consistency evaluation and sample classification

Input: Feature vector set $\{v_i\}_{i=1}^{N}$, initial labels $\{y_i\}_{i=1}^{N}$, base neighbor number Kbase, entropy threshold θ_h, support threshold θ_p

Output: Sample categorization results $\{Category_i\}_{i=1}^{N}$

1: for $i = 1$ to N do

2: Compute K_i via the adaptive K-value determination strategy

3: Find $N_K(i) \leftarrow$ the K_i nearest neighbors of v_i

4: Compute neighbor label distribution $\{n_j\}$ and probabilities $p_j = n_j/K_i$

(Continued)

Algorithm 1 (continued)

Compute normalized entropy $H_{norm} = \frac{-\sum_{j=1}^{C} p_j \log p_j}{\log C}$
Compute self-support score $p_{self}(i) = n_{y_i}/K_i$
Determine majority label $y_{majority}(i) = arg\max_j n_j$
if $H_{norm}(i) < \theta_h$ and $p_{self}(i) > \theta_p$ and $y_{majority}(i) = y_i$ then
    $Category_i \leftarrow$ "easy sample"
else if $H_{norm}(i) < \theta_h$ and $p_{self}(i) < \theta_p$ and $y_{majority}(i) \neq y_i$ then
    $Category_i \leftarrow$ "noisy lable"
else if $H_{norm}(i) > \theta_h$ and $p_{self}(i) < \theta_p$ and $y_{majority}(i) \neq y_i$ then
    $Category_i \leftarrow$ "hard sample"
end if
end for

3.3 Noise Label Correction Based on Label Propagation

After identifying "noisy labels" and "hard samples", this stage utilizes the constructed K-nearest neighbor graph and employs a label propagation algorithm to assign appropriate new labels to the noisy labels.

3.3.1 Algorithm Initialization

Probability Transition Matrix P: The previously constructed adjacency matrix W is row-normalized to obtain the transition matrix P, where $P_{ij} = w_{ij}/\sum_k w_{ik}$. P_{ij} represents the probability of label propagation from node j to node i.

Soft Label Matrix F: An initial N × C soft label matrix $F^{(0)}$ is defined. For samples already identified as "clean samples" and "hard samples", their labels are fixed in one-hot encoding form. For noisy samples awaiting correction, their initial soft labels can be randomly initialized or uniformly distributed.

3.3.2 Iterative Propagation

Label propagation proceeds through the following iterative process:

Propagation Step: $F^{(t)} = P \times F^{(t-1)}$. At each step, every node receives label information from its neighbors, with the weight determined by the edge weights (similarity).

Anchor Reset Step: The soft labels of all clean samples ("clean samples" and "hard samples") in $F^{(t)}$ are clamped back to their original one-hot encodings. This step is crucial, as it ensures that the labels of reliable samples act as "beacons" or "anchors" during propagation and are not contaminated by noise.

Repeat Steps 1 and 2 until the soft label matrix F converges (i.e., the change between consecutive iterations falls below a predefined threshold) or the maximum number of iterations is reached.

3.3.3 Final Label Determination

After the iterations conclude, for each noisy sample i, the class with the highest probability in its converged soft label vector F_i is selected as its corrected new label:

$$\widehat{y_i} = arg\max_j F_{ij} \tag{5}$$

Through this process, the labels of noisy samples are corrected to align with the consensus of their most behaviorally similar neighbors in the feature space, thereby achieving label purification. The advantage of

this method lies in its natural alignment of decision boundaries with low-density regions of the feature space, adhering to the cluster assumption and consequently enhancing the overall generalization capability and robustness of the model. The noise label correction process based on label propagation is outlined below (Algorithm 2):

Algorithm 2: Label propagation based noise correction

Input: Graph G = (V, E, W), sample categorization results, initial labels $\{y_i\}_{i=1}^{N}$, maximum iterations T_{max}, convergence threshold ϵ

