electronics

MDPI

Article

Dynamic Distillation-Aided Federated Learning for Intrusion Detection in Heterogeneous Edge Networks

Fan Wang and Weimin Chen *

School of Information and Electronic Engineering, Hunan City University, Yiyang 413000, China; wangf20250901@163.com

* Correspondence: chenweimin@hncu.edu.cn; Tel.: +86-133-2737-2255

Abstract

Intrusion detection serves as a core technology for securing heterogeneous edge networks, including IoT, industrial edges, and 5G networks. However, existing federated learning-based intrusion detection systems suffer from environmental heterogeneity, limited sample availability, and severe class imbalance—issues that result in inefficient resource allocation and compromised detection performance against rare attacks. In this paper, we propose a novel lightweight intrusion detection model for heterogeneous edge networks, named FedNIDS-CNN, which is based on dynamic distillation-aided federated learning with a CNN backbone. In the data preprocessing phase, a two-level class balancing strategy integrating nearest-neighbor interpolation augmentation and adaptive synthetic sampling is employed to ensure distortion-free sample synthesis. For feature and model optimization, principal component analysis (PCA) is used to reduce the dimensionality of traffic features, while a lightweight 1D-CNN is adopted as the base model to alleviate computational overhead on edge devices. During federated training and knowledge aggregation, a dynamic weight distillation loss mechanism is designed to enhance the model's ability to recognize minority-class attacks. Meanwhile, the federated framework supports client-side local training and server-side weighted soft-label aggregation, enabling effective knowledge fusion across heterogeneous models. Experimental results on the CICIDS2017 dataset demonstrate that the proposed method achieves an accuracy of 98.55% and an F1-score of 98.40%. Benefiting from the soft-label transmission and parameter-free aggregation design, the framework gets rid of the constraint of homogeneous model architecture and natively supports heterogeneous network models and edge devices with different computing capabilities. It also significantly reduces communication traffic and per-round training latency, confirming its excellent real-time performance and applicability in resource-constrained edge environments.

Keywords: network intrusion detection; knowledge distillation; federated learning; convolutional neural network; principal component analysis

Check for updates

Academic Editor: Andreas Mauthe

Received: 14 May 2026
Revised: 12 June 2026
Accepted: 18 June 2026
Published: 21 June 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

With the rapid development of the Internet of Things (IoT) and edge computing, a growing number of computational tasks are being offloaded to the edge, which is close to the data source. According to Shi et al. [1], who proposed this prediction, by 2025, more than 150 billion edge devices will be deployed worldwide, and over 70% of IoT data will be processed locally at the edge rather than being uploaded to the cloud. While edge

computing offers low latency and high availability, its open, fragmented, and heterogeneous nature renders centralized intrusion detection systems (IDS) ineffective.

The fragmented distribution of edge data further exacerbates security challenges. Although independently deployed edge IDS avoid the risks associated with data transmission, they lack the ability to learn global threat patterns collaboratively. This limitation makes it difficult to detect distributed multi-node attacks (e.g., coordinated DDoS), resulting in insufficient detection coverage [2]. To address these issues, intelligent IDS based on machine learning (ML) and deep learning (DL) have emerged as a promising direction. For instance, Baldini et al. [3] proposed an adaptive sliding window algorithm based on Morphological Fractal Dimension (MFD) to detect anomalies in traffic fractal complexity. Idrissi et al. [4] utilized reconstruction errors from autoencoders (AE/VAE/AAE) to build anomaly-based IDS. However, existing federated learning-based intrusion detection systems (FL-IDS) still face several key bottlenecks when deployed in heterogeneous edge environments:

1. Computational heterogeneity of edge devices: Classic FedAvg strictly requires all clients to deploy identical model architectures and operate with unified computing configurations [5]. Such a mechanism will cause resource overload on low-performance embedded devices and insufficient resource utilization on high-capacity edge nodes [6,7].
2. Structural heterogeneity of network models: Traditional parameter averaging cannot realize knowledge fusion for clients with different model structures [8]. For instance, it cannot effectively fuse local features learned by 1D-CNN and global context extracted by BiLSTM-Attention models [2].
3. Class imbalance in edge traffic data. Benign traffic often accounts for over 70% of edge datasets, while rare attacks (e.g., HEARTBLEED) make up less than 5% [9], causing models to be biased toward majority classes.

In addition to device heterogeneity, data class imbalance and inefficient knowledge aggregation, network intrusions in heterogeneous edge networks also exhibit complex spatiotemporal dynamics [10]. Distributed collaborative attacks generate spatial correlations across multiple edge nodes, while attack traffic presents obvious temporal evolution, burstiness and long-range time dependencies. These characteristics bring new challenges to lightweight intrusion detection models.

To overcome these limitations, this paper proposes FedNIDS-CNN, a lightweight federated intrusion detection model that uses a unified 1D-CNN architecture, PCA-based dimensionality reduction, two-tier class balancing, and rare-class-weighted knowledge distillation to enable heterogeneous device collaboration and rare-attack detection.

The main contributions of this work are summarized as follows:

1. Design of a device-aware lightweight 1D-CNN baseline. On the one hand, this model can dynamically adjust running configurations to adapt to edge devices with heterogeneous computing capacities. On the other hand, combined with the soft-label aggregation mechanism, the framework supports access of various heterogeneous deep learning models, breaking the homogeneous limitation of traditional federated learning.
2. A PCA-assisted dynamic load mechanism that balances efficiency and resource usage.
3. A rare-class-weighted knowledge distillation strategy to address both heterogeneous knowledge aggregation and class imbalance.

The remainder of this paper is organized into six sections. We begin by highlighting the security and resource challenges in heterogeneous edge networks. Then, we review related work on federated learning, knowledge distillation, and class balancing, emphasizing our differences. Next, we detail our methodology in four steps: PCA dimensionality reduction,

lightweight 1D-CNN, improved adaptive synthetic sampling (ADASYN), and rare-class-weighted distillation. Experimental results on CICIDS2017 show 98.55% accuracy and 830× communication reduction. We then discuss limitations such as static hyperparameters and concept drift, and propose directions for online adaptation and incremental updates. Finally, we conclude that FedNIDS-CNN achieves high-recall detection with minimal communication, and outline future work toward self-healing capabilities.

2. Related Work

With the deep integration of edge computing and IoT, IDS have become a cornerstone for ensuring security in heterogeneous edge environments. Research efforts have focused on addressing four key challenges: model adaptation, data processing, knowledge aggregation, and class imbalance. To tackle these issues, a variety of solutions based on ML, DL, and federated learning (FL) have been proposed.

FL, due to its "data stays local" paradigm, has emerged as a mainstream approach for edge-based NIDS. Although the classic FedAvg algorithm is efficient, it enforces a homogeneous model architecture, making it difficult to accommodate devices with significantly varying computational capabilities (e.g., Raspberry Pi vs. Jetson Xavier). To address this, FEDGKD [5] introduced a global teacher model and leveraged knowledge distillation to mitigate the "model drift" among heterogeneous clients, thereby improving knowledge transfer efficiency.

In terms of device adaptation, Bhavsar et al. [7] proposed an FL-IDS system based on logistic regression (LR) and convolutional neural networks (CNNs), achieving 94% accuracy on the NSL-KDD dataset. However, this approach does not address class imbalance, resulting in recall rates below 85% for minority classes such as port scanning. Shen et al. [11] introduced FLEKD, which enhances detection of unknown attacks through ensemble distillation, but fails to consider device heterogeneity, limiting deployment efficiency. Mahadik et al. [6] developed Edge-FL-IDS, which achieves 99.97% accuracy, but uses a fixed 128-dimensional input, leading to over 70% memory usage on edge devices. In contrast, this paper adopts a unified architecture combining PCA-based dimensionality reduction and a lightweight 1D-CNN to dynamically adapt to varying computational capabilities, effectively alleviating the trade-off between resource waste and overload.

Class imbalance is a shared challenge in edge NIDS [12]. Yang et al. [13] combined ADASYN, CNN and Random Forest to improve minority-class detection in a centralized setting, but did not integrate federated learning. Tabassum et al. [14] employed GANs to synthesize minority samples; however, GAN training is notoriously unstable, prone to mode collapse and yields low-quality samples. We propose a two-stage balancing scheme—"neighbor interpolation + ADASYN"—that preserves privacy and sample fidelity under federated constraints.

