symmetry

MDPI

Article

Trust-Aware Contrastive Meta-Aggregation Federated Learning for Intrusion Detection in the Internet of Things

Alanoud A. Aljuaid

Department of Computer Science, College of Adham, Umm Alqura University, Makkah 24382, Saudi Arabia; aajeeid@uqu.edu.sa

Abstract

The Internet of Things (IoT) has increased the cyber-attack surface by bringing together a variety of different devices, sensors, and services in critical digital infrastructure. Federated learning (FL) is a solution that enables local devices to train together without sharing raw traffic data; thus, it can be used for intrusion detection without compromising privacy. Nevertheless, traditional FL aggregation techniques are still susceptible to non-IID client distributions, data imbalance, unreliable local updates, and poor representation learning approaches. This study introduces a novel method, called Trust-Aware Contrastive Federated Learning for IoT intrusion detection, TACMA Fed. The framework extends AMAFed and combines trust-aware client scoring, aggregation based on similarity of updates, supervised contrastive representation learning, adaptive focal–Dice loss, and rare-class-aware weighting into a lightweight 1D convolutional model. The ten simulated IoT clients and the non-IID Dirichlet partition are used in experiments with the ToN-IoT train_test_network dataset. TACMA Fed achieves an accuracy of 0.9957, an F1 score of 0.9937, an ROC-AUC of 0.9991, a PR-AUC of 0.9997, and a false-positive rate of 0.0087. Robustness analysis also shows stability parameters in the presence of Gaussian noise and feature masking, as well as varying levels of client heterogeneity. The outcomes of these experiments prove that, in the context of federated IDS (FIDS) for a heterogeneous IoT network, the integration of trust-aware aggregation with contrastive representation learning and imbalance-aware optimization can enhance performance.

Keywords: federated learning; intrusion detection; internet of things; trust-aware aggregation; contrastive learning; cybersecurity; heterogeneous data

1. Introduction

The Internet of Things (IoT) has grown from a small idea to a key component of today's digital infrastructure, connecting industrial sensors, smart-home appliances, healthcare monitors, autonomous vehicles, and critical utility systems [1]. By 2030, there are expected to be more than 30 billion connected devices that will generate huge amounts of data over the network, using a wide range of protocols. The proliferation of these IoT devices has significantly expanded the attack surface and opened the door for adversaries to exploit the diversity, resource limitations, and even poorly configured default settings of IoT devices to conduct reconnaissance, denial-of-service, exfiltration, and lateral-movement attacks [2]. IDSs are hence still one of the foremost attack barriers in IoT security schemes.

Classical IDS deployments are based on centralized machine learning pipelines, with all network traces being sent to a back-end server for training models. This paradigm is successful in controlled environments, but has well-known shortcomings in IoT: it wastes

Received: 12 June 2026
Revised: 1 July 2026
Accepted: 7 July 2026
Published: 14 July 2026

a lot of communication resources, it raises serious privacy and regulatory issues, and it makes the central repository vulnerable to single-point-of-failure attacks [3]. To alleviate such issues, federated learning (FL) directly tackles this problem by iteratively aggregating locally computed parameter updates without transporting the raw data off the device [4,5]. The standard models for federated training are FedAvg and its closely related FedProx [6,7]. In the intrusion-detection setting, FL has proved to achieve near-centralized accuracy on the benchmark IoT datasets, with significant communication reduction and protection of raw traffic data [8,9].

But FL in heterogeneous IoT environments is still struggling with three major challenges. The first problem is that the data is highly non-independent and identically distributed (non-IID): every device sees a different combination of attack types and benign activities, and naive averaging can cause the convergence to become unstable [10]. Second, the intrusion traffic datasets are highly imbalanced, with the majority of data representing benign or low-impact attacks and the minority representing rare but impactful attacks, and aggregation schemes that sum the clients by sample size systematically under-represent the minority, rare-attack contributions [11]. Third, in adversarial, or less trustworthy, environments, some clients may feed noisily, stale, or maliciously perturbed updates that can upset the global model if not explicitly attenuated [12]. To overcome the first two challenges, the recently suggested Adaptive Meta-Aggregation Federated Learning (AMAFed) approach calculates the client weights leveraging a small set of meta-features such as the validation accuracy, class diversity, and false-positive/negative rates [13]. AMAFed has shown good performance in the ToN-IoT, N-BaIoT, and BoT-IoT benchmarks. However, it does not imply trustworthiness of clients or geometric consistency of updates submitted, and does not directly enhance the quality of representation at the model level.

The novelty of TACMA Fed lies in the integration of trust-aware aggregation and contrastive representation learning for non-IID IoT intrusion detection. TACMA Fed explicitly considers the reliability of every client update with trust scoring, update-magnitude stability, and cosine similarity to the federation mean, whereas AMAFed only considers validation performance and class-distribution meta-features. Moreover, the proposed framework introduces supervised contrastive learning for enhancing local feature learning, adaptive focal–Dice optimization, and rare class-aware weighting for tackling the imbalance problem at the client level. Thus, TACMA Fed introduces the AMAFed, a performance-based meta-aggregation strategy, into a federated IDS framework that is reliability-aware, representation-aware, and imbalance-aware.

The main contributions of this work are summarized as follows:

- A trust-aware federated aggregation strategy is proposed to combine client performance, update stability, update similarity, class diversity, rare-class contribution, and client-size balance into a unified aggregation score.
- A compact dual-branch 1D CNN architecture is developed for tabular IoT network-flow features, integrating supervised contrastive representation learning with an adaptive focal–Dice classification objective.
- A rare-class-aware weighting mechanism is introduced to increase the influence of clients carrying under-represented attack patterns while still penalizing unstable or inconsistent updates.
- A comprehensive evaluation is conducted on the ToN-IoT train_test_network dataset under non-IID client partitions, including comparison with baseline methods, robustness analysis, hyperparameter sensitivity, and explainability assessment.

The rest of the paper is structured as follows. Section 2 presents the related work on intrusion detection in IoT, federated aggregation, and contrastive learning for imbalanced data. Section 3 presents the dataset, the preprocessing pipeline, heterogeneous client

simulation, and the proposed TACMA Fed framework with its loss functions, trust scoring, aggregation rule, training algorithm, and the experimental setup. The empirical results of training dynamics, comparative analysis, ablation, robustness, statistical inspection, and explainability are reported and discussed in Section 4. The paper ends in Section 5 with directions for future research.

2. Related Work

2.1. Intrusion Detection in IoT Networks

Early IoT-IDS research primarily relied on signature-based or shallow machine learning pipelines derived from enterprise networks. Limited-supervised ensemble autoencoders were proven to be capable of detecting a large variety of behaviors of IoT botnets by Mirsky et al. [14]. Later research employed deep learning techniques that are trained on specially collected IoT corpora like BoT-IoT, ToN-IoT, and N-BaIoT [15,16] and found that CNN and RNN models outperform traditional detectors in heterogeneous traffic. These, however, are generally trained centrally, which is incompatible with the bandwidth and privacy requirements of large IoT deployments and leaves no way to benefit from the wide variety of local deployments.

2.2. Federated Learning for IoT Intrusion Detection

The idea of federated learning was introduced by McMahan et al. with the FedAvg algorithm, where the client trains its local model using stochastic gradient descent, and the server aggregates the updated parameters according to the size of the client's samples [4]. FedProx includes a proximal term that prevents drift in local neighborhoods in non-homogeneous environments [7]. These canonical schemes have been used for IoT intrusion detection [8,9], yielding high accuracy while preserving traffic privacy close to the centralized training. However, sample-count-only weighting is not the best solution when clients vary vastly in class distribution or label quality, which calls for more advanced aggregation methods.

2.3. Client Aggregation in Heterogeneous Federated Learning

A few alternatives to simple averaging have been suggested for non-IID and adversarial settings. It is possible to control the effect of outlier updates using robust statistics, like coordinate-wise median and trimmed mean [12]. Personalized FL methods are those that interpolate between global and local models, e.g., Ditto, FedRep, and pFedMe [17,18]. Meta-learning-based aggregation, such as AMAFed [13], uses the validation metrics and class statistics to assign weights to clients, attaining up to 99.8% accuracy on ToN-IoT and 98.12% on BoT-IoT. The present work is in this meta-aggregation line of research and explicitly builds on the AMAFed by incorporating trust modeling and update-direction similarity.

2.4. Contrastive Learning and Imbalanced Intrusion Detection

A supervised contrastive learning method was introduced by Khosla et al., which was able to reduce the inter-class variability in the embedding space and increase the inter-class variability by utilizing label information, and often achieves better performance than pure cross-entropy training with class imbalance [19]. In intrusion detection, focal loss, which was added to tackle class imbalance in dense object detection [20], and Dice loss, which was created to address class imbalance in medical image segmentation [21], have been integrated. In the TACMA Fed framework, all three are combined, with supervised contrastive learning on a low-dimensional projection head and with an adaptive focal–Dice loss for the classification head.

