Metric Annealed Federated Learning (MAFL) - An Efficient and Hierarchical Approach towards NIDS for IoT Edge Devices

Yeasin Ibne Kadir
dept. Computer Science & Engineering
BRAC University
Dhaka, Bangladesh
yeasin.ibne.kadir@g.bracu.ac.bd

Md. Minhazul Islam Royel
dept. Computer Science & Engineering
BRAC University
Dhaka, Bangladesh
minhazul.islam.royel@g.bracu.ac.bd

Tahmid Bin Haque
dept.Computer Science & Engineering
BRAC University
Dhaka, Bangladesh
tahmid.bin.haque@g.bracu.ac.bd

Andalib Iftakher
dept. Computer Science & Engineering
BRAC University
Dhaka, Bangladesh
andalib.iftakher@g.bracu.ac.bd

Farhan Sadi Enan
dept. Computer Science & Engineering
BRAC University
Dhaka, Bangladesh
farhan.sadi@g.bracu.ac.bd

Amitabha Chakrabarty, PhD
Supervisor & Professor
BRAC University
Dhaka, Bangladesh
amitabha@bracu.ac.bd

I. *Abstract*

The proliferation of Internet of Things (IoT) devices has created unprecedented security challenges for network intrusion detection systems. Unfortunately, the drawbacks of such distribution are emerging at an alarming rate as attackers develop new methods to gain unauthorized access to IoT networks. Over the past few years, the frequency of attacks like Distributed Denial of Service (DDoS) has significantly risen. Current state-of-the-art approaches remain insufficient to provide a comprehensive solution. Numerous studies have been attempted using deep learning architectures and supervised machine learning methodologies for binary and multiclass classification tasks. However, these efforts have been consistently hindered by class imbalance issues within the available datasets, resulting in suboptimal model performance. Moreover, literature on compromised node detection is limited and lightweight machine learning models have a false positive rate of around 30 percent. Deep Learning models perform with high accuracy (99%) in binary classification although they struggle with multi-classifications. Additionally, ubiquitous data-driven approaches are parameter heavy and executed in controlled environments. Given the scarcity of real-world datasets, this study proposes a new Metric Annealed (MA) Federated Learning approach with Flower Framework to represent real-life scenarios and prioritize security. To address resource constraints, a hierarchical CNN-BiLSTM client model is introduced with an adaptive probabilistic client selected MA averaging aggregation strategy. The hierarchical model employs approximately 14,000 parameters, enabling the simultaneous training of both binary and multi-classification heads, while comparable models require in a range of 16,000 and 254,000 parameters. The proposed model achieved an accuracy of 98.8% in binary classification and 81.1% in multiclass classification while maintaining a low false positive rate, achieving efficient performance and resource utilization compared to state-of-the-art DL and FL models in precision, recall, and F1-score. Hyperparameter tuning through systematic optimization identified optimal temperature decay and weight parameters for energy-based evaluation. Comparative results demonstrate that the MAFL model offers superior scalability, privacy preservation, and detection performance compared to centralized IDS approaches, establishing Metric Annealed Federated Learning (MAFL) as a viable and efficient framework for secure intrusion detection in distributed IoT networks.

Keywords: MAFL, Federated Learning, NIDS, Simulated annealing, IoT

II. Introduction

The adoption of Internet of Things (IoT) technology has expanded rapidly over the past decade. Kevin Ashton first used the term “Internet of Things,” or IoT, in his presentation, and from there, IoT has spread into various sectors. Various applications in schools, offices, health care systems, multiple industries, and transportation field have been noted by Vishwakarma et al. [1]. As of 2024, the global industrial Internet of Things (IoT) market was valued at $194.4 billion and is projected to reach $286.3 billion by 2029, growing at an annual rate of 8.1% [2]. Its distributed nature promotes efficiency, scalability, and robustness of operations, makes it an essential component of the modern ecosystem. However, the proliferation of IoT and related distributed systems has amplified security risks, as the frequency and sophistication

of cyber-attacks continue to escalate. These attacks include downloading and spreading malware, command injection [3], Denial of Service (DoS) [4], key extraction attacks, buffer overflow, runtime hardware trojans, IoT Routing Attacks, Bruteforce, and Ransomware. Through these attacks, sensitive data is being compromised, which can affect individuals both socially and financially.