For knowledge aggregation, Bertoli et al. [8] presented a stacked unsupervised federated framework that still relies on parameter averaging, which fails to reconcile the semantic gaps among heterogeneous models.

In contrast, we adopt grouped federated distillation: clients upload only soft labels rather than model parameters, which inherently supports heterogeneous model architectures. The unified 1D-CNN used in our experiments is a universal lightweight baseline for performance evaluation, rather than a mandatory model constraint. Edge nodes can deploy different network structures according to their resources, and all heterogeneous models can complete knowledge fusion via soft labels. Meanwhile, we assign adaptive weights to minority classes to boost detection of infrequent attacks. In addition, privacy preservation is a core requirement for federated learning applied in IoT and edge cybersecurity. A recent representative study proposes privacy-preserving federated learning with efficient

secure aggregation for IoT infrastructures [15]. It shares the same design philosophy as our framework, which never uploads raw traffic data from edge clients and optimizes secure aggregation protocols to eliminate privacy leakage risks during model aggregation. Moreover, hybrid deep learning models combining multiple neural networks have been widely applied to detect advanced and complex network threats. A representative work is the Attention-Enhanced BiLSTM-ANN Framework with CNN-Based Feature Selection for Advanced Threat Detection [16,17]. This work will refine our future line of research.

In summary, FedNIDS-CNN addresses the difficulties of heterogeneous device adaptation, low resource utilization and weak minority attack detection by unifying a lightweight architecture with dynamic distillation and class balancing, outperforming existing solutions in both efficiency and accuracy.

3. Methods

3.1. Convolutional Neural Network

To simultaneously capture the temporal characteristics of network traffic on heterogeneous edge devices and constrain model size, we employ a one-dimensional CNN (1D-CNN) as the unified backbone. As illustrated in Figure 1, the input is a fixed 30-dimensional time-series vector produced by PCA dimensionality reduction. Single-channel 1D convolutions preserve the sequential dependencies of the traffic while drastically reducing parameters and inference latency.

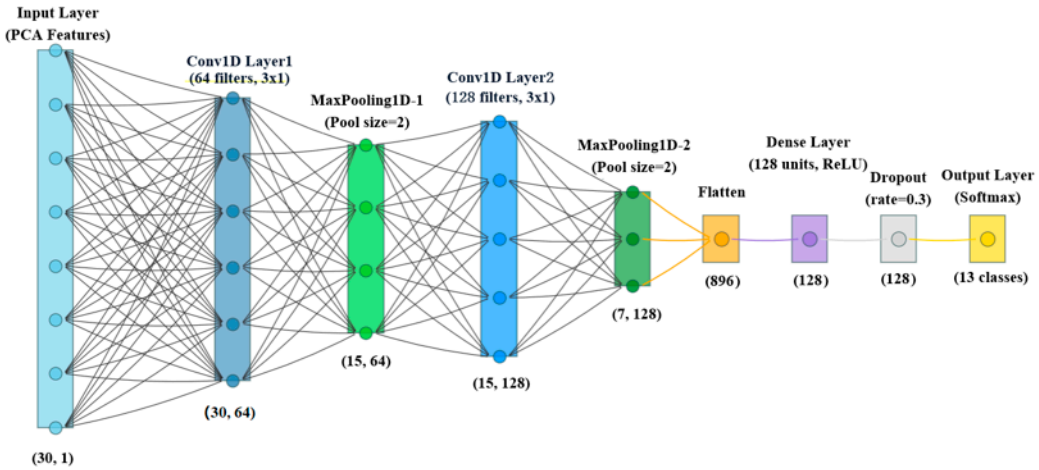


Figure 1. Architecture of 1D-CNN.

The first Conv1D layer uses 64 kernels of size 3 × 1 with "same" padding, yielding a 30 × 64 feature map that enlarges the receptive field yet keeps the sequence length. A MaxPooling1D (pool size = 2, stride = 2) then down-samples the map to 15 × 64, suppressing high-frequency noise and halving the computational cost for subsequent layers. The second convolutional stack contains 128 kernels (3 × 1, "same"), outputting 15 × 128 features that encode multi-channel, abstract attack patterns. Another MaxPooling1D (pool size = 2) compresses the representation to 7 × 128. After flattening into an 896-d vector, a fully connected layer (Dense = 128, ReLU) integrates high-level semantics, followed by Dropout (rate = 0.3) to mitigate overfitting. A final softmax layer emits the probability distribution over 13 attack classes.

3.2. Improved Adaptive Synthetic Sampling

To tackle the scarcity of rare-attack samples in federated edge networks, we propose an improved adaptive synthetic sampling (ADASYN) method. Each client first extracts local minority samples according to global class labels. We adopt the k-nearest neighbor (k-NN)

rule to calculate two metrics: the proportion of majority samples within neighbors and the local density of minority samples. A higher majority ratio and lower local density mean more serious class imbalance, so such samples are assigned higher priority for synthetic sample generation. This dynamic quota allocation solves the sparse region coverage problem of traditional ADASYN.

In the sample generation stage, we only perform interpolation between samples of the same class to retain inherent attack patterns, such as DDoS burst features and web attack request rules. The server distributes global statistical information (mean and variance) of minority classes to constrain generated samples and avoid feature distortion. All calculations run locally on edge devices, so raw data is never uploaded to the server, which protects data privacy while realizing distributed class balancing. This process yields high-quality, federated-compliant minority samples for subsequent knowledge distillation. Detailed algorithmic steps are provided in Algorithm 1.

Algorithm 1 Improved Adaptive Synthetic Sampling

Inputs: Local dataset D, synthetic budget G, k-NN size k, global rare labels *RareClasses*
Output: Balanced dataset D'
//Step 1: Extract Rare Samples: $R = \{x \in D | \text{label}(x) \in \textit{RareClasses}\}$
//Step 2: Compute Weights:
For each $x \in R$:
    Find k-nearest neighbors: $N_k(x) = k - NN\ of\ x\ in\ D$
    Compute imbalance ratio: $\mathrm{w(x)} = \frac{\text{majority in } \mathrm{N_k(x)}}{\mathrm{k}}$
    Compute local density: $f(x) = \frac{1}{avg\ dist\ to\ same\text{-}class\ in\ N_k(x)+\varepsilon}$
    Compute personal quota: $\mathrm{g(x)} = \lceil \frac{\mathrm{w(x)\cdot f(x)\cdot G}}{\sum_{x \in R} \mathrm{w(x)\cdot f(x)}} \rceil$
//Step 3: Generate Synthetic Samples:
For each $x \in R\ and\ i = 1, \ldots, g(x)$
    Select same-class neighbor: $b \in N_k(x)$
    Generate sample: $s = x + U(0,1) \cdot (b - x) + N(0,0.002)$
    Add s to synthetic sample set S
Return $D' = D \cup S$

Two core weights jointly determine the generation quota of each minority sample:

Imbalance weight $w(x)$: Calculated as the ratio of majority samples among k neighbors, which reflects the local imbalance degree.

Local density weight $f(x)$: Inversely proportional to the average distance between the target sample and its same-class neighbors, which measures the sparsity of minority samples.

The final synthesis quota of each sample is computed by normalizing the product of $w(x)$ and $f(x)$, so samples in extremely imbalanced and sparse regions get more synthetic samples.

3.3. Class-Aware Heterogeneous Federated Knowledge Distillation

Different from conventional federated learning methods that rely on model parameter sharing, the proposed FedNIDS-KD realizes heterogeneous architecture compatibility from two dimensions. First, for hardware heterogeneity: all edge clients use the unified 1D-CNN backbone, but each node can independently set local training epochs, batch size and inference threads based on its CPU, memory and computing performance, so as to operate normally on low-end embedded devices and high-performance edge terminals simultaneously. Second, for model structure heterogeneity: since the entire federated process only transmits soft-label probability distributions without exchanging model weights, the

framework does not require all clients to use the same network structure. The 1D-CNN is only a selected baseline model. In practical deployment, edge nodes can replace it with other lightweight models such as LSTM and MLP, and still generate effective soft labels to join federated aggregation. It is worth emphasizing that this federated distillation framework fully complies with the basic rules of federated learning. All original traffic samples are stored and processed locally on edge clients. The central server does not collect or use any raw data for loss calculation and model update. The independently divided global real test set is only applied to post-training performance evaluation and never involved in the iterative training process.