Output: Corrected labels $\{\widehat{y_i}\}_{i=1}^{N}$

Construct transition matrix P, where $P_{ij} = W_{ij} / \sum_{k=1}^{N} W_{ik}$

Initialize soft label matrix $F^{(0)} \in \mathbb{R}^{N \times C}$

for $i = 1$ to N do

  if $Category_i \in$ {"easy sample", "hard sample"} then

    $F_i^{(0)} \leftarrow$ one-hot encoding of y_i

  else

    $F_i^{(0)} \leftarrow$ random initialization

  end if

end for

for t = 1 to T_{max} do

  Propagate: $F^{(t)} = P \times F^{(t-1)}$

  Reset Anchors:

  for $i = 1$ to N do

    if $Category_i \in$ {"easy sample","hard sample"} then

      $F_i^{(t)} \leftarrow$ one-hot encoding of y_i

    end if

  end for

  if $\left\| F^{(t)} - F^{(t-1)} \right\|_F < \epsilon$ then

    break

  end if

  end if

  for $i = 1$ to N do

    $\widehat{y_i} = arg\max_j F_{ij}^{(t)}$

  end for

4 Experimental Verification

4.1 Experimental Setup

Datasets and Noise Settings: Two representative benchmark datasets for network traffic detection were selected: CICIDS2017 and NSL-KDD. The CICIDS2017 dataset contains real traffic records of various attack types in modern network environments, covering common threats such as DDoS attacks. As a classic intrusion detection dataset, NSL-KDD provides rich network connection features and annotation information.

To simulate label noise scenarios in real network environments, and with reference to the settings of works such as MCRe [26] and ULDC [27], two different types of noise injection mechanisms were designed: Asymmetric noisy labels simulate the systematic bias in multi-party collaborative labeling. Due to business continuity considerations, the system tends to classify suspicious traffic as benign, resulting in a significantly

higher probability of malicious samples being mislabeled as benign compared to the reverse mislabeling. Under this scenario, the reliability of labels for the malicious class decreases. Symmetric noisy labels simulate random errors by a single annotator, where mislabeling has no directionality. Fig. 4 shows the distribution characteristics of the two types of noise when the noise ratio is 40% (Fig. 4).

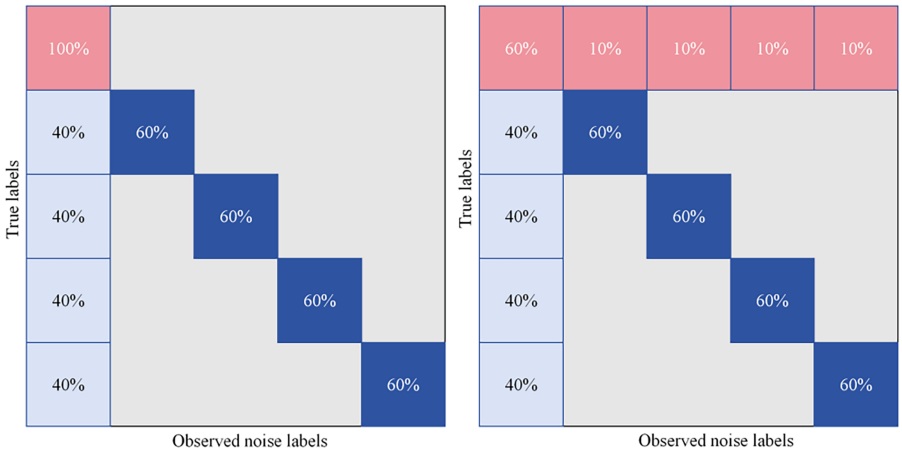

Figure 4: Detailed settings for noise scenarios.