New challenges emerge with every advancement, including the aggregation of parameters from training on non-Identical and Independent Distributed (non-IID) data and the increased data dimensionality, necessitating data compression techniques ([5], [6]).

Additionally, the need to transmit model parameters between central servers and edge devices results in high communication overhead ([6]–[9]). Furthermore, studies by McMahan et al [10] showed that, specific tweaking and adjustments to make the system more efficient. The authors used structured and sketched updates to reduce communication expenses by 98% compared to traditional approaches. Although several experimental studies approached to lessen the tradeoff between accuracy and computational cost, none came up with a concrete solution or addressed any limitations. This highlights the pressing necessity for optimized, resource-aware IDS solutions that can effectively balance detection accuracy and classification performance.

These limitations hinder practical deployment in resource-constrained IoT environments.

The study aims to overcome the limitations of existing IDS approaches by enhancing scalability, minimizing computational and communication overhead, and improving detection accuracy in both binary and multiclass classification tasks. The specific objectives of this research are as follows:

1) Derive thorough data insights by visualizing and exploring benchmark IDS datasets and to apply pre-processing techniques that enable reliable model training while addressing issues of data scarcity and class imbalance.
2) Designing a two-headed IDS model capable of jointly performing binary and multiclass intrusion detection tasks, with attention to reducing false positives and improving compromised node detection.
3) Develop a Federated Learning (FL) client-side deep learning model optimized for resource-constrained IoT devices.
4) Benchmarking the proposed IDS framework against state-of-the-art deep learning (DL) and machine learning (ML) models.
5) To develop and evaluate a novel client selection method based on simulated annealing to enhance training efficiency, particularly under non-Identical data conditions.

An efficient approach is sought to minimize resource consumption, with an emphasis on scalability, communication overhead re- duction, and data processing efficiency. To address privacy concerns, FL will be employed to avoid sharing sensitive data. Additionally, the study aims to tackle non-IID data distribution challenges, ensuring the proposed IDS solution is appli- cable in such scenarios. Achieving more than half of the targeted objectives will be considered as a success of this study.

III. Literature Review

Numerous studies on IoT and distributed systems have been conducted in recent times as the growing uses of tinyML and Artificial Intelligence revolutionize the era. A survey conducted by Vishwakarma [1], presented various advancements in IoT and its application in multiple services with the incorporation of blockchain, wireless sensor network (WSN), machine learning (ML), and big data analysis. General working process of each platform and their unique strengths and weaknesses.

Study conducted by Kim et al [5] on One-Class SVM, Isolation Forest (IF), and Local Outlier Factor (LOF) applied to the ADFA-LD dataset. Various feature extraction methods, including Doc2Vec and RNN-based AE, were used to convert variable-length logs into fixed-size vectors. IF with RNN-DAE achieved the highest AUROC accuracy (0.8708), but the study was limited to system call logs, signifying a requirement for broader features like API calls and network data.

Incorporating feature reduction methods like AutoEncoders (AE) and Principal Component Analysis (PCA) can further improve the efficiency of processing large time series datasets. Other studies on AutoEncoders and PCA have been seen as beneficial approaches as they show significant reductions in computational processing ([11], [12]).

Intrusion Detection Systems (IDS) are progressively utilizing Federated Learning (FL) methodologies to elevate security strategies in decentralized and resource deficient settings. Isolation Forest is used in real-time anomaly detection. Designing a common iForest for all clients is crucial for IID datasets. Xiang et al. [13] modeled FLiForest, an FL integration of simple iForest to provide real-time security to an IoT-edge continuum. Locally, iForest has been trained based on collecting large amounts of multimedia data from the end devices. Transferring the global weights in each iteration, every client creates a complete isolation tree. With multimedia datasets, FLiForest outperforms other isolation forests, indicating that FL integration is impactful in anomaly detection with privacy. Lu et al. [6] discussed some solutions for privacy and security in FLForest, which utilizes isolation forests to detect and exclude malicious updates, enhancing Byzantine-robustness and trustworthiness in FL models.