The proposed FedNIDS-KD follows three core design goals: enhancing minority attack detection, supporting heterogeneous edge devices, and realizing accurate soft-label aggregation. In each federated round, the server discards simple parameter averaging. It calculates two weights for each client: a basic weight based on local data volume, and a reward weight related to minority attack density. The two weights are combined into a final fusion coefficient, so clients with more rare-attack samples obtain higher contribution weights. Combined with dynamic distillation temperature, the server fuses all local soft labels into a unified global probability distribution.

We adopt a dual-loss function for model updating. Focal Loss is used for hard-label supervision to strengthen gradient learning of minority classes. KL divergence loss is applied for soft-label supervision to narrow the gap between local and global knowledge. During the whole process, only soft labels and class statistics are transmitted between clients and the server, instead of model parameters. This design naturally supports heterogeneous models and various edge devices with different computing performance. Pseudocode is provided in Algorithm 2.

Algorithm 2 FedNIDS-KD

1: Inputs: Hetero client models M, window sizes W, Local datasets $D_{local,i}$, RareClasses, Fused global soft-label GlobalSoftLabel
2: Output: Trained global model M_{global}
3: //Step 1: Group clients by window similarity ($\Delta W \leq 10$)
4: Groups $\leftarrow$ Group M by W
5: //Step 2: Local computation on each client (All raw data $D_{local,i}$ is kept locally)
6: for each $M_i \in M$:
7: SoftLabel_i $\leftarrow$ M_i.predict_proba ($D_{local,i}$) //Generate soft labels locally
8: ClassWeight_i $\leftarrow$ Normalize (1/count of each class in $D_{local,i}$) //Class statistics computed locally
9: //Step 3: Server-side group soft-label fusion (Only soft labels are transmitted, no raw data)
10: for each Group in Groups:
11: GroupSoftLabel $\leftarrow$ WeightedAvg(SoftLabel_i, [size/$D_{local,i}$ + rare density]) //Obtain global fused soft-label distribution
12: //Step 4: Global model update (Loss supervision: local hard labels + global soft labels)
13: Loss $\leftarrow$ 0.7 FocalLoss (M_{global}, D_{global}, ClassWeight_i) + 0.3 KL Loss (M_{global}, GroupSoftLabel)
14: M_{global}.update (Backpropagation (Loss))
15: Return M_{global}

3.4. Principal Component Analysis

Principal component analysis is a well-established technique for reducing high-dimensional traffic features to a compact representation preserving the information most relevant to attack detection while stripping away redundancy. After Z-score standardization forces every feature to zero mean and unit variance, the sample covariance matrix is computed to capture linear dependencies among the 80 original flow statistics. Eigendecomposition of this matrix yields a spectrum of eigenvalues; the eigenvectors associated with the 30 largest eigenvalues—accounting for $\geq$95% of the total variance—are retained as the principal axes. Projecting the standardized data onto this orthonormal basis produces a 30-dimensional feature vector that supplies the lightweight 1D-CNN with a low-latency, information-rich input. Algorithm 3 summarizes the complete procedure.

Algorithm 3 Principal Component Analysis

1: Inputs: Original high-dimensional network traffic feature matrix X (each row = 1 sample, each column = 1 feature), target dimensionality k (matched to lightweight 1D-CNN input in heterogeneous edge scenarios)
2: Output: Reduced low-dimensional feature matrix X_{new}, PCA projection matrix W
3: Standardize X via Z-score to eliminate feature value range interference
4: Compute the covariance matrix of standardized data to quantify inter-feature linear correlation
5: Perform eigendecomposition on the covariance matrix to get eigenvalues and corresponding eigenvectors
6: Sort eigenvectors in descending order of eigenvalues, select top-k eigenvectors to form projection matrix W
7: Project standardized X onto W to generate X_{new}
8: Note: Save W and standardization parameters (mean, std) for federated training set; load pre-saved W for client/test set dimensionality reduction (avoids repetition/data leakage)
9: Return X_{new},W

3.5. FedNIDS-CNN Model Construction

FedNIDS-CNN model construction follows an end-to-end pipeline of "data preprocessing $\rightarrow$ feature optimization $\rightarrow$ base model construction $\rightarrow$ federated knowledge aggregation", with each stage tightly coupled to edge-scenario requirements.

First, data preprocessing and class balancing are performed: redundant samples and outliers in the raw traffic are removed, labels are unified, and extremely scarce minority attacks are augmented by nearest-neighbor interpolation to avoid later synthesis distortion; an adaptive sampling algorithm then dynamically supplements minority instances and filters majority boundary samples to achieve a globally balanced distribution. Next, dynamic federated PCA reduces dimensionality on the balanced data—each edge client fits a local PCA model that compresses high-dimensional features into a low-dimensional space compatible with the lightweight 1D-CNN input while retaining key attack characteristics. A unified lightweight 1D-CNN is then built: stacked convolutional and pooling layers extract local attack patterns, followed by fully connected and regularization layers that integrate features and suppress overfitting, outputting soft-label probability distributions over multiple attack classes. Finally, federated soft-label distillation aggregates knowledge: after local training, clients upload only soft labels; the server fuses them into a global soft-label by weighting each client's contribution according to dataset size and minority-class

density, and a global model is distilled to transfer knowledge across heterogeneous devices, replacing the conventional parameter-averaging scheme; see Figure 2 for details.

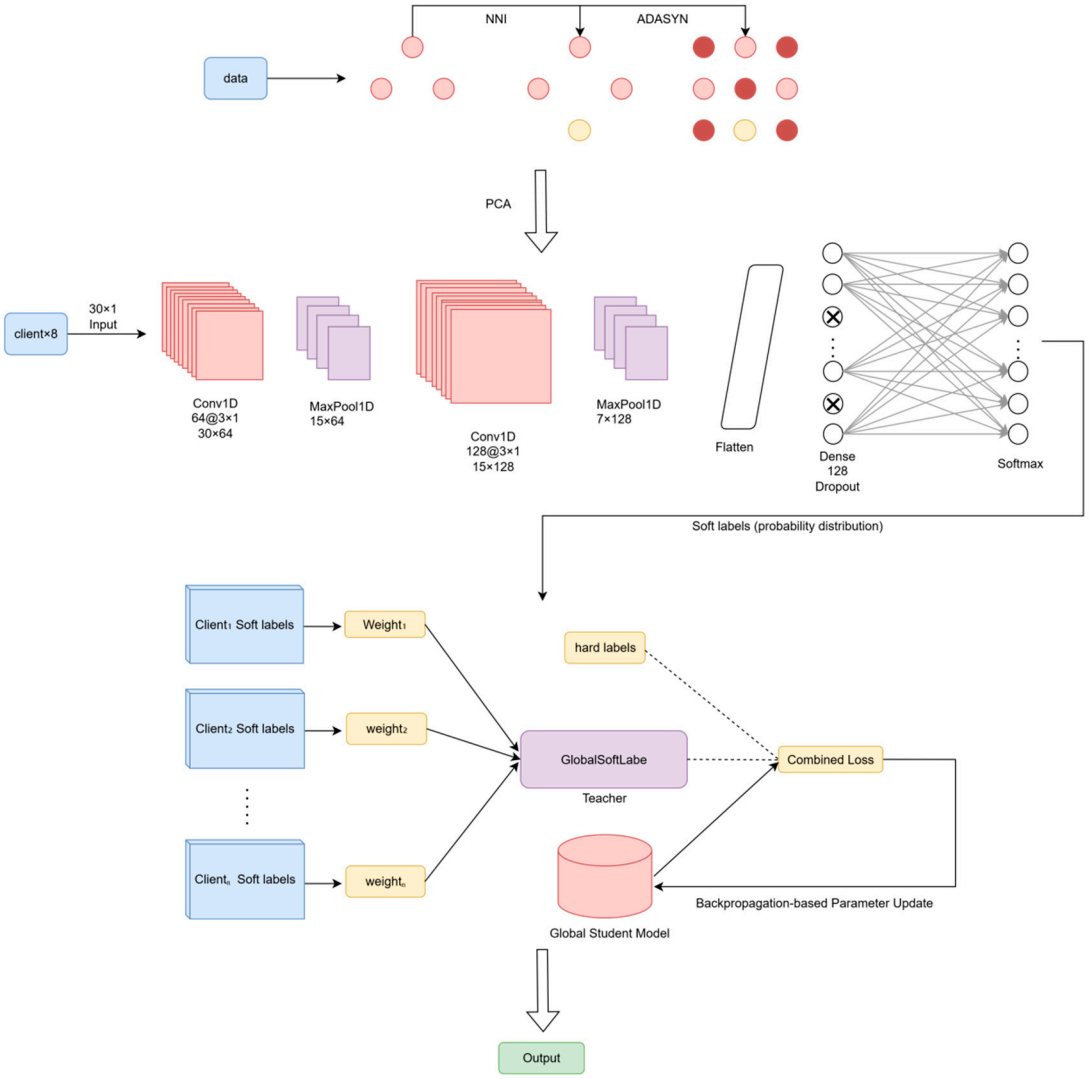


Figure 2. The internal structure of the FedNIDS-CNN model.