Baseline Models: To comprehensively evaluate the effectiveness of the proposed method, five representative noise handling methods were selected as baseline models:

- INCV [28]: An iterative training strategy that progressively filters out high-confidence samples during training to construct a cleaned dataset and retrain the supervised model.
- CL [29]: A noise identification method based on confident learning, which estimates the noise transition matrix and joint feature distribution to identify potentially mislabeled samples.
- FINE [30]: Detects noisy samples using feature distribution modeling and neighbor consistency analysis, and trains a classifier based on the cleaned data.
- Co-Teaching: Simultaneously trains two neural networks with different initializations, allowing them to mutually select potentially clean samples for each other's parameter updates during training.

4.2 Experimental Analysis of Noisy Label Correction

After identifying potential noisy labels, precise correction is a crucial step for improving the quality of the training set and, consequently, enhancing model performance. To quantitatively evaluate the effectiveness of each method in the correction phase, the clean sample rate across all data was calculated, directly reflecting the accuracy and reliability of the label correction process.

On the CICIDS-2017 dataset, the correction performance of the proposed method was compared against baseline models such as INCV, CL, FINE, and ULDC under symmetric and asymmetric noise types, across different noise ratios (ranging from 10% to 90%). The experimental results are shown in Fig. 5:

(1) Overall Leading Performance: Under the vast majority of noise settings, the clean sample rate achieved by our proposed method, NCRT, significantly outperforms all baseline methods. This advantage is particularly pronounced in challenging scenarios with high noise ratios (≥40%). For example, at a symmetric noise ratio of 50%, NCRT achieved a clean sample rate of 94.55%, while the second-best performer, ULDC, reached 85.95%. Methods like INCV, CL, and FINE all fell below 70%. This indicates that NCRT's correction mechanism possesses stronger robustness, effectively correcting errors even when the dataset contains a large number of incorrect labels.

(2) Robustness to High Noise Ratios: As the injected noise ratio increases, the clean sample rates of all baseline methods show a significant downward trend. For instance, under asymmetric noise, when ratio increases from 10% to 90%, the clean sample rates of INCV, CL, and FINE plummet from approximately 98% to about 41%, 51%, and 34%, respectively. In stark contrast, the NCRT method demonstrates exceptional stability. Under symmetric noise, even with a noise ratio as high as 90%, NCRT maintains a clean sample rate of 87.42%; under 90% asymmetric noise, it achieves 88.44%, with a decline far smaller than that of baseline methods. This proves that the correction strategy based on graph-structured neighbor consistency and label propagation can better resist the interference of large-scale noise, and its correction capability does not degrade sharply with increasing noise density.

(3) Effectiveness Against Asymmetric Noise: Asymmetric noise, due to its directional labeling bias, is typically more difficult to correct than completely random symmetric noise. The experimental results confirm this, as most baselines perform slightly worse in asymmetric scenarios compared to symmetric ones. However, NCRT maintains leading and stable high performance under both noise types. For example, at ratio = 50%, NCRT achieves clean sample rates of 94.55% and 84.71% under symmetric and asymmetric noise, respectively, both significantly outperforming the best baseline at the same noise level. This is attributed to the core design of NCRT: it identifies systematic labeling errors by analyzing the local neighbor consensus of samples in the feature space. When a certain class of samples is systematically mislabeled as another class, their "true similar" neighbors in the feature space form a strong label consistency signal, which is accurately captured and utilized by the algorithm for correction.