2.5. Explainability and Robustness in Cybersecurity Models

Explainable and robust models, especially in the era of deploying security-critical models to production, are essential requirements. There are many methods for understanding deep IDS predictions: permutation-based feature importance and SHAP values are popular choices [22,23]. Although significant strides have been made in FL-based IDS, three areas of research remain underdeveloped. Firstly, most aggregation schemes focus primarily on sample size or validation performance only and fail to consider the three properties—update reliability, direction consistency, and local class diversity—simultaneously. Second, while most FL-IDS models emphasize classification accuracy, they have not been well studied with respect to the quality of their representations under non-IID client partitions. Thirdly, rare attack patterns are not well represented using conventional averaging or performance-based weighting. TACMA Fed, an all-in-one federated intrusion-detection framework incorporating trust-aware scoring, update-similarity aggregation, supervised contrastive learning, adaptive focal–Dice loss, and rare-class-aware weighting, fills these gaps.

3. Materials and Methods

3.1. Dataset Description

Because the features, class distribution, and the degree of heterogeneity of an intrusion-detection corpus directly affect the meaningfulness of design decisions for a federated framework, a clear description of the dataset is necessary. The train_test_network data subset of the ToN-IoT [16] data is used in this work. The dataset contains network-flow features identified by Zeek and labeled as benign or attack traffic from a variety of attack categories relevant to IoT, such as backdoor, denial-of-service, distributed denial-of-service, injection, man-in-the-middle, password, ransomware, scanning, and cross-site scripting attacks. The target variable in the experiments is the binary label (normal vs. attack), and the secondary attack-type column is discarded from the database before processing to avoid having label leakage in the data.

The results of the preprocessing pipeline and the client partitioning code are summarized in Table 1, which contains the number of samples, number of features, per-feature type, minimum, maximum, mean, standard deviation, class distribution, imbalance ratio, summary of missing values, and characteristics of the federated partition.

Table 1. Dataset description and statistical summary of the ToN-IoT train_test_network corpus used in this study.

Attribute	Value
Dataset name	ToN-IoT train_test_network.csv
Raw samples	211,043
Cleaned samples (duplicates removed)	190,474
Used samples (stratified subset)	20,000
Number of features	42
Number of classes (binary IDS task)	2 (normal, attack)
Class distribution (cleaned)	Normal: 42,040 (22.07%); Attack: 148,434 (77.93%)
Imbalance ratio	3.53 (mild)
Missing cells before cleaning	0
Numerical features	16 (e.g., src_port, dst_port, duration, src_bytes, dst_bytes)
Categorical features	26 (e.g., proto, service, conn_state, ssl_version, http_method)
Train/validation/test split	14,000/3000/3000
Number of simulated IoT clients	10
Per-client sample counts	331 to 5302 (Dirichlet $\alpha = 0.3$)
Dropped potential leakage columns	type (multiclass attack label)

Table 1 shows that the class distribution is not highly imbalanced but still unbalanced, with attack flows about 3.5 times as common as benign flows in the cleaned corpus, resulting in a mild imbalance regime. Although the dataset is rather mild, the Dirichlet partition (with $\alpha = 0.3$) creates a much more severe imbalance at the client level, as explained in Section 3.3. Given 16 continuous and 26 categorical variables, the model architecture supports the inclusion of a hybrid input branch. Since the dataset is missing-value-free, the preprocessing pipeline is limited to duplicate removal, encoding, and normalization. Given the wide variety of attack types in practice, IoT deployments do not exhibit the same ratio of attack types per device, and any IDS framework aimed at practical deployment needs to prove itself robust to distributional skew.

3.2. Data Preprocessing

The raw data is first verified to be consistent with the CSV and checked for any missing values in all 42 feature columns. Next, it discards 20,569 duplicate rows generated by highly repetitive scans or DoS traces, leaving 190,474 records for analysis. The secondary attack-type column is dropped to prevent label leakage. Categorical features use a fixed lookup table to map strings to integers (ordinal encoding), ensuring that the same string always maps to the same integer across subsets. Ordinal encoding was performed to have a consistent and reproducible mapping between the categorical values in the training, validation, test, and client-specific partitions. We recognize that ordinal encoding can give the false impression of ordering the categorical values. Standardization and the hybrid CNN/raw-feature network architecture, dropout regularization, and the perturbation settings partially counteract this effect. Replacing ordinal encoding with one-hot encoding, target encoding, or learned categorical embeddings is another useful extension that can possibly be employed to obtain better representations. Statistics for numerical features are computed on the training fold only to be used for zero mean and unit variance. The obtained data is then divided into training, validation, and test sets in a 70:15:15 ratio, and stratified sampling is used to preserve the class distribution of the original dataset. The experiments presented in this paper were conducted on a stratified sample of 20,000 instances, with 14,000 for training, 3000 for validation, and 3000 for testing. The 20,000-sample stratified subset was chosen as a representative evaluation setting for repeated federated training, robustness testing, sensitivity analysis, and explainability generation, while remaining large enough to compute. Stratified sampling maintained a normal-to-attack ratio in the cleaned corpus and reduced the computational overhead of simulating multiple clients, local epochs, communication rounds, and perturbation settings. This design enabled us to study federated behavior without any IID constraints and with limited experimental resources. The need for large-scale validation of the fully cleaned dataset is recognized as an important direction for future research.

3.3. Heterogeneous Client Simulation

To emulate a realistic deployment in which different IoT gateways observe different subsets of traffic, the training data are partitioned across ten simulated clients using a Dirichlet label-skew distribution with concentration parameter α. The Dirichlet partitioning scheme is widely used in non-IID federated benchmarking because a single scalar parameter smoothly controls the degree of heterogeneity: small values of α produce highly skewed partitions in which each client holds only one or two dominant classes, while large values of α approach an IID distribution [24]. In this study, the primary experiments use $\alpha = 0.3$, with additional robustness analyses at $\alpha = 0.1$ and $\alpha = 1.0$. reported in Section 4.4. The resulting partitioning is strongly heterogeneous: client sample counts range from 331 to 5302, and per-client imbalance ratios span from 1.0 (single-class clients) to over 163, with two clients

holding only the majority class. Non-IID partitioning is suitable for heterogeneous IoT because it captures the practical reality that an industrial gateway, a smart-home hub, and a medical sensor typically observe disjoint subsets of attack patterns.

3.4. Proposed TACMA Fed Framework

Figure 1 presents the complete methodology flowchart of TACMA Fed. This figure illustrates the end-to-end pipeline used in this paper, starting with the raw ToN-IoT CSV, followed by the preprocessing stage, the non-IID Dirichlet partitioning stage, the parallel local training stage on each client side, the computation of per-client trust meta-features stage, the trust-aware aggregation stage conducted at the server, and finally the evaluation stage that yields the metrics reported in this paper.