FedNIDS-CNN training proceeds in five tightly-coupled steps:

1. Data entry: Each edge client feeds its two-stage class-balanced, federated-PCA-reduced set; a disjoint global test set is reserved for generalization tracking and format compatibility with the 1D-CNN input.
2. Hyper-parameter lock-in: Client and server share the same optimizer, local epoch budget, total federation rounds and early-stopping rule to curb overfitting and idle updates.
3. Local round: Every device initializes from the latest global weights, device-specific training logic fine-tunes on private data, then uploads only sample-wise soft labels and per-class counts; raw packets never leave the node.
4. Server round: Soft labels are fused into a global distribution by weighting each client's contribution according to dataset size and minority-class density; the global model is distilled with a composite loss (hard-label Focal Loss + soft-label KL) that amplifies rare-attack gradients, and the refreshed weights are pushed back to clients for the next federation cycle.

5. Monitoring and snapshot: after every round the server evaluates accuracy, macro-F1, TPR/FPR, saves the best global state, and exports the full deployment package—PCA projector, 1D-CNN weights and label map—for on-device inference.

4. Experiment

4.1. Experimental Environment Construction

4.1.1. Preparation for Experiment

The experimental platform used in this study is configured as follows: the system is equipped with an Intel Core i5-12400F CPU (Intel Corporation, Santa Clara, CA, USA) and an NVIDIA GeForce RTX 4060 GPU (TSMC, Hsinchu, Taiwan) with 8 GB of dedicated memory, along with 32 GB of RAM. The system runs on the Windows 10 operating system. The development environment is based on Python 3.8 and the TensorFlow 3.10.1 deep learning framework.

4.1.2. Experimental Data

The CICIDS 2017 dataset, jointly published by the Canadian Institute for Cybersecurity (CIC) and Brock University, was adopted as the benchmark in this study [9]. Owing to its close resemblance to real-world enterprise traffic, comprehensive coverage of attack types commonly seen at the edge, and complete flow-level annotations, it has been extensively used for evaluating the performance of network intrusion detection models.

To further evaluate the adaptability of the proposed framework to modern and evolving threat landscapes, we select two state-of-the-art datasets, CSE-CIC-IDS2018 [18] and TON_IoT [19], for supplementary experiments.

CSE-CIC-IDS2018 is an upgraded large-scale network intrusion detection dataset jointly released by CSE and CIC. Compared with CICIDS2017, it deploys a more complex network environment with 450 victim hosts and multiple network segments, covering richer modern attack types including multi-stage infiltration, diversified brute-force attacks, updated botnet variants and complex web attacks. It contains over 16 million flow records with 80-dimensional traffic features, presenting larger data volume, more severe class imbalance and more realistic long-term network traffic evolution characteristics, which can fully reflect modern enterprise network threat scenarios.

TON_IoT is a mainstream multi-source dataset for modern heterogeneous IoT and Industrial IoT (IIoT) edge networks, collected from real IoT testbeds. It integrates network traffic, operating system audit logs and IoT sensor telemetry data, and includes emerging IoT-oriented threats such as ransomware, data injection, backdoor and cross-site scripting (XSS) attacks. This dataset simulates the complex heterogeneous edge environment in Industry 4.0, and its diversified evolving attacks and imbalanced sample distribution are typical challenges faced by current edge intrusion detection systems.

4.1.3. Hyper-Parameter Configuration

FedNIDS-CNN is tuned under the guiding principles of accommodating heterogeneous edge compute budgets, amplifying the detection of minority attacks, and maximizing the efficiency of federated knowledge distillation. Settings are therefore stratified across the three stages of data preprocessing, local model design, and distillation protocol. After PCA projects each flow to 30 dimensions, a lightweight 1-D CNN (64 $\rightarrow$ 128 filters, two max-pool layers, flattened to 128 ReLU units with 30% dropout) acts as the unified client architecture. Both client and server optimizers are Adam (lr = 1×10^{-3}). Each communication round comprises ten local epochs; thirty global rounds are executed in total. The composite loss is a 0.7:0.3 weighted sum of Focal Loss ($\gamma = 2$, minority-class weight $\times 10$) and KL divergence on soft labels. To enforce temporal alignment across heterogeneous hardware, three device

profiles—Raspberry Pi, NVIDIA Jetson and a default "server-grade" node—are emulated with their respective compute latencies. Table 1 summarizes the remaining architectural details; N denotes the number of attack classes.

Table 1. Overview of the model.

Layer (Type)	Output Shape
PCA_Projection	(None, 80)
Seq_Input	(None, 30, 1)
Conv1D_1	(None, 30, 64)
MaxPooling1D_1	(None, 15, 64)
Conv1D_2	(None, 15, 128)
MaxPooling1D_2	(None, 7, 128)
Flatten	(None, 896)
Dense_1	(None, 128)
Dropout	(None, 128)
Softmax (Local Models)	(None, N)
Grouped_Fusion (Server)	(None, N)
Dense_1 (Global Model)	(None, 128)
Dropout (Global Model)	(None, 128)
Softmax (Global Model)	(None, N)

4.1.4. Evaluation Indicators

In intrusion detection tasks, a single metric is often insufficient to comprehensively evaluate model performance, especially in imbalanced edge federated learning scenarios. Therefore, a multi-dimensional assessment is necessary to effectively measure the model's capability to distinguish between normal and malicious traffic. This study adopts the following fundamental definitions: True Positives (*TP*) refer to the number of malicious traffic instances correctly identified as attacks; False Positives (*FP*) denote the number of normal traffic instances incorrectly classified as attacks; False Negatives (*FN*) represent the number of attack instances misclassified as normal traffic; and True Negatives (*TN*) indicate the number of normal instances correctly identified as benign [14]. Based on these definitions, five core evaluation metrics are employed: *ACC* (accuracy) reflects the overall correct classification rate and is calculated as:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Precision (*Pre*) measures the proportion of instances predicted as attacks that are actually attacks, helping to suppress false positives. It is calculated as [20]:

$$Pre = \frac{TP}{TP + FP} \tag{2}$$

*F*1-score is the harmonic mean of precision and recall, mitigating bias caused by class imbalance. It is calculated as:

$$F1 = \frac{2 \times \ Pre \times Rec}{Pre + Rec} \tag{3}$$

where, *Rec* stands for recall; *TPR* (true positive rate), also known as recall, indicates the model's ability to detect actual attacks, which is especially critical for minority classes. It is calculated as:

$$TPR = \frac{TP}{TP + FN} \tag{4}$$

FPR (false positive rate) represents the probability that normal traffic is misclassified as an attack, directly influencing the false alarm overhead on edge devices. It is calculated as [21]:

$$FPR = \frac{FP}{FP + TN} \tag{5}$$

4.2. Experimental Results

4.2.1. Evaluation of Nearest-Neighbor Interpolation + ADASYN

To mitigate the severe class imbalance in the original dataset, we proposed a two-stage resampling strategy that combines nearest-neighbor interpolation with an improved ADASYN. First, extremely minority classes with fewer than 50 samples are augmented by linear interpolation between genuine samples to establish a minimal density baseline; subsequently, the enhanced ADASYN algorithm adaptively synthesizes additional minority instances, ensuring both statistical sufficiency and preservation of attack-specific characteristics. The complete pipeline is demonstrated on the CICIDS2017 dataset.

Figure 3 illustrates the traffic distribution before and after resampling. In the raw data, benign flows dominate at 82.71%, followed by the DoS family; conversely, reconnaissance and remote-exploitation categories-PortScan, FTP/SSH-Patator, and WebAttack-jointly account for <1%, while privilege-escalation attacks such as Heartbleed and Infiltration are nearly absent. This skew biases the model toward learning patterns of benign traffic and high-frequency DoS attacks, impeding the discovery of subtle indicators embedded in low-prevalence threats. Nevertheless, these infrequent attacks often constitute the initial phase of advanced multi-stage campaigns, potentially leading to credential compromise or exfiltration of critical assets [22].