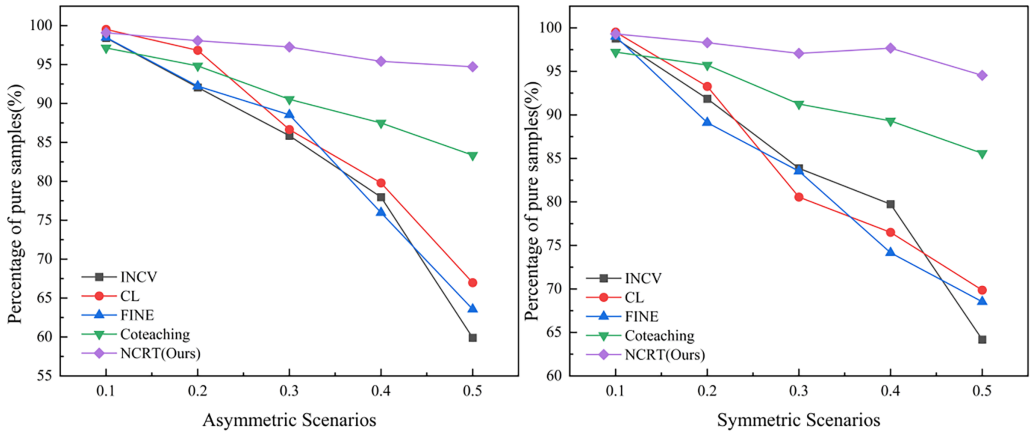

Figure 5: Percentage of pure samples under symmetric and asymmetric scenarios.

4.3 Experimental Analysis of Detection Performance on Test Samples

To ultimately evaluate the improvement in model generalization ability brought by each noise handling method, classifiers were retrained on datasets processed by different methods, and their classification accuracy (Acc%) was evaluated on a clean test set. This experiment aims to address the core question: After noise processing, what is the model's detection capability in real, noise-free scenarios?

We compared the proposed method (NCRT/Ours) with mainstream baselines such as INCV, CL, FINE, and Co-Teaching on the CICIDS-2017 and NSL-KDD datasets. The experimental results (Fig. 6) clearly illustrate the impact of different methods on the final model performance under various types and proportions of noise.

(1) Overwhelming Performance Advantage: The proposed NCRT framework achieved the highest test accuracy across all experimental settings, with an extremely significant advantage. Under symmetric noise scenarios, NCRT's accuracy remained stable between 96.65% and 97.74%; under asymmetric noise, it also maintained a high level between 92.46% and 96.68%. In contrast, the best baseline, ULDC, achieved an accuracy of 68.60% under 50% symmetric noise and 81.34% under 50% asymmetric noise, with gaps of approximately 28 and 11 percentage points compared to NCRT, respectively. This fully demonstrates that the model trained on data processed by NCRT exhibits far superior generalization capabilities compared to existing methods.

(2) Exceptional Robustness and Stability: As the noise ratio increased from 10% to 50%, the performance of all baseline models showed a sharp decline. For example, INCV's accuracy under symmetric noise dropped sharply from 87.31% to 49.23%, while CL's accuracy under asymmetric noise decreased from 91.63% to 51.43%. This indicates that these methods nearly fail when noise exceeds half. In contrast, NCRT's performance curve remained exceptionally stable. Under symmetric noise, the fluctuation in accuracy was less than 1.1%; under asymmetric noise, the maximum decline was only about 4.2%. This stability, which is highly insensitive to noise intensity, demonstrates that NCRT possesses strong anti-interference capabilities, ensuring that the model can learn reliable feature representations even under adverse labeling conditions.

(3) Effectiveness in Handling Complex Asymmetric Noise: Asymmetric noise, due to its systematic bias, poses a stronger interference to model learning and is generally considered a more severe challenge. The experimental results show that most baselines (e.g., INCV, CL, FINE) experienced a faster performance degradation under asymmetric noise compared to symmetric noise. However, NCRT performed excellently and stably under both noise types. Particularly under high proportions of asymmetric noise (40%, 50%), NCRT (92.73%, 92.46%) maintained an advantage of over 10 percentage points compared to the best-performing baseline, ULDC (81.94%, 81.34%). This validates that NCRT's strategy of identifying systematic labeling errors based on neighbor consistency and performing correction through graph propagation is particularly effective in solving directional noise problems.