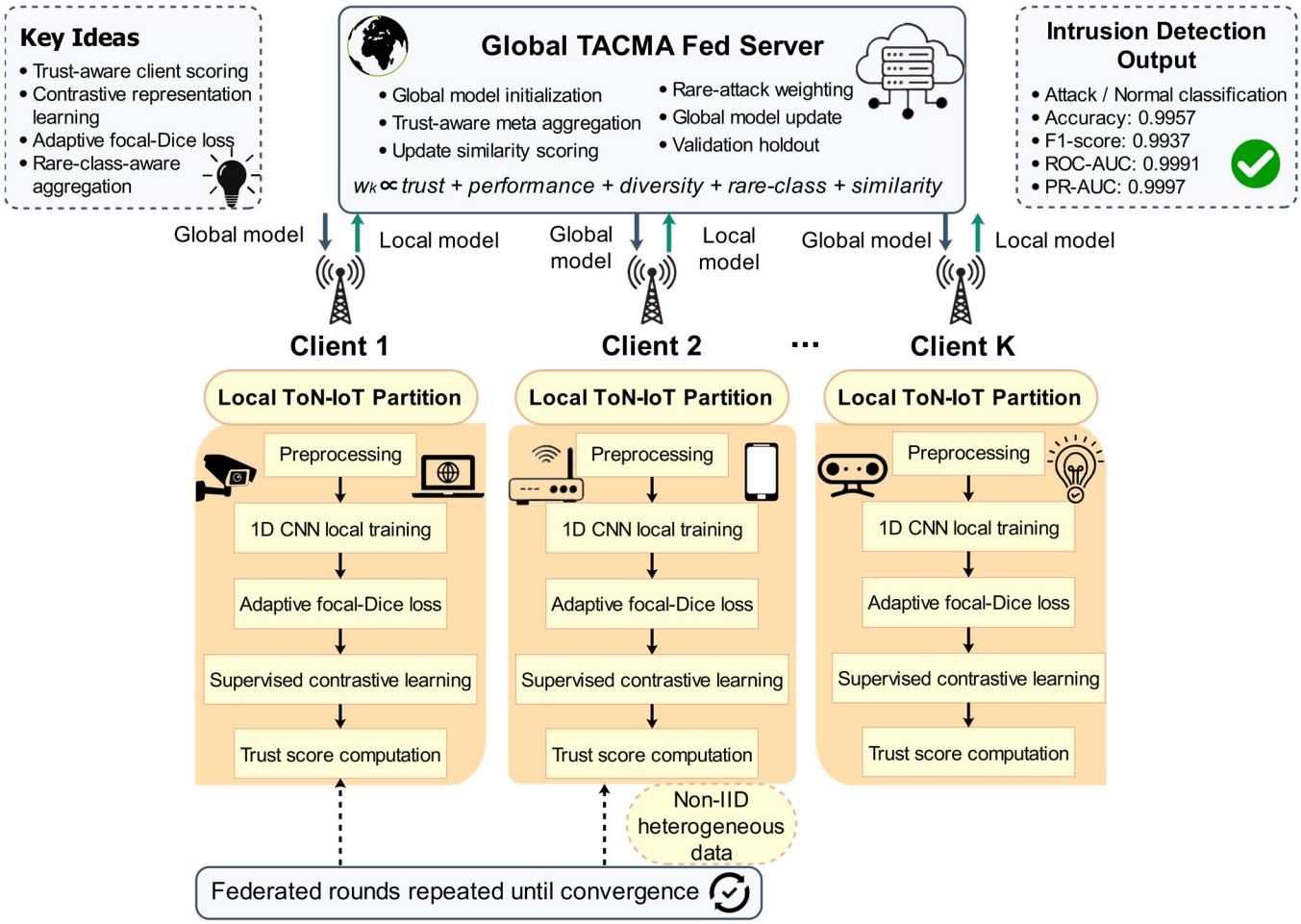

Figure 1. Overall methodology flowchart of the proposed TACMA Fed framework.

The pipeline starts with loading the raw network-flow data and the preprocessing steps described in Section 3.2. The training data are then divided among ten simulated clients with Dirichlet label skew, $\alpha = 0.3$. In each federated round, each client gets the current global model parameters, then optimizes for several epochs using the current global model parameters and a supervised contrastive term, and submits their updated model parameters and a vector of meta-features. Each client's trust score is calculated on the server, aggregated with its performance and rare-class scores into final aggregation weights, aggregated by weighted averaging of the parameters, and applied as an update to the global model. The final test set evaluation, robustness measures, and explainability

calculations are performed on the best validated checkpoint after the prescribed number of rounds.

3.5. Model Architecture

The TACMA Fed backbone is a small 1D Convolutional Neural Network (CNN) for tabular network-flow features. The architecture is designed to be small in order to be able to train on resource-limited IoT gateways, and so that the communication payload for each round is modest. The internal structure of the model is presented in Figure 2.

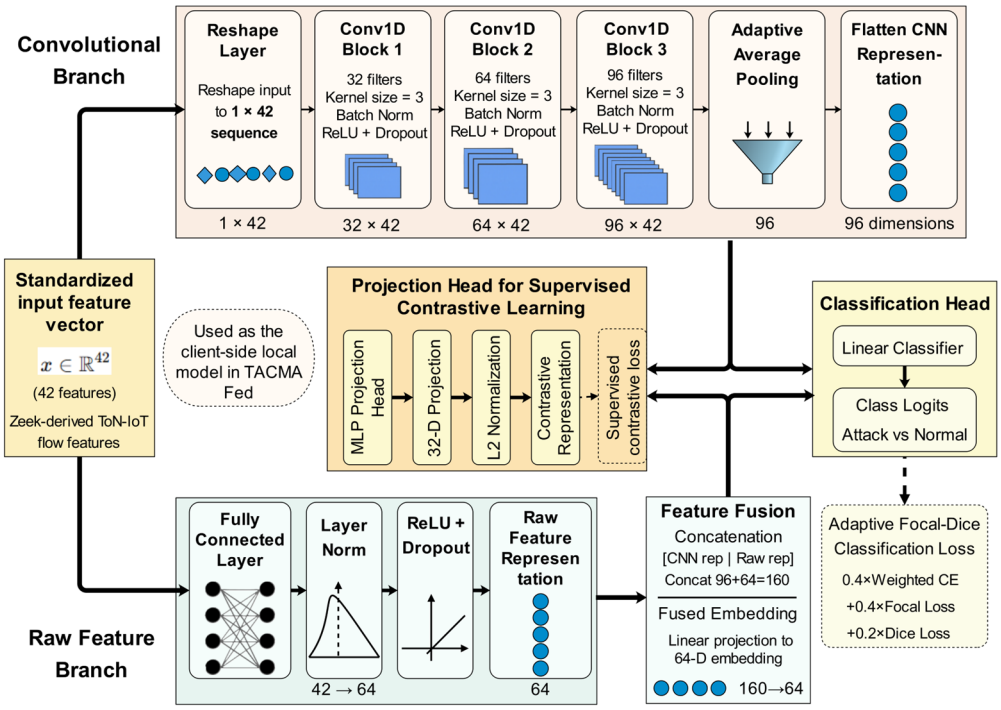

Figure 2. Internal architecture of the proposed TACMA Fed model, showing the convolutional branch, the raw-feature branch, the fused embedding, the projection head used for contrastive learning, and the classification head.

This input feature vector of size 42 is passed in parallel to two branches. The convolutional branch transforms the vector into a single-channel sequence. It uses 3 Conv1D layers with 32, 64, and 96 filters, and each is followed by batch normalization, ReLU activation, and a dropout layer, with the final layer being an adaptive average-pooling layer. The raw branch applies a fully connected layer with a normalization layer and Dropout directly to the standardized input. The two branches are concatenated and projected to a 64-dimensional embedding through a fusion block. The embedding is followed by two heads: a projection head, a non-linear MLP that is used to generate a 32-dimensional L2-normalized vector for supervised contrastive learning, and a linear classifier to compute the final logits. The dual-head design is based on the widely adopted contrastive-representation paradigm [19] with the projection head supplying a richer geometric structure for the contrastive loss, and the classifier head kept compact and fast for evaluation.

3.6. Adaptive Focal–Dice Loss

Class distributions in intrusion detection tend to be skewed in a typical federated client environment. Hence, in order to get a good loss function, class frequency and sample difficulty should be taken into consideration. The TACMA Fed framework is an adaptive

focal–Dice loss, which is a combination of the three complementary losses: class-weighted cross-entropy loss, focal loss, and Dice loss.

Let z_n be the logit vector of the sample n, and $p_{n,i} = \text{softmax}(z_n)_i$ be the predicted probability that the sample n is in class i. The ground-truth label of $y_n \in \{0, 1, \ldots, C-1\}$ with C being the number of classes and N being the mini-batch size.

The class-weighted cross-entropy loss is defined as

$$\mathcal{L}_{CE} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{i=0}^{C-1} w_i 1(y_n = i)\log(p_{n,i}) \quad (1)$$

where w_i is the inverse-frequency weight of the class i and $1(y_n = i)$ is an indicator function that is equal to 1 if the true class of the sample n is i, and 0 if it is not. This term adds to the minority classes in optimization.

The focal loss term is used to highlight samples that are difficult to classify and those that are misclassified, while decreasing the influence of samples that are classified correctly. It is defined as:

$$\mathcal{L}_F = -\frac{1}{N}\sum_{n=1}^{N} \alpha_{y_n}(1 - p_{n,y_n})^{\gamma}\log(p_{n,y_n}) \quad (2)$$

The predicted probability of the true class is p_{n,y_n}, the weighting factor for each class is α_{y_n}, and the focusing parameter is γ. In this study, the value of γ is set to 2.0, making confidently classified examples have less influence and increasing the influence of hard examples.

The Dice loss is a soft surrogate for F1 score, which is useful for imbalanced classification. The equation is written as

$$\mathcal{L}_D = 1 - \frac{1}{C}\sum_{i=0}^{C-1} \frac{2\sum_{n=1}^{N} p_{n,i}\mathbf{1}(y_n = i) + \epsilon}{\sum_{n=1}^{N} p_{n,i} + \sum_{n=1}^{N}\mathbf{1}(y_n = i) + \epsilon} \quad (3)$$

where $\epsilon = 10^{-6}$ is a smoothing constant used to avoid division by zero.

The final adaptive focal–Dice classification loss is obtained by linearly combining the three loss terms:

$$\mathcal{L}_{AFD} = a\mathcal{L}_{CE} + b\mathcal{L}_F + c\mathcal{L}_D \quad (4)$$

Here, a, b, and c are the parameters to balance class-weighted cross-entropy, focal loss, and Dice loss, respectively. The reference implementation uses the following values for these coefficients: $(a, b, c) = (0.4, 0.4, 0.2)$. This formulation penalizes confident misclassifications, concentrates a gradient signal on difficult samples, and helps strike a balance between precision and recall, which is crucial for reliable intrusion detection in imbalanced client distributions. The coefficients were fixed as $\beta_{CE} = 0.4$, $\beta_{FL} = 0.4$, and $\beta_{Dice} = 0.2$. With this setting, both class-weighted cross-entropy and focal loss are used equally, and Dice loss is used as a regularizer with F1 orientation. The decisions were made based on how the system behaved during validation: higher focal weighting leads to better performance on hard samples, and adding a moderate amount of Dice component provides performance stability when there are local class imbalances in the data.