A wide range of studies ([7]–[9], [14]–[16]) on communication overhead reduction and resource constraints using various parametric optimizations were carried out utilizing scarcity techniques such as minimizing redundant data transmission, reducing overall resource usage ([6], [15]) put forward a quantized representation of model weights using fewer bits by Ternary Federated Averaging (T-FedAvg), a quantization-based protocol to optimize communication in FL, low-rank projection, and the Communication-Efficient Federated Optimization (CEFO) method to compress model parameters and reduce communication bandwidth. It also included some model performance challenges like performance

TABLE I
COMPARISON OF FEDERATED LEARNING AGGREGATION STRATEGIES

Feature / Issue	**FedAvg [18]**	**FedProx**	**FedAdam / FedYogi / FedGrad [19]**
Handling high α	Poor with high α	Handles high α	Robust to high α but slower
Client drops	Not handled	Not handled	Not handled
Non-convex problems	Poor	Better than Adam	Good under heterogeneity
Default learning rate	1.0 [18]	Depends on variance	Tuned penalty coefficient
Gradient privacy	Gradient leakage risk	–	–
Non-IID data	Biased avg, low generalization	Variance-based adjustment	Partial / controlled updates
Client drift	Present [20]	Handles heterogeneity	Penalized drift
Computation	Low, better than SGD	Simple, better convergence	Limited, cross-device
Assumptions	L-smooth, bounded variance [19]	–	Not all clients solve local subproblems

bias and imbalance, generalization, convergence and stability, and knowledge sharing. ([16], [17]) aggregated local updates at intermediate cluster nodes before transmission to the central server, reducing communication overhead while optimizing energy usage. Li et al. [9] Dynamic Weighted Aggregation Federated Learning (DAFL), a dynamic weighted aggregation method to prioritize local models based on detection accuracy and sample size, filtering out underperforming models to prevent degradation of the global model. A. A. et al. [8] also used a dynamic weighted aggregation method. Hard et al. [7] used a Coupled Input-Forget Gate (CIFG) recurrent neural network, which reduced the number of parameters per cell by 25% compared to traditional LSTMs.

IV. DATA DESCRIPTION

A. UNSW-NB15 Dataset

It consists of real and synthetic network traffic generated in a cyber range testbed. The dataset is divided into training and testing subsets as follows:

The dataset consists of 257,673 network flow records with 44 flow-based features, one binary label label, and a multi-class attack label attack_cat. After encoding categorical variables, an additional column attack_cat_labeled is added (total 46 columns). **Class Distribution:**

- Normal instances: 93,000
- Attack instances: 164,673

Categorical Features: proto, service, state, attack_cat Some fuzzy duplicates exist in proto (e.g., arp - narp, sctp - stp, rtp - irtp) which may require normalization.

B. Feature Selection

Given the high feature space of the UNSW-NB15 dataset, this method ensures efficient model training and inference while minimizing redundancy and preserving predictive accuracy. A Chi-squared (χ^2) test was conducted to evaluate the statistical dependence between each input feature and the target variable (attack labels). Features with the highest Chi-squared scores were ranked, and the top 20 most discriminative features were selected.

All top-ranked features have p-values close to 0, indicating strong statistical significance. Features such as sttl, dttl, and ct_state_ttl show the strongest associations with the target labels. Conversely, features such as dwin, swin, trans_depth, response_body_len, ct_ftp_cmd, is_ftp_login, ct_flw_http_mthd, and is_sm_ips_ports had p-values > 0.05 and were deemed not statistically significant.

C. Data Transformation

To prepare the UNSW-NB15 dataset for downstream modeling, several transformation steps were applied.

Normalization: All continuous numerical features were scaled to the range $[0, 1]$ using Min–Max scaling. This ensured that features contributed proportionally to model training and avoided bias from varying magnitudes.

Encoding: Categorical attributes such as protocol type (proto), service (service), and state (state) were transformed using label encoding. For example, the following illustrates label encoding for the binary target and a categorical feature:

Binary classification:

- Number of unique labels: 2
- Unique labels: [0, 1]

Multi-class classification:

- Number of unique labels: 10
- Unique labels: [0, 5, 7, 4, 2, 6, 3, 9, 8, 1]

The encoded multi-class labels correspond to the attack_cat column.