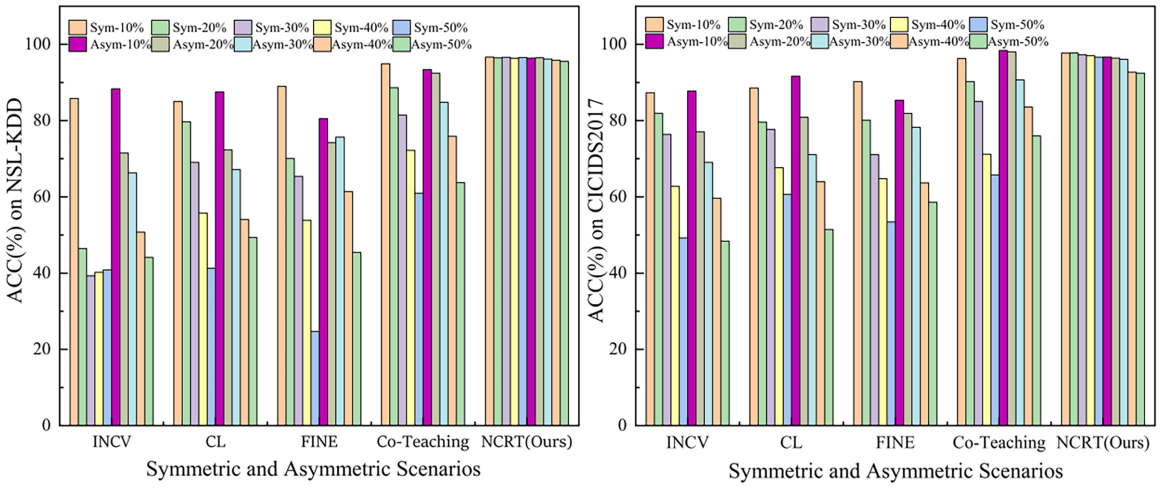

Figure 6: Classification Performance on the CICIDS2017 and NSL-KDD Datasets.

4.4 Discussion: Verification of Noise Tolerance Capability

The proposed NCRT framework aims to achieve a core objective: to exhibit a high degree of tolerance to label noise in training data for network traffic detection. This entails the ability to accurately identify noisy

labels, effectively correct them, and simultaneously avoid misclassifying valuable hard samples, ultimately enhancing the generalization performance of the model on a clean test set. Our experimental design and results systematically demonstrate the successful achievement of this goal from the following three interconnected dimensions:

First, in the noise identification phase, the framework demonstrates high precision and the ability to distinguish hard samples. This is the prerequisite for "tolerating" rather than simply "filtering out" noise. As shown in Table 2 (Sample Type Determination Criteria) and Algorithm 1, our decision-making is based on multi-dimensional neighbor consistency metrics (normalized entropy H_{norm}, self-label support p_{self}), neighbor majority label ($y_{majority}$). Experimental results indicate that this strategy effectively distinguishes between two types of samples with high loss values: samples where "neighbor labels are consistent but its own label is the minority" are identified as noise, while samples where "neighbor labels themselves are confused" are identified as hard samples and retained. In Fig. 6, even under a high noise ratio of 50%, the NCRT model maintains a test accuracy exceeding 90%. This indirectly yet strongly suggests that crucial boundary samples (hard samples) in the training data were not erroneously removed, allowing the model to learn discriminative features from them. In contrast, baseline methods that heavily rely on loss-based filtering experience a sharp performance drop under the same noise conditions, precisely because they cannot make this distinction, leading to "malnourishment" of the model.

Second, in the label correction phase, the framework reliably corrects identified noisy labels towards the consensus direction of their semantic neighbors through a graph propagation mechanism. This is the key action to achieve "tolerance" and improve data quality. The "clean sample rate" provided in Fig. 5 directly measures the quality of the corrected data. Across a wide range of noise ratios from 10% to 90%, NCRT's clean sample rate consistently and significantly surpasses all baselines. It exhibits exceptional robustness under high noise (≥40%) and the more challenging asymmetric noise settings. For example, under 50% symmetric noise, NCRT maintains a clean sample rate of 87.42%, while most baselines drop below 50% (Fig. 5). This result directly proves that the label propagation algorithm based on the adaptive K-nearest neighbor graph successfully guides the labels of most mislabeled samples to their correct classes by leveraging the local cluster structure in the feature space, achieving effective noise correction rather than mere identification.