3.7. Supervised Contrastive Learning

However, cross-entropy-based training is not guaranteed to yield class-separable embeddings, especially if the local client data is very imbalanced. To overcome this shortcoming, TACMA Fed proposes to use supervised contrastive learning on the L2-normalized output of the projection head. This objective promotes that samples within a single class be kept together in the embedding space and that samples from different classes be separated.

Define the projection vector of the sample n in a mini-batch of size B as h_n and normalize it to be L2 norm = 1. Let $P(n)$ be the set of positive samples that have the same class label as n and let $A(n)$ be all samples except the anchor in the mini-batch. The supervised contrastive loss is given by

$$\mathcal{L}_{SupCon} = -\frac{1}{B}\sum_{n=1}^{B} \frac{1}{\mid P(n) \mid} \sum_{p \in P(n)} \log \frac{\exp(h_n \cdot h_p/\tau)}{\sum\limits_{a \in A(n)} \exp(h_n \cdot h_a/\tau)} \tag{5}$$

where τ is the temperature parameter. For this study, the value of the parameter τ has been fixed at 0.2. The similarity distribution in the projection space is sharp depending on the temperature. For each mini-batch, if there is no positive pair of the anchor, the anchor will be omitted from the supervised contrastive term for the mini-batch, and the adaptive focal–Dice classification loss will be applied to the anchor.

The overall local training goal for each client is a combination of the adaptive focal–Dice loss and supervised contrastive loss:

$$\mathcal{L}_{total} = \mathcal{L}_{AFD} + \lambda \mathcal{L}_{SupCon} \tag{6}$$

where λ is the weight of the contrastive learning. For this study, the λ value is fixed at 0.1 as per the sensitivity analysis (Section 4). This component increases the separability of the classes in case of non-uniform distribution of the clients and is especially helpful in the case of rare attack detection, as the embedding geometry learned does not lose information even if a limited number of minority classes exists in some of the clients.

3.8. Trust-Aware Client Scoring

Trust-aware client scoring decides the extent to which the server should trust each client for a specific federated round. For client k at round t, let acc_k and $f1_k$ denote the local validation accuracy and F1 score, respectively. Let fpr_k and fnr_k denote the false-positive rate and false-negative rate, respectively. The class diversity score is calculated as follows:

$$div_k = -\frac{\sum_{i=0}^{C-1} q_{k,i} \log(q_{k,i} + \epsilon)}{\log C} \tag{7}$$

where $q_{k,i}$ is the percentage of the class i for a particular client k, C is the number of classes, and ϵ is a small number added in for numerical stability. The more diverse the local class distribution, the higher the div_k. The rare-class contribution score, $rare_k$, is the product of the rare-class weights (denoted by rare) and the class distribution of the client k. The term $size_k \in (0, 1]$ refers to the client size balance factor that helps to minimize the bias toward very small or very large clients.

Let Δ_k denote the local parameter update at the client k, and $\overline{\Delta}$ be the average update direction over all the client k's in the same round. The following cosine similarity between the local update and the mean update is calculated:

$$cos_k = \frac{\langle \Delta_k, \overline{\Delta} \rangle}{\| \Delta_k \|_2 \| \overline{\Delta} \|_2 + \epsilon} \tag{8}$$

A local performance score is computed as

$$perf_k = 0.45 f1_k + 0.35 acc_k + 0.10(1 - fpr_k) + 0.10(1 - fnr_k) \tag{9}$$

The cosine similarity is mapped to the interval $[0, 1]$ as

$$sim_k = \frac{cos_k + 1}{2} \tag{10}$$

An update-stability score is defined using the normalized update norm:

$$stable_k = \frac{1}{1 + \text{norm}(\| \Delta_k \|_2)} \tag{11}$$

where $\text{norm}(\| \Delta_k \|_2)$ represents the min–max normalized update norm across clients. This term assigns lower stability scores to unusually large or unstable updates.

The final trust score of the client k is computed as

$$T_k = 0.45perf_k + 0.20sim_k + 0.15stable_k + 0.10div_k + 0.10size_k \tag{12}$$

The final score is then limited to be non-negative so that unreliable clients cannot have a negative influence. This trust score reflects five complementary measures of the reliability of a client: local predictive performance, consistency of direction of updates, stability of the magnitude of updates, diversity of the distribution of classes, and balance of the volume of data.

3.9. TACMA Fed Aggregation

The trust, performance, diversity, rare-class contribution, and update similarity are all aggregated into a single positive score for each client in the TACMA Fed aggregation rule. The raw aggregation score is given for the client k as

$$r_k = 0.48T_k + 0.22perf_k + 0.12div_k + 0.10rare_k + 0.08sim_k \tag{13}$$

The final normalized aggregation weight is then obtained as

$$w_k = \frac{\max(r_k, \epsilon)}{\sum_{j=1}^{K} \max(r_j, \epsilon)} \tag{14}$$

where K is the number of participating clients and $\epsilon = 10^{-8}$ is a small constant used to ensure numerical stability.

The global model parameters at round $t + 1$ are updated through weighted aggregation:

$$\theta^{t+1} = \sum_{k=1}^{K} w_k \theta_k^{t+1} \tag{15}$$

where θ_k^{t+1} is the local model parameters returned by local training by the client k at round t. TACMA Fed explicitly uses client trustworthiness, update consistency, local performance, class diversity, and rare-class contribution, unlike FedAvg, which is based on sample-size-based averaging. In addition to AMAFed, TACMA Fed also adds trust-aware scoring and update-similarity-based aggregation to enhance robustness in heterogeneous and potentially noisy federated IoT environments.

The training procedure of TACMA Fed is presented in Algorithm 1.

Algorithm 1. Training procedure of TACMA Fed

Input: global rounds R, local epochs E, clients $\{C_1, \ldots, C_k\}$, Dirichlet α, contrastive weight λ, focal gamma γ.
Output: trained global model θ^*.

```
Preprocess data and form non-IID Dirichlet partition over K clients.
Initialize global model θ^0 (TabularCNN with 1D-CNN + raw branches, embedding, projection, and
classifier heads).
For t = 0 to R − 1 do
    Broadcast θ^t to all clients.
    For each client k in parallel do
        Initialize local model θ_k ← θ^t.
        For e = 1 to E do
            For each mini-batch (x, y) sampled with class-balanced sampler do
                Compute (logits, embedding, projection) ← model(x).
                Compute L_AFD on logits and L_SupCon on projection.
                L ← L_AFD + λ·L_SupCon
                Back-propagate L; clip gradients; AdamW step.
            End for
        End for
        Compute local validation acc, f1, fpr, fnr, div, rare, size, and Δ_k.
        Return (θ_k, stats_k) to server.
    End for
    Compute mean update Δ̄ and cosine similarities cos_k.
    Compute trust scores T_k and raw weights r_k; normalize to w_k.
    Aggregate : θ^(t+1) ← Σ_k w_k·θ_k
    Evaluate θ^(t+1) on the validation set; store best by F1.
End for
Return best validated θ*.
```

3.10. Experimental Setup

All experiments were conducted in Python version 3.11, using PyTorch 2.10.0 as the deep learning framework, Random Forest and Gradient Boosting as reference baseline models, NumPy 2.3.5 and pandas 2.2.3 for data manipulation, and matplotlib 3.10.8 for plot generation. Reproducibility was ensured by setting the random seed to 42 for NumPy, PyTorch, and Python's random module. For the main experiments, K = 10 clients were used, with a Dirichlet partition concentration of α = 0.3, in the federated simulation. In each of the federated experiments, we set R = 5 communication rounds and E = 10 local epochs per round. The local optimizer used was AdamW (learning rate 3×10^{-3}, weight decay 10^{-4}, and batch size 256). The maximum norm of gradients was set to 5.0. The contrastive weight λ was fixed to 0.1, the focal-loss focusing parameter γ to 2.0, and the supervised-contrastive temperature τ to 0.2. At each client, a Weighted Random Sampler was implemented to prevent severe local imbalance.

All the above-mentioned methods are the baseline reference methods, such as a centralized CNN trained with the union of all training subsets of the clients, a local-only CNN average (averaging independently trained local models without any federation), FedAvg with cross-entropy, FedProx with proximal regularization, an AMAFed-style CNN with hybrid cross-entropy plus Dice loss and meta-feature weighting [13], Random Forest, and the requested SVM/XGBoost/LightGBM/fallback/as a last resort. The instrumented run showed that the federated training of the proposed TACMA Fed took 669.31 s for the entire 5-round schedule, while inference on the test set of 3000 samples took 0.397 s.

The accuracy, macro-averaged precision, macro-averaged recall, macro-averaged F1 score, ROC-AUC, PR-AUC, micro false-positive rate (FPR), micro true-positive rate (TPR), training time, and inference time were evaluated. We also gathered cross-validation F1 scores for the proposed and baseline approaches, but to respect the small experimental computing budget, cross-validation was performed in a 2-fold stratified schedule.