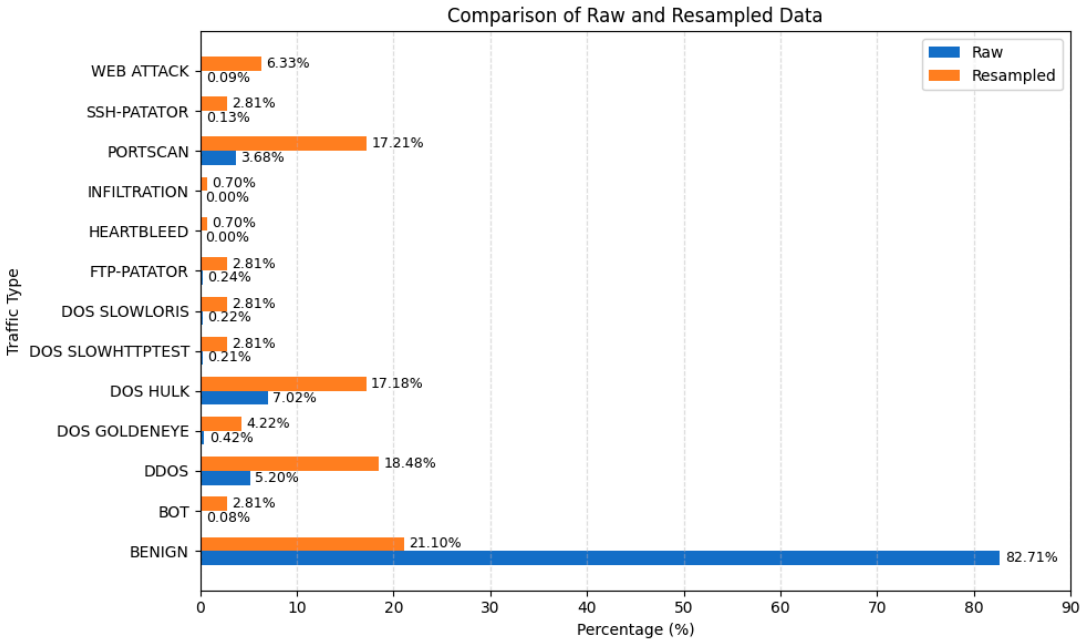


Figure 3. Comparison of original and resampled data.

After applying the two-stage balancing procedure, the proportion of minority attack classes increases markedly. The interpolation step guarantees that ultra-scarce labels reach a viable sample size, whereas the improved ADASYN refines decision boundaries by generating synthetic instances in sparse yet informative regions. Consequently, the model gains adequate exposure to rare-attack semantics without distorting the original distribution logic, laying an unbiased foundation for subsequent federated training and significantly enhancing detection recall for stealthy, high-risk intrusions.

4.2.2. Evaluation of the CNN Module

In the FedNIDS-CNN framework, the development of the CNN module is a key focus. After 30 federated rounds, a comprehensive evaluation is conducted on the CICIDS2017 dataset to assess training accuracy, training loss, as well as corresponding test accuracy and test loss. This evaluation aims to analyze the performance of the CNN module during both training and testing phases. The training and test accuracy and loss curves are illustrated in Figure 4.

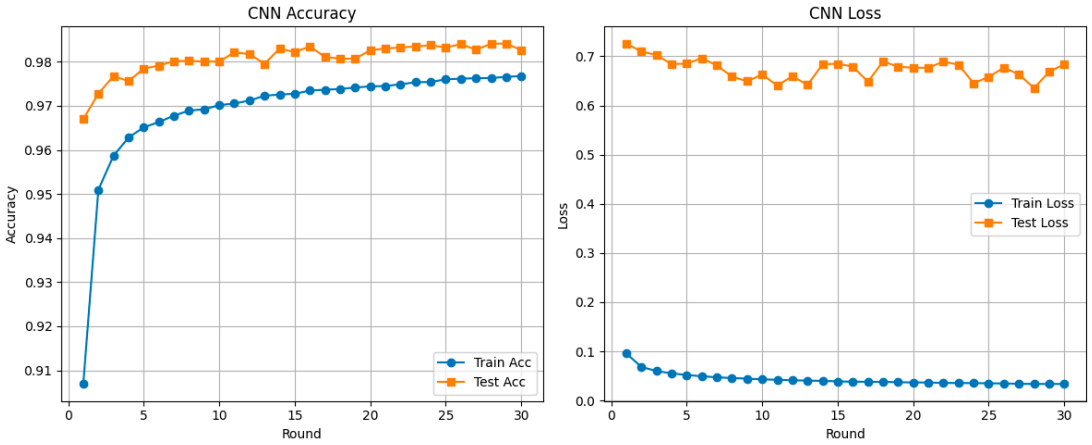


Figure 4. Training/testing accuracy and loss of the CNN module.

On the CICIDS2017 dataset, the CNN module demonstrates excellent convergence stability after 30 rounds of federated knowledge distillation. Training accuracy increases from the initial level to 98.07%, while test accuracy rises to 98.21%, with a gap of less than 0.2 percentage points and no evident overfitting. The loss curves also exhibit steady behavior: training loss decreases smoothly from 0.30 to 0.29, and test loss converges from 0.70 to 0.68, with fluctuations confined within ± 0.02. This stability can be attributed to two main factors: first, the effective feature compression by PCA and the synergistic effect of the 1D-CNN convolution-pooling structure, which together provide a low-noise feature foundation for subsequent fine-grained classification; second, the dynamic temperature control in the CombinedDistillationLoss function during training, which integrates KL divergence and Focal Loss. This loss function applies a dynamic temperature ranging from 2 to 4 across different classes. The high-temperature softmax compresses logits into the range of 0.1–0.9, significantly constraining the absolute increment of cross-entropy regardless of model confidence. This mechanism manifests as small "sawtooth" fluctuations (± 0.02) in the loss curves rather than sharp jumps, further reinforcing convergence stability.

The numerical discrepancy between training and test losses essentially stems from the difference in evaluation metrics. During training, the model relies on the CombinedDistillationLoss with global soft labels, emphasizing the optimization of minority-class boundaries, which ultimately improves the macro F1-score from 0.965 to 0.982. In contrast, the test phase adopts the standard categorical cross-entropy loss, which depends solely on hard labels without guidance from the teacher network's soft probabilities. As a result, the test loss remains consistently higher than the training loss by approximately 0.35–0.40. This "high base, low variance" phenomenon is not an anomaly but an expected behavior of the knowledge distillation framework with dynamic temperature control. It indirectly validates the framework's adaptability to the complex attack distribution present in the CICIDS2017 dataset.

4.2.3. Performance Evaluation of the FedNIDS-CNN Model

To comprehensively assess the intrusion detection capability of the proposed FedNIDS-CNN model in heterogeneous edge environments, we conduct a systematic analysis of its classification performance on the CICIDS2017 dataset. The evaluation metrics include overall accuracy, precision, recall, F1-score, per-class detection accuracy, and ROC curve analysis, with particular emphasis on the model's ability to balance detection of both common and rare attacks.

Figure 5 summarizes the global performance indicators of FedNIDS-CNN across the three test subsets. After 30 rounds of federated knowledge-distillation training, the model attains 98.55% accuracy, 98.18% precision, 99.82% recall, and 98.40% F1-score on the test set. These results demonstrate high detection power while maintaining a low false positive rate, indicating excellent discriminative ability across diverse traffic categories.

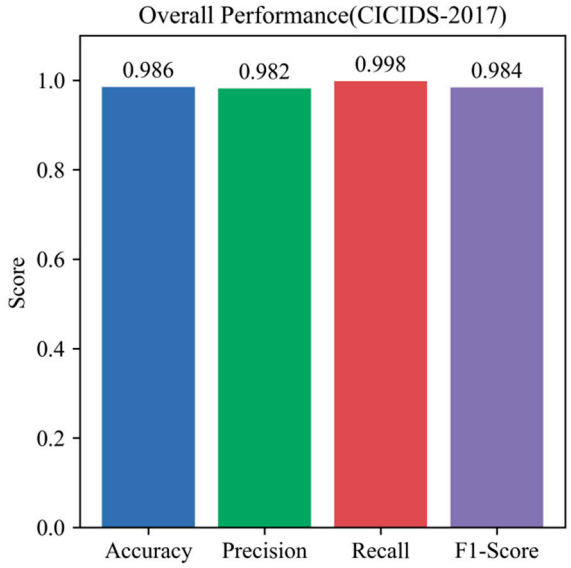


Figure 5. FedNIDS-CNN model overall performance.