Third, and ultimately, the data processed by the framework enables the training of a significantly more robust classification model with superior generalization capability. This is the ultimate manifestation and final validation of the "noise tolerance" capability. The comprehensive performance comparison in Fig. 6 provides the most direct evidence. On the CICIDS2017 and NSL-KDD datasets, models trained on data corrected by NCRT achieve leading classification accuracy across all noise types and ratios. Crucially, its performance curve remains remarkably stable as noise increases. For instance, under symmetric noise, when the noise ratio increases from 10% to 50%, the fluctuation in accuracy is less than 10%. This extremely low sensitivity to noise intensity is a hallmark of "high tolerance". It indicates that our framework not only cleanses the data but also, by preserving hard samples and performing corrections based on geometric structure, helps the model learn an intrinsic feature representation that is robust against noise interference.

5 Conclusion and Outlook

Network traffic detection, as a core component of cybersecurity, relies heavily on high-quality annotated data for its performance. However, the widespread issue of noisy labels in real-world environments severely limits the reliability and generalization capability of models. This leads to fragile model performance when confronting continuously evolving network threats. To address this fundamental challenge, this paper proposes a noise label identification and correction framework based on graph-structured neighbor consistency. The framework innovatively leverages the inherent clustering characteristics of network traffic in the

feature space. By constructing an adaptive K-nearest neighbor graph to model behavioral similarity among samples, and designing multi-dimensional consistency evaluation metrics, it achieves precise differentiation between noisy labels and hard samples. Furthermore, a label propagation algorithm simulates a threat intelligence sharing mechanism, utilizing reliable anchor samples in the graph to guide the correction of noisy labels toward the consensus direction of their semantic neighbors. Systematic experiments on real-world datasets such as CICIDS2017 and NSL-KDD demonstrate that our method significantly outperforms mainstream baselines in terms of noise identification, label correction, and final classification performance. The framework proposed in this paper is based on a core observation: network traffic tends to exhibit semantic clustering characteristics in the feature space, which provides the theoretical foundation for our graph-structured neighbor consistency-based method. However, we recognize that the universality of this assumption faces several boundary conditions in practice. First, in high-dimensional sparse feature representations, distance metrics may become ineffective, thereby affecting the reliability of neighbor graph construction. This study aims to mitigate this issue by designing a dimensionality-reducing feature extractor and employing scale-insensitive cosine similarity. Second, in scenarios of extreme class imbalance, minority classes may fail to form dense clusters, causing neighbor relationships to be dominated by majority classes. Although the adaptive K-value strategy employed in this paper can reduce the number of neighbors in sparse regions, thereby partially alleviating the impact, handling extremely rare categories still requires further research. Finally, the validity of the clustering assumption fundamentally depends on the representational capability of the feature extractor. The MLP-based feature extractor adopted in this paper demonstrated good discriminative performance on the experimental datasets, but its generalization ability to more complex or heterogeneous traffic patterns remains to be explored.

Looking ahead, we plan to deepen this research in the following directions: first, to explore more robust feature learning and graph construction methods to enhance the adaptability of the approach in high-dimensional sparse and extremely imbalanced scenarios; second, to study the modeling of dynamic and mixed noise patterns to address more realistic annotation environments; and third, to optimize the computational and storage efficiency of the framework by exploring lightweight graph algorithms and online processing mechanisms to support large-scale deployment.

Acknowledgement: The authors would like to thank the support and help from the People's Armed Police Force of China Engineering University, College of Information Engineering subject group, which funded this work under the All-Army Military Theory Research Project, Armed Police Force Military Theory Research Project. The language polishing of this paper was assisted by the GPT-5 artificial intelligence model.

Funding Statement: The authors would like to thank the support and help from the People's Armed Police Force of China Engineering University, College of Information Engineering subject group, which funded this work under the All-Army Military Theory Research Project, Armed Police Force Military Theory Research Project (WJJY22JL0498).