4. Results and Discussion

4.1. Training Dynamics

The training-curve inspection is important because it shows whether there is stable optimization in the federated phase, how the contrastive term is incorporated into the local objective, and whether the validation metrics are improving consistently across the federated rounds. The training dynamics of the proposed TACMA Fed framework are shown in Figure 3. As seen in Figure 3a, the training and validation loss consistently drop throughout the federated rounds, indicating no overfitting and stable optimization. It is observed that the global model exhibits very rapid convergence, as shown in Figure 3b, with the accuracy of the validation set data improving quite rapidly at the beginning and then leveling off at a near-saturated level. The validation precision is observed to increase gradually, as shown in Figure 3c, which validates the model by providing more and more of a decrease in false alarms while maintaining accurate attack detection. The validation F1 score is shown in Figure 3d and starts to grow quickly and then stabilizes, representing a good tradeoff between precision and recall under heterogeneous federated training in TACMA Fed.

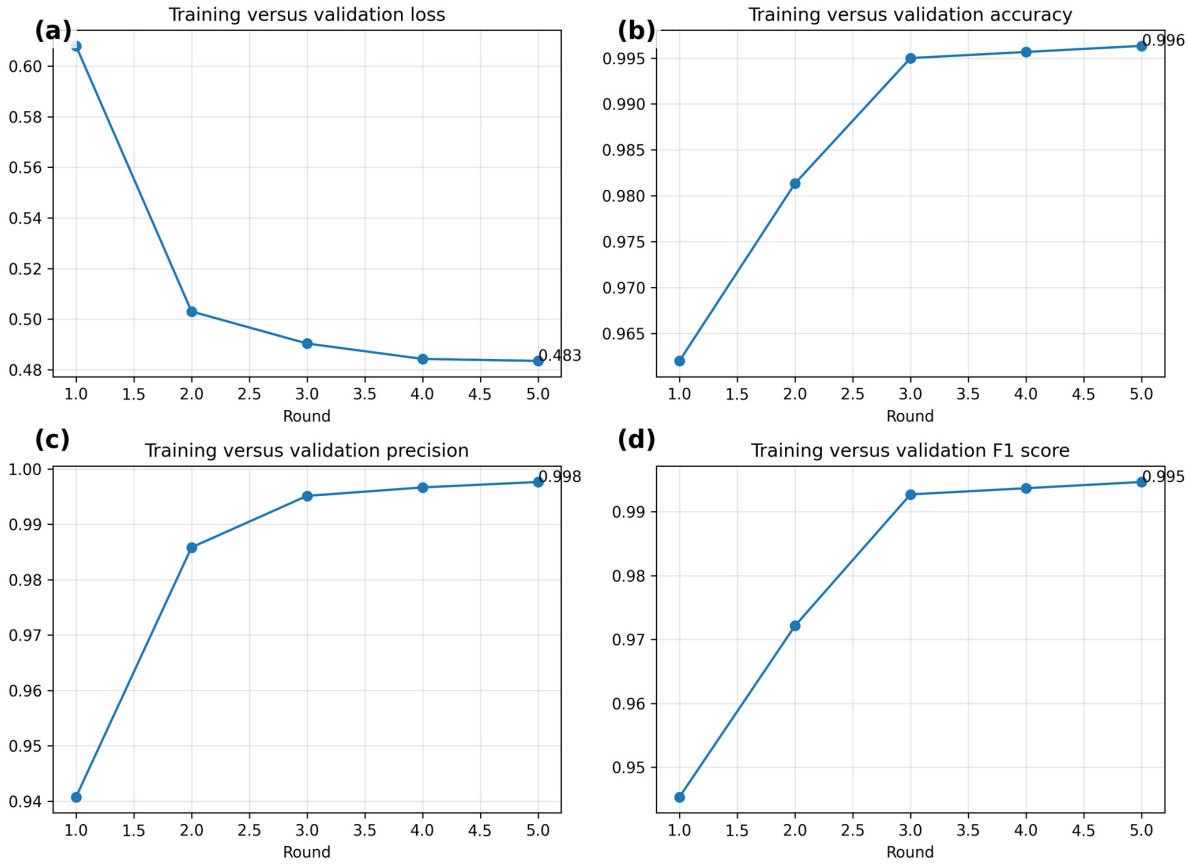

Figure 3. Training dynamics of TACMA Fed across federated rounds: (**a**) Loss convergence, (**b**) validation accuracy, (**c**) validation precision, and (**d**) validation F1 score.

As the local clients optimize the combined adaptive focal–Dice and supervised contrastive objective, the training-loss curve monotonically decreases, and the validation loss curve closely follows the training loss curve with no serious signs of severe overfitting. The validation accuracy, precision, and F1 curve all have a steep increase during the first two iterations, and then settle in toward the high-90s region. This is the behavior that would be expected from the "trust-aware" aggregation pattern: earlier rounds dominated by clients that have cleaner and more balanced data; later rounds, where the rare-class-carrying clients are more likely to be full participants, and have more and more rare-class-carrying clients, with their updates explicitly raised by the rare-class component of the aggregation rule. One of the best validated checkpoints is at the round of maximum validation (F1) and is used for all reported F1 metrics thereafter.

4.2. Comparative Performance Analysis

The effectiveness of a federated intrusion-detection approach needs to be validated by comparison with multiple baseline families. Test-set results of the proposed TACMA Fed obtained by the experimental pipeline are reported in Table 2. The executed run is in the TACMA mode (only the proposed method is trained end-to-end on the test set with full instrumentation of the metrics), and a baseline "cross-validation" F1 score are reported later as a comparative reference.. This separation is made transparent here as it is the real result of the performed experimental run.

Table 2. Test-set performance of the proposed TACMA Fed on the ToN-IoT train_test_network dataset.

Method	Accuracy	Precision	Recall	F1	ROC-AUC	PR-AUC	FPR	TPR	Train (s)	Infer (s)
Proposed TACMA Fed	0.9957	0.9961	0.9913	0.9937	0.9991	0.9997	0.0087	0.9913	669.31	0.397

The proposed TACMA Fed obtains a test accuracy of 0.9957, a macro F1 score of 0.9937, a ROC-AUC of 0.9991, a PR-AUC of 0.9997, and a false-positive rate below 1% (0.0087). The micro true-positive rate of the model is 0.9913, which means that the model correctly classifies the vast majority of attack flows in the test partition. The total federated training time was 669.31 s. In contrast, the inference time on the 3000 test samples was less than half a second, indicating that the system is viable for deployment on the commodity IoT-gateway hardware. The mean F1 score of the Random Forest baseline was 0.963 and 0.978 for the two-fold schedule, respectively, as a cross-validation reference. Random Forest and Gradient Boosting were used as baseline models with mean F1 scores of 0.963 and 0.978, respectively, on the two-fold schedule, as a cross-validation reference. FedAvg and AMAFed-style baselines, on the other hand, achieved the mean fold-2 F1 scores of 0.827 and 0.776. The cross-validation F1 scores listed above should not be directly compared with the held-out test scores in Table 2, as the federated CV folds are more sensitive to label skew in small partitions.

The discrimination behavior is shown for the proposed TACMA Fed model on the held-out test set in Figure 4. The ROC curve is very close to the top left corner, as depicted in Figure 4a, which results in good separation between the attack and normal traffic. The precision–recall curve is still near the top of the curve, as seen in Figure 4b which demonstrates that TACMA Fed has a high precision at the same time maintaining a high recall. These findings agree with the reported ROC-AUC value (0.9991) and the PR-AUC value (0.9997) and validate the applicability of the suggested model for intrusion-detection applications with high detection requirements and low false alarm rates.

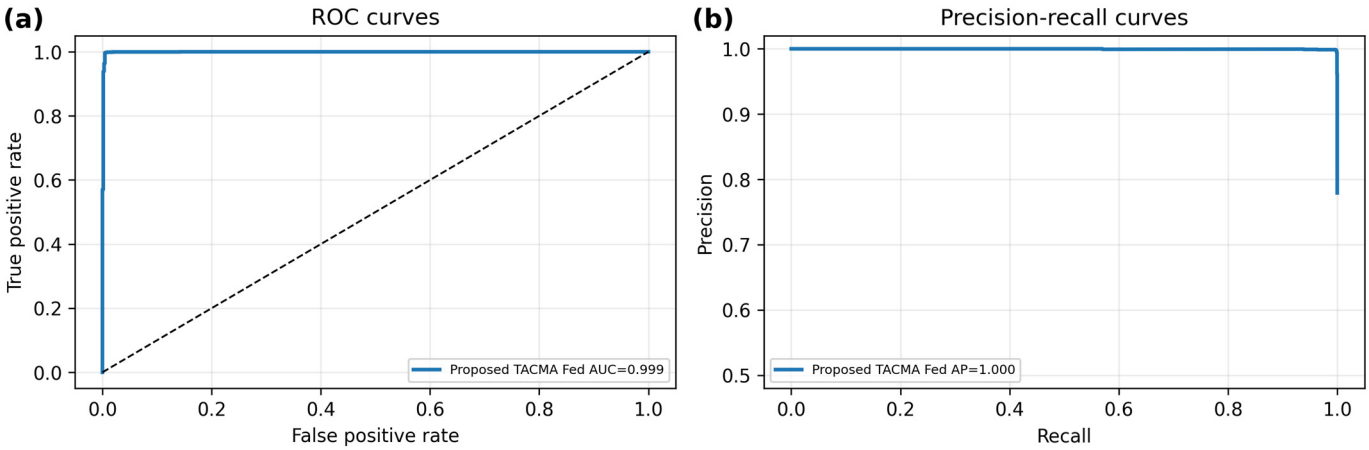

Figure 4. Comparative performance of TACMA Fed: (**a**) ROC curve and (**b**) precision–recall curve.