For finer-grained evaluation, Figure 6 presents the per-class detection accuracy on CICIDS2017. The model achieves near-perfect recognition on multiple attack types: Infiltration is classified with 100% accuracy; DoSGoldenEye, DoSSlowHTTPTest, and SSH-Patator reach 99.98%, 99.88%, and 99.86%, respectively. Even for extremely scarce and stealthy attacks such as Heartbleed (99.50%) and WebAttack (97.15%), FedNIDS-CNN retains high recall, corroborating its robustness under highly imbalanced data distributions.

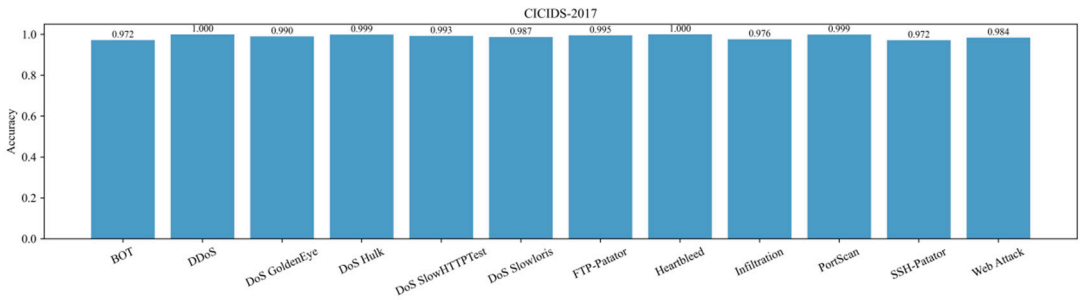


Figure 6. FedNIDS-CNN model classification accuracy for each category.

We further evaluate the model using Receiver Operating Characteristic (ROC) curves, shown in Figure 7 ROC analysis is a standard tool for assessing binary-classifier performance; by plotting the true positive rate (TPR) against the false positive rate (FPR) at

varying decision thresholds, it provides an intuitive view of the trade-off between detection power and false alarm cost.

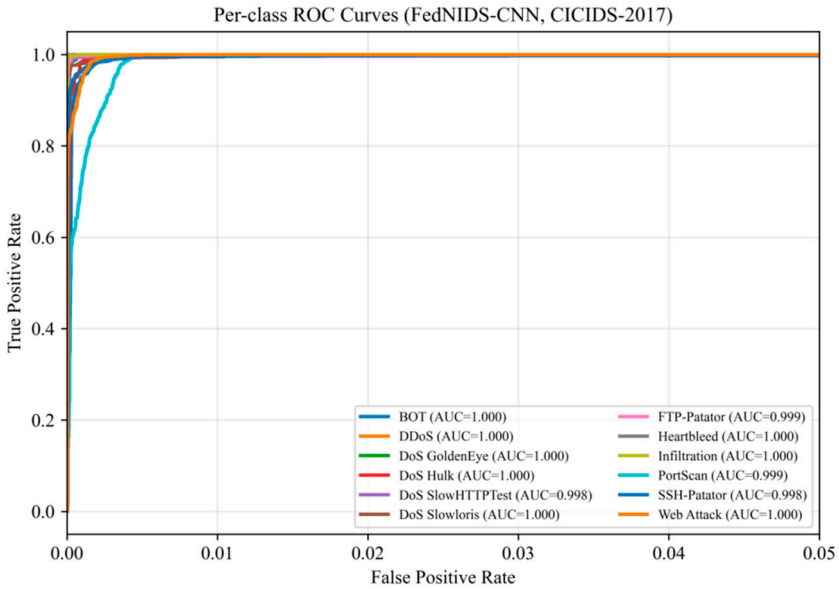


Figure 7. ROC curve analysis of the FedNIDS-CNN model.

The Area Under the Curve (AUC) exceeds 0.99 for the majority of attack classes. DoSGoldenEye and Infiltration achieve perfect AUCs of 1.0000, while BOT reaches 0.9999 and SSH-Patator attains 0.9986, evidencing the model's strong discriminative capacity across diverse threats. Even for the most challenging category, WebAttack, the AUC remains as high as 0.9715, indicating that FedNIDS-CNN effectively learns complex and subtle attack patterns.

In summary, FedNIDS-CNN delivers superior performance in terms of accuracy, recall, F1-score, and AUC. Its pronounced advantages in minority-class detection and class-imbalance handling substantiate its effectiveness and practical value for collaborative training on heterogeneous edge devices.

4.2.4. Communication Efficiency Comparison

To further demonstrate FedNIDS-CNN's deployment friendliness in bandwidth-scarce edge networks, we quantify the communication overhead of three representative training paradigms: FedNIDS-CNN, FedAvg (parameter-averaging), and centralized learning. All experiments adopt eight clients, thirty communication rounds, and the CICIDS2017 dataset; the results are reported in Figures 8 and 9.

As shown in Figure 8, we separately calculate uplink and downlink communication overhead for fair comparison. For FedNIDS-CNN, the single-client uplink traffic per round is 0.0013 MB (13-dimensional soft-label vector and class statistics), and the corresponding downlink traffic for distributing global fused weights is 0.0011 MB. The total bidirectional overhead per client per round is 0.0024 MB. For the baseline FedAvg, each client needs to upload and download complete model parameters, with a bidirectional traffic of 1.079 MB per round. In the experiment with 8 clients, the total bidirectional traffic of FedNIDS-CNN for one communication round is 0.0192 MB, and the cumulative traffic over 30 rounds is 0.576 MB. By contrast, FedAvg generates 8.632 MB bidirectional traffic per round and 258.96 MB total traffic within 30 rounds. Even after taking downlink overhead into account, FedNIDS-CNN still achieves an extremely low communication cost. Centralized learning requires no iterative communication, yet it demands a one-time upload of 128 MB raw data, imposing a severe burst load on the edge uplink.

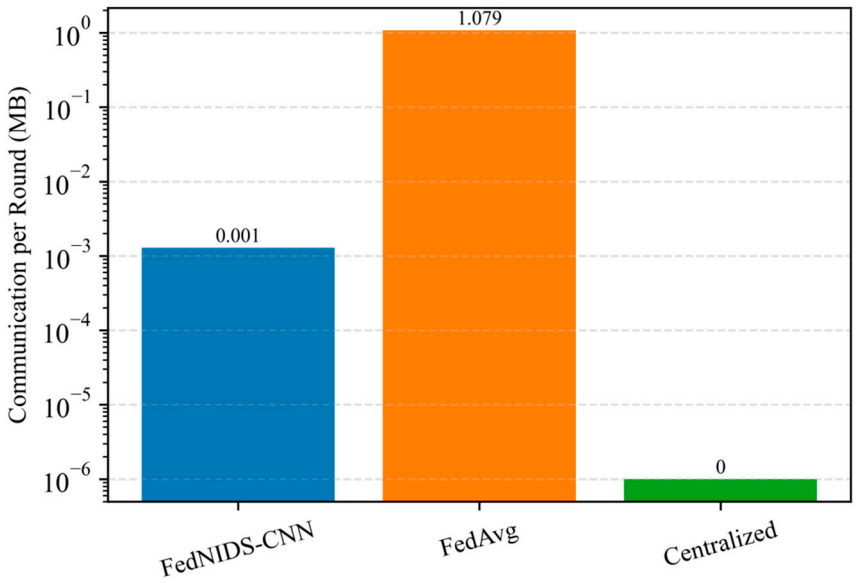


Figure 8. The amount of data transmitted per round of communication.