Author Contributions: The authors confirm contribution to the paper as follows: conceptualization, methodology, Yuheng Gu; formal analysis, investigation, Yuheng Gu, Yu Yang; writing—original draft preparation, Yuheng Gu, Zhuoyun Yang; writing—review and editing, Yuheng Gu, Shangjun Wu; supervision, resources, funding acquisition, Yu Yang. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author, [Yuheng Gu], upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Shen M, Liu Y, Zhu L, Xu K, Du X, Guizani N. Optimizing feature selection for efficient encrypted traffic classification: a systematic approach. IEEE Netw. 2020;34(4):20–7. doi:10.1109/MNET.011.1900366.
2. Chen M, Wang X, He M, Jin L, Javeed K, Wang X. A network traffic classification model based on metric learning. Comput Mater Contin. 2020;64(2):941–59. doi:10.32604/cmc.2020.09802.
3. Dong W, Yu J, Lin X, Gou G, Xiong G. Deep learning and pre-training technology for encrypted traffic classification: a comprehensive review. Neurocomputing. 2025;617(5):128444. doi:10.1016/j.neucom.2024.128444.
4. Lin X, Xiong G, Gou G, Li Z, Shi J, Yu J. ET-BERT: a contextualized datagram representation with pre-training transformers for encrypted traffic classification. In: Proceedings of the ACM Web Conference 2022; 2022 Apr 25–29; Virtual. p. 633–42.
5. Shen M, Zhang J, Zhu L, Xu K, Du X. Accurate decentralized application identification via encrypted traffic analysis using graph neural networks. IEEE Trans Inf Forensics Secur. 2021;16:2367–80. doi:10.1109/TIFS.2021.3050608.
6. Shen M, Ye K, Liu X, Zhu L, Kang J, Yu S, et al. Machine learning-powered encrypted network traffic analysis: a comprehensive survey. IEEE Commun Surv Tutor. 2023;25(1):791–824. doi:10.1109/COMST.2022.3208196.
7. Zhao Y, Xia Q, Sun Y, Wen Z, Ma L, Ying S. Open set label noise learning with robust sample selection and margin-guided module. Knowl Based Syst. 2025;321:113670. doi:10.1016/j.knosys.2025.113670.
8. Pan W, Wei W, Zhu F, Deng Y. Enhanced sample selection with confidence tracking: identifying correctly labeled yet hard-to-learn samples in noisy data. Proc AAAI Conf Artif Intell. 2025;39(19):19795–803. doi:10.1609/aaai.v39i19.34180.
9. Anderson B, McGrew D. Machine learning for encrypted malware traffic classification: accounting for noisy labels and non-stationarity. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2017 Aug 13–17; Halifax, NS, Canada. p. 1723–32.
10. Shi S, Guo Y, Wang D, Zhu Y, Han Z. Distributionally robust federated learning for network traffic classification with noisy labels. IEEE Trans Mob Comput. 2024;23(5):6212–26. doi:10.1109/TMC.2023.3319657.
11. Luo Y, Yang G. Enhancing robustness of deep networks against noisy labels based on a two-phase formulation of their learning behavior. In: Proceedings of the 2023 IEEE International Conference on Multimedia and Expo (ICME); 2023 Jul 10–14; Brisbane, Australia. p. 1763–8.
12. Zhang Q, Zhu Y, Yang M, Jin G, Zhu Y, Chen Q. Cross-to-merge training with class balance strategy for learning with noisy labels. Expert Syst Appl. 2024;249(5):123846. doi:10.1016/j.eswa.2024.123846.
13. Laila DA, Aljawarneh M, Al-Na'amneh Q, Sulaiman RB. Optimizing intrusion detection systems through benchmarking of ensemble classifiers on diverse network attacks. STAP J Secur Risk Manag. 2025;2025(1):71–84. doi:10.63180/jsrm.thestap.2025.1.4.
14. Wang L, Wang H, Luo X, Sui Y. MalWhiteout: reducing label errors in Android malware detection. In: Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering; 2022 Oct 10–14; Rochester, MI, USA. p. 1–13.
15. Medina-Arco JG, Magán-Carrión R, Rodríguez-Gómez RA, García-Teodoro P. Methodology for the detection of contaminated training datasets for machine learning-based network intrusion-detection systems. Sensors. 2024;24(2):479. doi:10.3390/s24020479.
16. Li G, Zhang R, Sun Z, Xing L, Zhang Y. DSDIR: a two-stage method for addressing noisy long-tailed problems in malicious traffic detection. In: ICASSP 2025—2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2025 Apr 6–11; Hyderabad, India. p. 1–5.
17. Dong C, Liu L, Shang J. Label noise in adversarial training: a novel perspective to study robust overfitting. arXiv:2110.03135. 2021.
18. Meng Q, Yuan Q, Tao J, Wang P, Lu S, Li G, et al. When unknown threat meets label noise: a self-correcting framework. IEEE Trans Dependable Secure Comput. 2026;23(1):905–22. doi:10.1109/tdsc.2025.3611908.
19. Han B, Yao Q, Yu X, Niu G, Xu M, Hu W, et al. Co-teaching: robust training of deep neural networks with extremely noisy labels. Adv Neural Inf Process Syst. 2018;31:1–11. doi:10.2139/ssrn.4911734.