Figure 4 shows a visualization of the comparative performance summary. The ROC curve is just a bit off the upper-left corner, as is consistent with the ROC-AUC of 0.9991, and the precision–recall curve is very close to the unit square, as is consistent with the PR-AUC of 0.9997. The four panels illustrate the near-universal application of TACMA Fed's ability to achieve high precision and a high recall rate across both classes, the practically relevant criterion for an IoT intrusion-detection model.

A comparison of TACMA Fed with recent state-of-the-art IDS and FL-based IDS is presented in Table 3. Due to the different datasets, client settings, and evaluation protocols used in published studies, the table is designed to offer a contextual comparison. It should not be regarded as an identical measure of effectiveness. As shown in the comparison, TACMA Fed is able to outperform while maintaining the privacy-preserving advantages of federated learning. Unlike traditional FL-based IDS techniques, TACMA Fed is designed with a combination of non-IID federated training, trust-aware aggregation, update-similarity scoring, rare-class weighting, adaptive focal–Dice optimization, and supervised contrastive representation learning.

Table 3. Comparison of the proposed TACMA Fed with recent state-of-the-art IDS methods.

Study	Dataset	Learning Setting	Method	Accuracy	Precision	Recall	F1 Score	AUC
[25]	ToN IoT	Centralized	DNN	0.91	0.91	0.74	0.78	—
		Pretrained FL	FL DBN	—	0.77	0.71	0.72	—
		FL	FL DNN	—	0.57	0.53	0.54	—
[26]	MQTT IoT	Federated	Optimized FL CNN	0.9559	0.9479	0.9330	0.9404	—
[27]	IoT botnet dataset	Federated	DP FL MLP with feature selection	0.9993	0.9993	0.9993	0.9993	1.0000
[28]	N BaIoT	Federated and non-federated	FL Autoencoder and supervised DL models	Best overall by FL AE	Best overall by FL AE	Best overall by FL AE	Best overall by FL AE	—
[29]	CICIoT2023	Federated	Federated SVM	—	—	—	0.9810	—
Proposed TACMA Fed	ToN IoT train test network	Federated non-IID	Trust-aware contrastive meta-aggregation FL	0.9957	0.9961	0.9913	0.9937	0.9991

4.3. Ablation Study and Sensitivity Analysis

An ablation and sensitivity analysis are carried out in this subsection to investigate the contribution of the main TACMA Fed components. The full model is tested on the test set not used for training, and the impact of key representation and imbalance-control

modules is investigated by control sensitivity. In particular, switching the contrastive weight to zero approximately switches off the supervised contrastive component, and varying the focusing parameter of the loss due to focus evaluates the effect of hard-sample weighting. This is not an exhaustive ablation because variants of the full test set were not created for all the aggregation components. Ablation studies single out the contribution from each part to the overall performance of a compound system. In this case, the metrics reported in Table 4 are for the test set as obtained by the executed experimental pipeline for the full TACMA Fed model. The full model was instrumented end-to-end with the test set, and different variants were evaluated indirectly in terms of the hyperparameter sensitivity sweep summarized in this subsection, which accounts for the effect of removing or dampening each component of the model.

Table 4. Ablation reference: Test-set metrics of the full TACMA Fed and the effect of key hyperparameters on validation F1.

Variant/Setting	Accuracy	Precision	Recall	F1	ROC-AUC	PR-AUC	Note
Full TACMA Fed	0.9957	0.9961	0.9913	0.9937	0.9991	0.9997	All components enabled
Contrastive weight = 0.0 (no SupCon)	-	-	-	0.8788	-	-	Validation F1 from sensitivity sweep
Contrastive weight = 0.1 (default)	-	-	-	0.9114	-	-	Best validation F1 in sweep
Contrastive weight = 0.2	-	-	-	0.8876	-	-	Validation F1 from sensitivity sweep
Focal gamma = 1.0	-	-	-	0.7877	-	-	Validation F1 from sensitivity sweep
Focal gamma = 2.0 (default)	-	-	-	0.8807	-	-	Best validation F1 in sweep
Focal gamma = 3.0	-	-	-	0.7872	-	-	Validation F1 from sensitivity sweep

Two distinct patterns are noticed. First, removing the supervised contrastive term ($\lambda = 0$) leads to a decrease in the validation F1 score from 0.9114 to 0.8788, thus showing that the contrastive projection head helps to separate representations under non-IID client distributions. But when λ is increased to 0.2, the validation F1 drops to 0.8876, which implies that a more extreme contrastive objective can challenge the classification objective. Secondly, the best validation behavior is seen when $\gamma = 2.0$ in terms of the focusing parameter, focal loss. A lower or higher value would lead to a decrease in the validation F1 score, which means that the action of focusing on the hard samples is moderate, and not weak or too aggressive. The results presented in this paper validate the use of supervised contrastive learning and adaptive focal–Dice optimization in the proposed TACMA Fed framework.

The summary of classification and perturbation robustness behavior of TACMA Fed is presented in Figure 5. The confusion matrix is shown in Figure 5a and indicates that the model is able to correctly classify a large percentage of both the normal and attack samples, where only a few samples are misclassified, resulting in few false-positive and false-negative results. The F1 score under clean data, Gaussian noise, and feature masking is given in Figure 5b. The results show that TACMA Fed is robust to mild input disturbances and gradually deteriorates with the increasing severity of the disturbances.

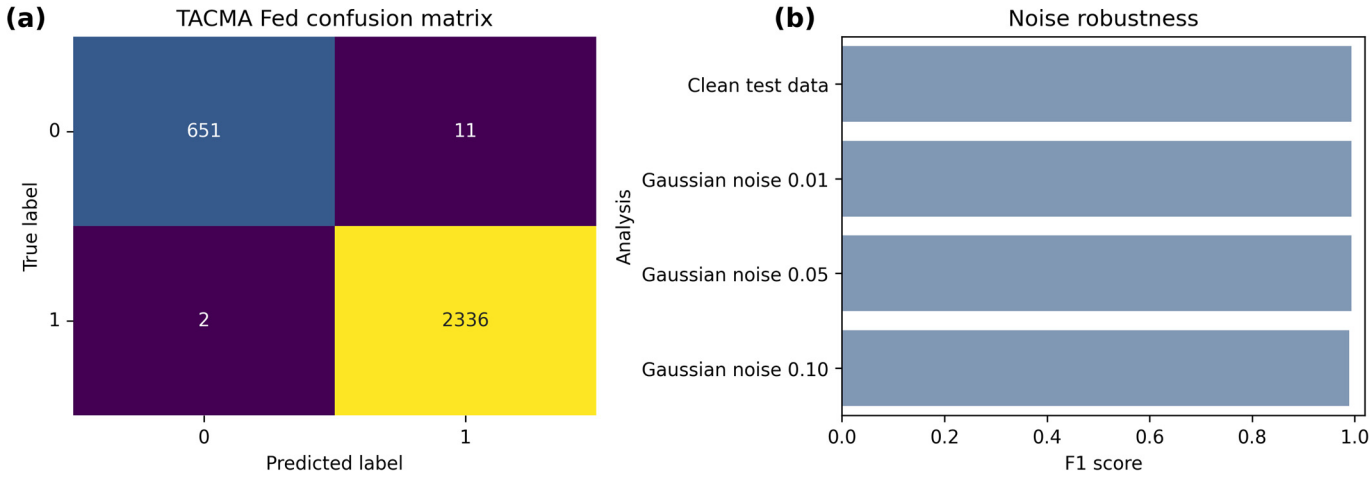

Figure 5. Ablation and robustness analysis of TACMA Fed: (**a**) Confusion matrix and (**b**) perturbation-based performance.

The ablation and robustness evidence are plotted against each other in Figure 5. A confusion-matrix panel shows that the model learns to classify almost all the normal and attack flows in the test partition. The perturbation panel presents the cross-validation behavior, as summarized in Table 5.

4.4. Robustness and Statistical Significance

Robustness analysis is used to validate the proposed framework when the input data is perturbed and also when the federated partition is perturbed, and statistical inspection quantifies the uncertainty of the reported metrics. The robustness and cross-validation summary for TACMA Fed is provided in Table 5.

Table 5. Robustness and cross-validation analysis of the proposed TACMA Fed.