Addressing Class Imbalance: To address severe class imbalance, weighted SMOTE with BorderlineSMOTE and adaptive k-neighbors pre-processing was applied before multi-class training. Using square root compression for hierarchy preservation, minority attack classes were augmented to reach at least 20% of the majority class size while maintaining proportional relationships between attack types.

Sequence Formation

To capture temporal behavior, flow records were reorganized into sliding windows of length 10 (seq_length = 10). Each sequence corresponds to 10 consecutive flows, and the majority label within a window was assigned as the sequence label.

Temporal Ordering

Prior to sequence formation, the dataset was sorted by timestamp to preserve the temporal relationships among flow records.

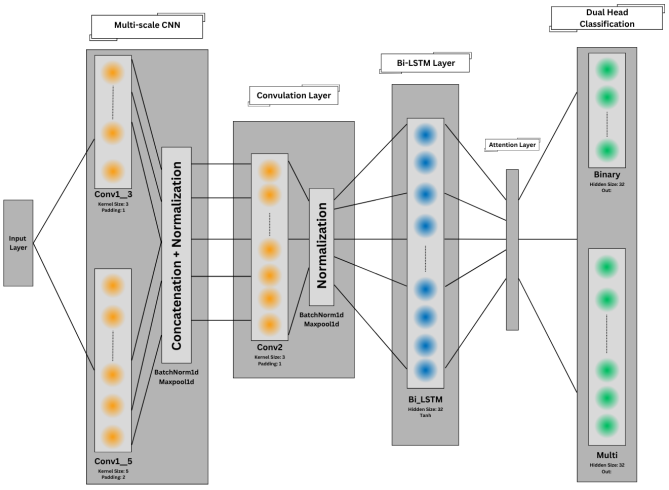

Fig. 1. Proposed hierarchical CNN-BiLSTM model architecture.

Target Handling and Partitioning

For training, the target columns label for binary or attack_cat for multi-class were separated. To simulate a federated learning environment, the dataset was partitioned into multiple client datasets using a Dirichlet distribution with parameter $\alpha = 0.5$.

V. Client Architecture Components

The proposed hierarchical CNN-BiLSTM architecture comprises shared feature extraction layers and dual classification heads for binary and multiclass prediction. The model contains 13,891 total parameters distributed across the following components:

- **CNN Block:** Parallel Conv1D branches with kernel sizes 3 and 5 extract multi-scale temporal features. BatchNorm and dropout ($p = 0.2$) improve generalization, followed by a second Conv1D layer and MaxPooling for dimensionality reduction.
- **BiLSTM Layer:** A bidirectional LSTM with hidden size 16 captures temporal dependencies in both directions. An attention mechanism with 2-layer feedforward network focuses on discriminative time steps using Tanh activation and Softmax weighting.
- **Binary Classification Head):** Three fully connected layers with ReLU activation and dropout ($p = 0.3$) output a single logit for binary classification using BCEWithLogitsLoss.
- **Multiclass Classification Head:** Three fully connected layers with ReLU activation and dropout ($p = 0.3$) output 9 class probabilities using CrossEntropyLoss.

Weight initialization employs Kaiming for convolutional layers, Xavier for linear layers, and orthogonal initialization for LSTM weights to ensure stable convergence. The architecture is illustrated in Fig. 1.

VI. Implementation of Selected Design

This section presents a comprehensive implementation of the proposed **MAFL (Metric Annealed Federated Learning)** framework for hierarchical intrusion detection in IoT edge devices. The implementation covers key components: Three-phase local training procedure, federated learning configuration, and our proposed metric-based annealing aggregation mechanism.

A. Three-Phase Local Training Procedure

Each federated client performed local training using a three-phase curriculum learning strategy designed to address the multi-task nature of hierarchical intrusion detection while preventing catastrophic interference between tasks. The training utilized the Adam optimizer with phase-specific learning rates and incorporated early stopping mechanisms (patience = 3) to ensure efficient convergence.