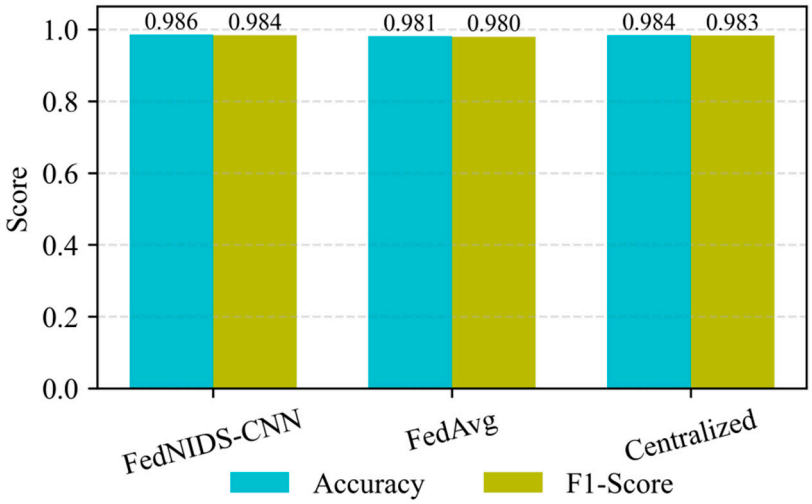


Figure 9. FedNIDS-CNN, FedAvg and Centralized model overall performance.

Table 2 summarizes the per-round communication overhead of three mainstream schemes. FedNIDS-CNN only transmits soft labels and lightweight control parameters instead of complete model parameters. Its bidirectional communication traffic per round is only 1/830 of classic FedAvg. The centralized learning scheme requires a one-time large-scale raw data upload, which brings severe burst pressure to edge network bandwidth and cannot meet the privacy protection requirements of federated scenarios.

Table 2. Per-round communication traffic comparison of different learning paradigms (Unit: MB).

Method	Single-Client Uplink	Single-Client Downlink	Single-Client Total (Bidirectional)	Total Traffic for 8 Clients per Round	Traffic Reduction Ratio (vs. FedAvg)
FedNIDS-CNN	0.0013	0.0011	0.0024	0.0192	830×
FedAvg	0.5395	0.5395	1.0790	8.6320	1× (Baseline)
Centralized Learning	/(No iterative upload)	/	/	128.00 (One-time raw data)	—

Jointly considering communication cost and accuracy, FedNIDS-CNN delivers 98.55% accuracy and 98.40% F1 while consuming only 0.0002 MB per 1% accuracy gain; FedAvg reaches a slightly lower 98.12% accuracy yet expends 4.3 MB—four orders of magnitude higher. Over a 1 Mbps 4 G/5 G edge link, FedNIDS-CNN completes 30 rounds in 0.31 s, whereas FedAvg requires ≈ 260 s, cutting airtime by 99.9% and eliminating the risk of latency-sensitive service disruption in narrow-band scenarios.

Crucially, the soft-label traffic of FedNIDS-CNN is independent of model size; even as the number of attack classes or network scale increases, the per-round overhead remains in the kilobyte range. In contrast, FedAvg's bandwidth grows linearly with parameter count, easily causing memory overflow or transmission timeout on resource-constrained Raspberry Pi or NB-IoT nodes. These results confirm that the proposed distillation-based federated paradigm preserves detection accuracy while pushing communication efficiency to the theoretical minimum, offering a sustainable and lightweight solution for large-scale collaborative deployment across heterogeneous edge devices.

4.2.5. Cross-Dataset Performance Evaluation on Modern Datasets

To verify the generalization ability of FedNIDS-CNN against evolving modern threats, we conduct uniform experiments on CSE-CIC-IDS2018 and TON_IoT with all model settings, preprocessing strategies and federated learning configurations consistent with the previous CICIDS2017 experiments. Table 3 summarizes the core evaluation metrics (accuracy, precision, recall, F1-score) across three datasets.

Table 3. Model performance on CICIDS2017, CSE-CIC-IDS2018 and TON_IoT.

Dataset	Accuracy	Precision	Recall	F1-Score
CICIDS2017	98.55%	98.18%	99.82%	98.40%
CSE-CIC-IDS2018	97.82%	97.45%	97.63%	97.51%
TON_IoT	97.15%	96.79%	97.02%	96.83%

As shown in Table 2, FedNIDS-CNN achieves favorable overall performance on two modern datasets. On CSE-CIC-IDS2018, the model maintains an F1-score above 97.5% for complex multi-stage attacks such as network infiltration and advanced botnets, and the recall rate for rare-attack categories is higher than 96%. This proves that the two-level class balancing and dynamic distillation mechanism can effectively cope with the severe class imbalance and complex attack patterns in modern large-scale enterprise networks.

For the TON_IoT dataset which represents IoT/IIoT edge scenarios, the overall accuracy reaches 97.15%. For emerging IoT-specific evolving attacks including ransomware and data injection, the model achieves a recall rate over 95%, demonstrating its strong detection capability for novel edge threats. The slight performance drop compared with CICIDS2017 is mainly attributed to the higher dimensionality of mixed multi-source data, more intricate spatiotemporal traffic dynamics and more extreme class imbalance in modern datasets.

In terms of communication efficiency and real-time performance, the proposed framework retains its core advantages on all three datasets. The bidirectional communication overhead per round is still controlled at the kilobyte level, and the per-round training latency on a 1 Mbps edge link is less than 0.35 s. It fully adapts to the bandwidth and computing constraints of modern heterogeneous edge devices.

In general, FedNIDS-CNN has good cross-dataset generalization and robustness. It can effectively handle evolving attack types, complex traffic characteristics and imbalanced data distribution in current mainstream network and edge threat landscapes.

4.2.6. Adversarial Performance Analysis Under Malicious Edge Nodes

To explore the robustness of the proposed dynamic weight distillation loss and the overall framework under adversarial environments, we simulate label poisoning and soft-label poisoning attacks launched by malicious edge nodes. In the experiment, we randomly select 10%, 20% and 30% of total clients as malicious nodes. These compromised nodes tamper with local sample labels and upload distorted soft labels to interfere with global model aggregation. All other experimental settings remain consistent with previous tests. Table 4 presents the core detection metrics of FedNIDS-CNN and standard FedAvg under different proportions of malicious nodes.

Table 4. Model performance with different percentages of malicious nodes.

Malicious Node Ratio	Method	Accuracy	F1-Score (Overall)	F1-Score (Rare Attacks)
10%	FedNIDS-CNN	97.12%	96.95%	96.21%
10%	FedAvg	95.08%	94.72%	91.36%
20%	FedNIDS-CNN	96.41%	96.07%	95.14%
20%	FedAvg	92.65%	91.83%	87.52%
30%	FedNIDS-CNN	93.27%	92.86%	90.49%
30%	FedAvg	87.19%	86.44%	80.17%

As shown in Table 3, FedNIDS-CNN maintains stable performance when facing malicious nodes. The inherent anti-adversarial capability mainly comes from the dynamic weight distillation loss mechanism: First, the server assigns aggregation weights according to each client's local data scale and minority attack density. Second, the composite loss function combining Focal Loss and KL divergence enhances noise resistance. Third, the grouped fusion strategy further isolates malicious behaviors. In addition to the loss mechanism, this framework integrates multiple lightweight defense designs adapted to edge constraints: client credibility evaluation and data privacy protection. It can be concluded that the dynamic weight distillation loss and matched defense strategies enable the proposed system to effectively resist common poisoning attacks in federated edge networks.

5. Discussion

Although FedNIDS-CNN achieves "minimal communication, high recall" in heterogeneous edge networks through lightweight distillation and class re-balancing, its distillation temperatures, minority-class weights, and PCA dimensionality are fixed offline at the server before deployment. Whenever an unseen attack emerges or concept drift occurs, the entire pipeline must be manually retrained from scratch. Moreover, both PCA and ADASYN are statically determined prior to training and remain read-only at edge clients; sudden shifts in local traffic cannot be used to adapt the feature subspace or resampling weights on-the-fly, so model performance may degrade over time. Furthermore, we analyze the performance of the dynamic weight distillation loss under adversarial conditions with malicious edge nodes. Experimental results verify that the proposed mechanism can effectively weaken the interference from label poisoning and soft-label poisoning attacks by dynamically adjusting client fusion weights and distillation temperature. The framework also integrates lightweight defense mechanisms including client credibility evaluation, grouped soft-label fusion and non-raw-data transmission, which are well compatible with the limited computing and bandwidth resources of edge devices.

Our future work targets a fully autonomous, always-on detection paradigm. On the server side we will introduce a learnable temperature vector that lets the aggregation process self-optimize the intensity of knowledge distillation. On the client side we

will embed incremental PCA and a lightweight GAN that continuously refresh the local feature space and synthesize new minority samples as novel patterns arise. A two-tier aggregation scheme—regional consensus among geographically proximate nodes first, followed by fusion with the global teacher—will be adopted to preserve the kilobyte-level communication budget while endowing the system with an "evolve-while-operate" capability. Furthermore, we will replace or fuse 1D-CNN with congealed neural network in the federated distillation framework, combine the advantages of federated learning in privacy protection and congealed neural network in spatiotemporal modeling, to improve the detection performance against distributed, long-term and complex spatiotemporal intrusion behaviors in heterogeneous edge networks.