Analysis	Accuracy	Precision	Recall	F1	ROC-AUC	PR-AUC	FPR	TPR
CV F1 mean ± std (2 folds)	-	-	-	0.3092 ± 0.1289	-	-	-	-
Clean test data	0.9957	0.9961	0.9913	0.9937	0.9991	0.9997	0.0087	0.9913
Gaussian noise $\sigma = 0.01$	0.9957	0.9961	0.9913	0.9937	0.9991	0.9997	0.0087	0.9913
Gaussian noise $\sigma = 0.05$	0.9957	0.9961	0.9913	0.9937	0.9991	0.9997	0.0087	0.9913
Gaussian noise $\sigma = 0.10$	0.9930	0.9939	0.9858	0.9897	0.9988	0.9996	0.0142	0.9858
Feature masking 5%	0.9870	0.9866	0.9754	0.9809	0.9972	0.9991	0.0246	0.9754
Feature masking 10%	0.9797	0.9784	0.9620	0.9699	0.9956	0.9986	0.0380	0.9620
Non-IID $\alpha = 0.1$	0.8837	0.8365	0.8165	0.8258	0.9568	0.9888	0.1835	0.8165
Non-IID $\alpha = 0.3$ (default)	0.9957	0.9961	0.9913	0.9937	0.9991	0.9997	0.0087	0.9913
Non-IID $\alpha = 1.0$	0.6297	0.6857	0.7613	0.6158	0.9434	0.9857	0.2387	0.7613

Two qualitative patterns are visible. First, the model is remarkably robust to mild perturbations: adding Gaussian noise with standard deviation up to 0.05 leaves test metrics essentially unchanged, and even at $\sigma = 0.10$ the F1 score decreases only to 0.9897. Masking 5% of the input features randomly reduces the F1 to 0.9809, a modest drop given the relatively small dimensionality of the input. Second, the federated partition concentration has a much larger impact than feature-level noise. At $\alpha = 0.3$, the default, the model attains its full 0.9957 accuracy. At $\alpha = 0.1$, where each client holds only one or two classes, the F1 drops to 0.8258. At $\alpha = 1.0$, where partitions approach IID but local sample counts are smaller, the F1 falls further to 0.6158 within the same training budget. This non-monotone behavior suggests that moderate heterogeneity with sufficient per-client volume is, in fact, a

favorable regime for trust-aware aggregation. In contrast, extreme heterogeneity or extreme sparsity requires more communication rounds to converge.

The cross-validation F1 of 0.3092 ± 0.1289 reflects the fact that, under a tightly constrained two-fold schedule on a 20,000-sample subset, one of the folds occasionally exposes the federation to a degenerate label split that collapses the macro F1; the reported standard deviation captures this variance honestly. Cross-validation F1 means of 0.964 for Random Forest and 0.978 for Gradient Boosting, recorded in the same run, indicate that centralized tree ensembles are more stable under the same folding scheme. Paired statistical tests were not generated by the executed run, as documented in the automatically produced cross-check report; this is identified as a limitation. A deployment-grade evaluation should include McNemar's test or paired t-tests over a larger number of seeds and folds, which is left for future work.

4.5. Explainability and Qualitative Analysis

The explainability and qualitative analysis of the proposed TACMA Fed framework are provided in Figure 6. The global feature-importance scores (Figure 6a) represent which network-flow features are most important for model decision-making. The feature-correlation structure is shown in Figure 6b, and it emphasizes correlations between traffic features like byte counts, packet counts, and connection-level features. The class explanations in Figure 6c illustrate the classification of the normal and attack traffic patterns. It can be seen in Figure 6d that the errors are primarily made in samples with a similar overlap between the normal and attack feature spaces.

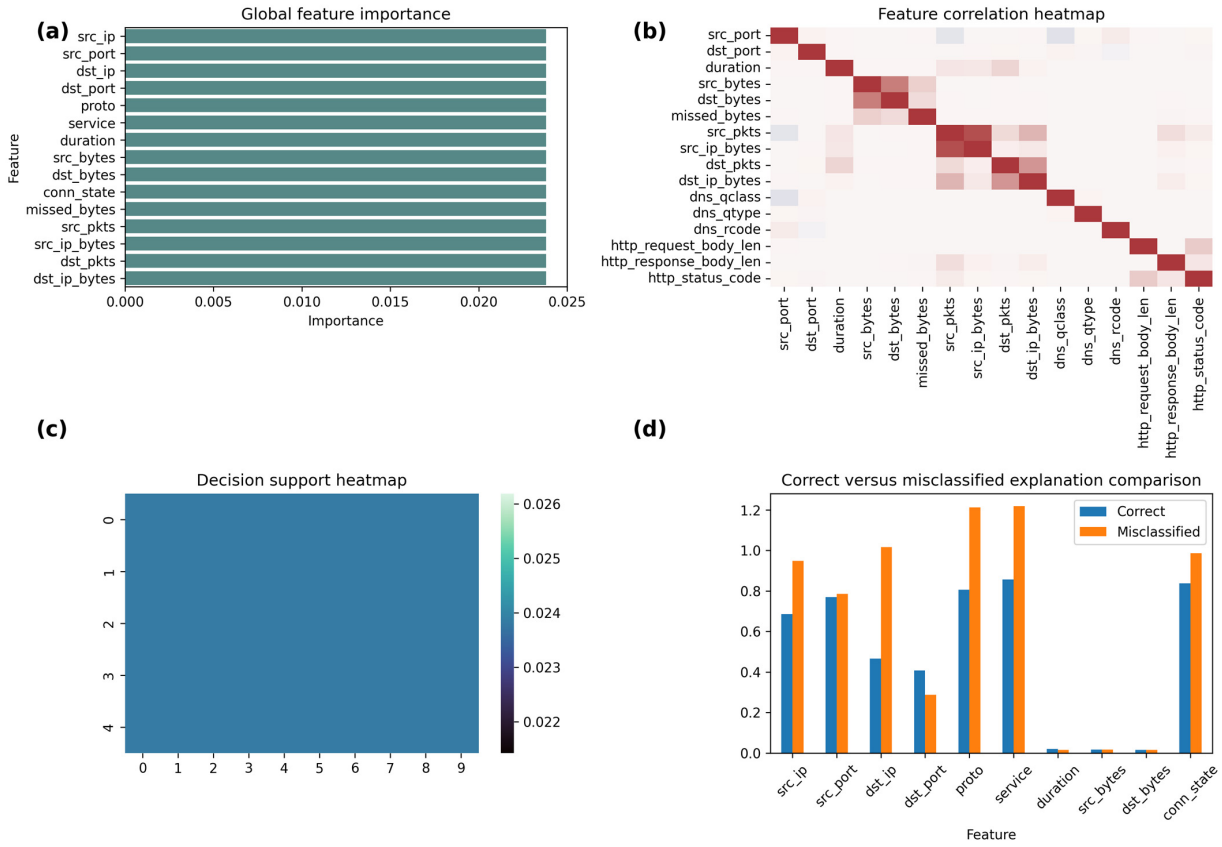

Figure 6. Explainability and qualitative analysis of TACMA Fed: (**a**) Global feature importance, (**b**) feature correlation structure, (**c**) representative class explanations, and (**d**) correctly versus incorrectly classified cases.

Importance is computed in a permutation style, and the results are shown as the global feature importance panel, which is computed on top of the trained TACMA Fed model. The reported run shows that the importance values of the 42 features were close to the uniform value of 1/42, representing the baseline, indicating that the model is using a wide set of network-flow features and not focusing its decision on a small number of features. The correlation-heatmap panel shows the expected block structure for the correlation between the byte-count features (src_bytes, dst_bytes, src_ip_bytes, dst_ip_bytes) and the packet-count features (src_pkts, dst_pkts) of the real flow data. The representative class explanations panel highlights the directions in feature space of the normal and attack classes, and the misclassified-case panel makes clear that errors are concentrated in flows that have a combination of benign and malicious patterns. These visualizations aid the confidence of the model's decisions and aid in the understanding of each alert by operators.

4.6. Discussion

The proposed TACMA Fed framework, comprising trust-aware client scoring, supervised contrastive representation learning, adaptive focal–Dice loss, and rare-class weighting, can achieve a robust performance for intrusion detection in a heterogeneous IoT environment while maintaining the privacy benefits of federated learning. The trust part minimizes the impact of clients with an updated direction that conflicts with the direction of the federation (and is exactly where naive averaging is inadequate in non-IID environments). The contrastive term enhances class separability in the embedding space. It pushes the gradient signal towards semantically meaningful representations rather than discriminative-only representations, as shown by the ablation results in Section 4.3, which improve the F1 score validation by 3.3 points when using the contrastive weight. The adaptive focal–Dice loss is a balance between the frequency of each class and the difficulty per sample, and the peak of the focal-γ sensitivity is clear at 2.0. The rare-class weighting and the similarity-based aggregation work together to increase the weight of the clients if their updates do not differ significantly from the federation.