20. Yu X, Han B, Yao J, Niu G, Tsang IW, Sugiyama M. How does disagreement help generalization against label corruption? In: Proceedings of the 36th International Conference on Machine Learning; 2019 Jun 9–15; Long Beach, CA, USA.
21. Wei H, Feng L, Chen X, An B. Combating noisy labels by agreement: a joint training method with co-regularization. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13–19; Seattle, WA, USA. p. 13723–32.
22. Xu J, Li Y, Deng RH. Differential training: a generic framework to reduce label noises for Android malware detection. In: Proceedings of the 2021 Network and Distributed System Security Symposium; 2021 Feb 21–25; Virtual.
23. Jiang L, Zhou Z, Leung T, Li LJ, Li FF. MentorNet: learning data-driven curriculum for very deep neural networks on corrupted labels. In: Proceedings of the 35th International Conference on Machine Learning; 2018 Jul 10–15; Stockholm, Sweden.
24. Liang J, Guo W, Luo T, Honavar V, Wang G, Xing X. FARE: enabling fine-grained attack categorization under low-quality labeled data. In: Proceedings 2021 Network and Distributed System Security Symposium; 2021 Feb 21–25; Virtual.
25. Qing Y, Yin Q, Deng X, Chen Y, Liu Z, Sun K, et al. Low-quality training data only? A robust framework for detecting encrypted malicious network traffic. arXiv:2309.04798. 2023.
26. Yuan Q, Gou G, Zhu Y, Zhu Y, Xiong G, Wang Y. MCRe: a unified framework for handling malicious traffic with noise labels based on multidimensional constraint representation. IEEE Trans Inf Forensics Secur. 2024;19(6):133–47. doi:10.1109/TIFS.2023.3318962.
27. Yuan Q, Zhu Y, Xiong G, Wang Y, Yu W, Lu B, et al. ULDC: unsupervised learning-based data cleaning for malicious traffic with high noise. Comput J. 2024;67(3):976–87. doi:10.1093/comjnl/bxad036.
28. Chen P, Liao B, Chen G, Zhang S. Understanding and utilizing deep neural networks trained with noisy labels. arXiv:1905.05040. 2019.
29. Northcutt C, Jiang L, Chuang I. Confident learning: estimating uncertainty in dataset labels. J Artif Intell Res. 2021;70:1373–411. doi:10.1613/jair.1.12125.
30. Kim T, Ko J, Choi J, Yun SY. Fine samples for learning with noisy labels. Adv Neural Inf Process Syst. 2021;34:24137–49. doi:10.1587/transinf.2021edp7127.