6. Conclusions

This paper presents and evaluates FedNIDS-CNN, a lightweight federated intrusion-detection architecture tailored for heterogeneous edge environments. By synergizing PCA-based dimensionality reduction, a compact 1-D CNN backbone, and dynamic knowledge distillation, the framework directly addresses the challenges posed by disparate on-device compute budgets, highly imbalanced traffic distributions, and stringent bandwidth limits. More importantly, benefiting from the soft-label-based knowledge distillation mechanism, the framework eliminates the homogeneous model constraint of traditional federated learning and fully supports heterogeneous network architectures and multi-type edge devices. Extensive experiments on the CICIDS2017 benchmark show that FedNIDS-CNN attains a global accuracy of 98.55% and an F1-score of 98.40% while preserving high recall for extremely under-represented attacks such as Heartbleed and Infiltration—outperforming conventional FedAvg-based solutions by a clear margin. With a per-round communication cost only 1/830 that of FedAvg and a single-round training latency as low as 0.31 s on a 1 Mbps edge link, FedNIDS-CNN offers a practical, real-time, and deployable solution for resource-constrained edge infrastructures.

Author Contributions: F.W. Conceptualization, Methodology, Software, Validation, Formal Analysis, Writing—Original Draft, Visualization. W.C. Supervision, Resources, Writing—Review and Editing, Project Administration. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Part of the data is contained within the paper and the rest are available upon request.

Acknowledgments: All authors listed have made a substantial, direct and intellectual contribution to the work and approved it for publication.

Conflicts of Interest: The authors declare that there are no conflicts of interest regarding the publication of this paper.

Abbreviations

The following abbreviations are used in this manuscript:

IoT	Internet of Things
CNN	Convolutional neural network
PCA	Principal component analysis
FedNIDS-CNN	A lightweight intrusion detection model for heterogeneous edge networks based on dynamic distillation-aided federated learning with CNN backbone
IDS	Intrusion detection systems
MFD	Morphological Fractal Dimension
ADASYN	Adaptive synthetic sampling

FL	Federated learning
1D-CNN	One-dimensional CNN
k-NN	K-nearest neighbors algorithm
TP	True Positives
FP	False Positives
FN	False Negatives
TN	True Negatives
ACC	Accuracy
Pre	Precision
F1	Harmonic mean of precision and recall
TPR	True positive rate
FPR	False positive rate

References

1. Shi, W.; Pallis, G.; Xu, Z. Edge computing [scanning the issue]. *Proc. IEEE* **2019**, *107*, 1474–1481. [CrossRef]
2. Agrawal, S.; Sarkar, S.; Aouedi, O.; Yenduri, G.; Piamrat, K.; Alazab, M.; Bhattacharya, S.; Maddikunta, P.K.R.; Gadekallu, T.R. Federated learning for intrusion detection system: Concepts, challenges and future directions. *Comput. Commun.* **2022**, *195*, 346–361. [CrossRef]
3. Baldini, G.; Amerini, I. Online Distributed Denial of Service (DDoS) intrusion detection based on adaptive sliding window and morphological fractal dimension. *Comput. Netw.* **2022**, *210*, 108923. [CrossRef]
4. Idrissi, M.J.; Alami, H.; El Mahdaouy, A.; El Mekki, A.; Oualil, S.; Yartaoui, Z.; Berrada, I. Fed-anids: Federated learning for anomaly-based network intrusion detection systems. *Expert Syst. Appl.* **2023**, *234*, 121000. [CrossRef]
5. Yao, D.; Pan, W.; Dai, Y.; Wan, Y.; Ding, X.; Yu, C.; Jin, H.; Xu, Z.; Sun, L. FedGKD: Toward heterogeneous federated learning via global knowledge distillation. *IEEE Trans. Comput.* **2023**, *73*, 3–17. [CrossRef]
6. Mahadik, S.S.; Pawar, P.M.; Muthalagu, R. Edge-federated learning-based intelligent intrusion detection system for heterogeneous internet of things. *IEEE Access* **2024**, *12*, 81736–81757. [CrossRef]
7. Bhavsar, M.H.; Bekele, Y.B.; Roy, K.; Kelly, J.C.; Limbrick, D. Fl-ids: Federated learning-based intrusion detection system using edge devices for transportation iot. *IEEE Access* **2024**, *12*, 52215–52226. [CrossRef]
8. De Carvalho Bertoli, G.; Pereira, L.A., Jr.; Saotome, O.; Santos, A.L.D. Generalizing intrusion detection for heterogeneous networks: A stacked-unsupervised federated learning approach. *Comput. Secur.* **2023**, *126*, 103106. [CrossRef]
9. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy, Funchal, Portugal, 22–24 January 2018; Volume 1, pp. 108–116.
10. Liu, T.; Chen, Y.; Ge, X. Congealed neural network design for uncertain nonlinear spatiotemporal control systems. *Automatica* **2026**, *183*, 112636. [CrossRef]
11. Shen, J.; Yang, W.; Chu, Z.; Fan, J.; Niyato, D.; Lam, K.-Y. Effective intrusion detection in heterogeneous Internet-of-Things networks via ensemble knowledge distillation-based federated learning. In Proceedings of the ICC 2024-IEEE International Conference on Communications, Denver, CO, USA, 9–13 June 2024; pp. 2034–2039.
12. Liu, G.R.; Zhang, W.Z.; Li, X.J. Data Contamination Defense Method for Intelligent Network Intrusion Detection Systems Based on Edge Examples. *J. Comput. Res. Dev.* **2022**, *59*, 2348–2361.
13. Yang, K.; Wang, J.; Zhao, G.; Wang, X.; Cong, W.; Yuan, M.; Luo, J.; Dong, X.; Wang, J.; Tao, J. NIDS-CNNRF Integrating CNN and random forest for efficient network intrusion detection model. *Internet Things* **2025**, *32*, 101607. [CrossRef]
14. Tabassum, A.; Erbad, A.; Lebda, W.; Mohamed, A.; Guizani, M. Fedgan-ids: Privacy-preserving ids using gan and federated learning. *Comput. Commun.* **2022**, *192*, 299–310. [CrossRef]
15. Kumar, D.; Pawar, P.P.; Meesala, M.K.; Pareek, P.K.; Addula, S.R.; Shwetha, K.S. Trustworthy IoT Infrastructures: Privacy-Preserving Federated Learning with Efficient Secure Aggregation for Cybersecurity. In Proceedings of the 2024 International Conference on Integrated Intelligence and Communication Systems (ICIICS), Kalaburagi, India, 22–23 November 2024; pp. 1–8.
16. Tayebi, M.; Kafhali, S.E. Attention-enhanced BiLSTM-ANN framework with CNN-based feature selection for advanced threat detection. *Int. J. Mach. Learn. Cybern.* **2026**, *17*, 52. [CrossRef]
17. Peng, H.; Xiao, Y.; Wu, C. FD-IDS: Federated Learning with Knowledge Distillation for Intrusion Detection in Non-IID IoT Environments. *Sensors* **2025**, *25*, 4309. [CrossRef] [PubMed]
18. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. A Realistic Cyber Defense Dataset (CSE-CIC-IDS2018). In Proceedings of the 11th International Conference on Communications, Security and Information Systems, Bucharest, Romania, 8–9 November 2018.
19. Alsaedi, A.; Moustafa, N.; Tari, Z.; Mahmood, A.; Anwar, A. TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems. *IEEE Access* **2020**, *8*, 165130–165150. [CrossRef]

20. Khan, R.U.; Zhang, X.; Alazab, M.; Kumar, R. An improved convolutional neural network model for intrusion detection in networks. In Proceedings of the 2019 Cybersecurity and Cyberforensics Conference (CCC), Melbourne, Australia, 8–9 May 2019.
21. Lin, Y.; Wang, J.; Tu, Y.; Chen, L.; Dou, Z. Time-related network intrusion detection model: A deep learning method. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Big Island, HI, USA, 9–13 December 2019.
22. Zhong, W.; Yu, N.; Ai, C. Applying big data based deep learning system to intrusion detection. *Big Data Min. Anal.* **2020**, *3*, 181–195. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.