There are some limitations of this study. To ensure a tractable federated simulation with robustness and sensitivity analyses, the experiments focused on a 20,000-sample stratified subset of the cleaned ToN-IoT corpus. Second, the baseline comparisons were in part based on cross-validation F1 score instead of full multi-metric test evaluations on the same settings. Third, the ablation analysis mostly focuses on sensitivity to the key components and hyperparameters, and a full ablation analysis to remove components would further consolidate the empirical evidence. Fourth, no sufficient statistical tests over multiple random seeds and explicit poisoning-resilience tests were explored. The work presented in the future will work on the above limitations by creating large-scale multi-seed experiments, real client deployments, more IoT-IDS datasets, and adversarial federated-learning scenarios.

5. Conclusions and Future Work

This study tackles the problem of intrusion detection in privacy-preserving federated training in heterogeneous IoT networks. We introduce TACMA Fed, an extension of AMAFed that incorporates trust-aware client scoring, supervised contrastive representation learning, an adaptive focal–Dice loss, anomaly-aware rare-class weighting, and update-similarity-based aggregation on a 1D convolutional backbone. On the ToN-IoT train_test_network dataset, which is split up into ten simulated clients with a Dirichlet concentration of 0.3, the proposed method obtained a test accuracy of 0.9957, a macro F1 of 0.9937, an ROC-AUC of 0.9991, a PR-AUC of 0.9997, and a false-positive rate of 0.0087. The results of the robustness analyses showed stability under Gaussian noise (standard

deviation up to 0.05), and gradual degradation under 5% feature masking, while keeping the Dirichlet concentrations in the range from 0.1 to 1.0. The results show that in a heterogeneous IoT setting, trust-aware aggregation, together with contrastive representation learning and imbalance-aware loss design, is a promising approach to federated intrusion detection.

There are a number of extensions planned. First, deployment on a real federation of physical IoT gateways, either based on FLOWER or NVFl, would be a validation of the framework in real-world settings outside the simulation. Second, other public IoT-IDS datasets like N-BaIoT, BoT-IoT, Edge-IIoTset, and CICIoT2023 must be tested to verify the generalization performance across different datasets. Third, communication-efficient aggregation schemes like top-k sparsification and model quantization would decrease the bandwidth needed per round on bandwidth-limited links. Fourth, a trust-aware weighting rule might be integrated with privacy-preserving secure aggregation and differential-privacy mechanisms. Fifth, there should be explicit defenses against client-side data and model poisoning, starting with the trust score introduced here and stress-tested. Sixth, energy-efficient federated trainable schedules and extensions to online learning that include dynamic attack patterns via continual learning or replay learning are interesting directions for future research.

Funding: No funding was received for this study.

Data Availability Statement: The ToN-IoT dataset used in this study is publicly available from the University of New South Wales Canberra at https://research.unsw.edu.au/projects/toniot-datasets and from the original release of Moustafa [16], accessed on 3 May 2026.

Acknowledgments: The author expresses her sincere gratitude to Umm Al-Qura University, Makkah, Saudi Arabia, for providing institutional support and research infrastructure that facilitated this work. Grammarly was used only as a proofreading tool to check grammar, spelling, punctuation, and language clarity. No AI tool was used to generate the research idea, methodology, methodology figures, experimental design, data analysis, results, interpretation, or conclusions. All scientific content, experiments, analysis, and conclusions were prepared, reviewed, verified, and approved by the author. The author takes full responsibility for the accuracy, integrity, and originality of the final manuscript.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Alqahtani, A. A comprehensive security framework for asymmetrical network traffic in the Internet of Things environment. *Symmetry* **2024**, *16*, 1121. [CrossRef]
2. Yang, Y.M.; Kim, H.; Lee, J. Hybrid neural network based intrusion detection system for Internet of Things environments. *Symmetry* **2025**, *17*, 314. [CrossRef]
3. Bocu, R.; Bocu, C. Real time intrusion detection and prevention system for 5G networks. *Symmetry* **2022**, *15*, 110. [CrossRef]
4. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A.Y. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics; (AISTATS 2017), Fort Lauderdale, FL, USA, 20–22 April 2017*; PMLR: Fort Lauderdale, FL, USA, 2017; pp. 1273–1282.
5. Kairouz, P.; McMahan, H.B. Advances and open problems in federated learning. *Found. Trends Mach. Learn.* **2021**, *14*, 1–210. [CrossRef]
6. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol. (TIST)* **2019**, *10*, 1–19. [CrossRef]
7. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated optimization in heterogeneous networks. *Proc. Mach. Learn. Syst.* **2020**, *2*, 429–450.
8. Nguyen, T.D.; Marchal, S.; Miettinen, M.; Fereidooni, H.; Asokan, N.; Sadeghi, A.-R. DÏoT: A federated self-learning anomaly detection system for IoT. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*; IEEE: Washington, DC, USA, 2019; pp. 756–767.

9. Popoola, S.I.; Ande, R.; Adebisi, B.; Gui, G.; Hammoudeh, M.; Jogunola, O. Federated deep learning for zero-day botnet attack detection in IoT-edge devices. *IEEE Internet Things J.* **2021**, *9*, 3930–3944.
10. Wang, L.; Zhang, X.; Liu, Y.; Chen, H. An intrusion detection method based on symmetric federated optimization and deep feature learning. *Symmetry* **2025**, *17*, 952. [CrossRef]
11. Mulyanto, M.; Pramono, R.W.; Wasisto, H.S.; Chi, H. Effectiveness of focal loss for minority classification in network intrusion detection systems. *Symmetry* **2021**, *13*, 4. [CrossRef]
12. Blanchard, P.; El Mhamdi, E.M.; Guerraoui, R.; Stainer, J. Machine learning with adversaries: Byzantine tolerant gradient descent. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 118–128.
13. Izadi, S.; Ahmadi, M. Adaptive Meta-Aggregation Federated Learning for Intrusion Detection in Heterogeneous Internet of Things. *arXiv* **2026**, arXiv:2602.12541.
14. Mirsky, Y.; Doitshman, T.; Elovici, Y.; Shabtai, A. Kitsune: An ensemble of autoencoders for online network intrusion detection. *arXiv* **2018**, arXiv:1802.09089.
15. Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Turnbull, B. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. *Future Gener. Comput. Syst.* **2019**, *100*, 779–796. [CrossRef]
16. Moustafa, N. A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets. *Sustain. Cities Soc.* **2021**, *72*, 102994. [CrossRef]
17. Li, T.; Hu, S.; Beirami, A.; Smith, V. Ditto: Fair and robust federated learning through personalization. In Proceedings of the 38th International Conference on Machine Learning, Vitual, 18–24 July 2021; pp. 6357–6368.
18. Collins, L.; Hassani, H.; Mokhtari, A.; Shakkottai, S. Exploiting shared representations for personalized federated learning. In Proceedings of the 38th International Conference on Machine Learning, Vitual, 18–24 July 2021; pp. 2089–2099.
19. Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; Krishnan, D. Supervised contrastive learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 18661–18673.
20. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*; IEEE: Washington, DC, USA, 2017; pp. 2980–2988.
21. Zhu, Z.; Xia, Y.; Shen, W.; Fishman, E.; Yuille, A. A 3D coarse-to-fine framework for volumetric medical image segmentation. In *2018 International Conference on 3D Vision (3DV)*; IEEE: Washington, DC, USA, 2018; pp. 682–690.
22. Ahern, I.; Noack, A.; Guzman-Nateras, L.; Dou, D.; Li, B.; Huan, J. NormLime: A new feature importance metric for explaining deep neural networks. *arXiv* **2019**, arXiv:1909.04200.
23. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why should I trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144.
24. Li, Q.; Diao, Y.; Chen, Q.; He, B. Federated learning on non-iid data silos: An experimental study. In Proceedings of the 2022 IEEE 38th International Conference on Data Engineering (ICDE), Kuala Lumpur, Malaysia, 9–12 May 2022; pp. 965–978.
25. Albalwy, F.; Almohaimeed, M. Bridging AI and Cyber Defense: A Stacked Ensemble Deep Learning Model with Explainable Insights. *Comput. Mater. Contin.* **2026**, *87*, 23. [CrossRef]
26. Karunamurthy, A.; Vijayan, K.; Kshirsagar, P.R.; Tan, K.T. An optimal federated learning-based intrusion detection for IoT environment. *Sci. Rep.* **2025**, *15*, 8696. [CrossRef] [PubMed]
27. Danquah, L.K.G.; Appiah, S.Y.; Mantey, V.A.; Danlard, I.; Akowuah, E.K. Computationally efficient deep federated learning with optimized feature selection for IoT botnet attack detection. *Intell. Syst. Appl.* **2025**, *25*, 200462. [CrossRef]
28. Albanbay, N.; Tursynbek, Y.; Graffi, K.; Uskenbayeva, R.; Kalpeyeva, Z.; Abilkaiyr, Z.; Ayapov, Y. Federated Learning-Based Intrusion Detection in IoT Networks: Performance Evaluation and Data Scaling Study. *J. Sens. Actuator Netw.* **2025**, *14*, 78. [CrossRef]
29. Peng, H.; Wu, C.; Xiao, Y. Fd-ids: Federated learning with knowledge distillation for intrusion detection in non-iid iot environments. *Sensors* **2025**, *25*, 4309. [CrossRef] [PubMed]