1) Phase 1: Binary Classification Foundation: The first phase focused exclusively on training the binary classification capability to distinguish between normal and attack traffic. Training parameters included Binary Cross-Entropy with Logits Loss, class weighting (pos_weight $= \frac{\text{num\_normal}}{\text{num\_attacks}}$), Adam optimizer with learning rate $lr = 0.001$, and early stopping with patience of 3 epochs. The model typically converged within 7–10 epochs, with the best performing model checkpointed.

2) Phase 2: Multiclass Attack Type Classification: The second phase trained the multiclass classification head to discriminate among the nine attack types. The model was initialized with weights from the best binary checkpoint. Training used Cross-Entropy Loss with inverse frequency class weights computed after SMOTE augmentation, a new Adam optimizer instance with $lr = 0.001$, and attack samples only with a maximum of 20 epochs. During this phase, all model parameters remained trainable. While the multiclass head learned attack-specific discrimination patterns, the shared feature extractors adapted to capture fine-grained attack characteristics, potentially degrading binary classification performance.

3) Phase 3: Joint Fine-Tuning and Task Balancing: The final phase addressed performance degradation by jointly optimizing both tasks with a combined loss function that re-balanced shared feature representations. Training used a weighted combination loss with adaptive scaling, a new Adam instance with reduced learning rate $(lr \times 0.1)$ for stable fine-tuning, and all samples with both binary and attack type labels for a maximum of 20 epochs.

The joint loss function was computed as:

$$L_{\text{total}} = \begin{cases} L_{\text{binary}}, & \text{for normal samples} \\ L_{\text{binary}} + \lambda \times L_{\text{multiclass}}, & \text{for attack samples} \end{cases} \quad (1)$$

For centralized settings:

$$\lambda = 0.2 + 0.3 \times \min(1.0, \frac{\text{batch\_count}}{100}) \quad (2)$$

For federated settings, the warm-up schedule is reformulated using epoch-normalized progress:

$$\lambda(e, b, B_{\text{client}}) = 0.2 + 0.3 \times \min\left(1, e + \frac{b}{B_{\text{client}}}\right) \quad (3)$$

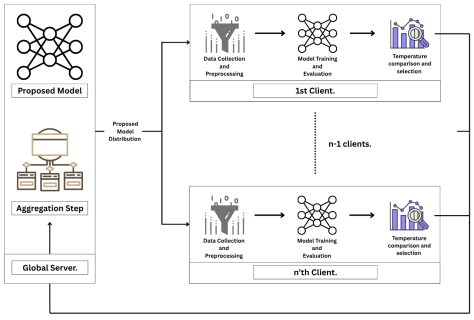

Fig. 2. MAFL pipeline process showing client selection, local training, and metric-based aggregation.

where e is the current local epoch, b is the batch index, and $B_{\text{client}} = \left\lfloor \frac{N_{\text{client}}}{\text{batch size}} \right\rfloor$ is the number of batches per epoch for that client.

The adaptive multiclass weight λ started at 0.2 and increased linearly to 0.5 over the first epoch. The 10× learning rate reduction prevented catastrophic forgetting by allowing small, stable adjustments. The best joint model was saved and sent to the server for aggregation.

The three-phase curriculum provides several advantages: (1) task hierarchy alignment with natural NIDS operation (detect attack $\rightarrow$ classify type), (2) catastrophic interference prevention through fine-tuning, (3) phase-specific class imbalance handling (pos_weight for binary, SMOTE + class weights for multiclass), (4) improved convergence efficiency over joint training, and (5) federated compatibility allowing clients to adapt to unique Non-IID data distributions.

B. Metric Annealed Federated Learning (MAFL) Algorithm

MAFL introduces a novel client selection and aggregation method for federated learning in IoT environments that evaluates clients using a performance-based energy metric. An adaptive temperature parameter with exponential probability function determines client acceptance, allowing low-performing clients controlled participation while ensuring exploration of diverse clients.

The federated learning process followed an iterative client-server model using the Flower framework. During each round, selected clients received the current global model weights, performed the complete three-phase local training procedure on their Non-IID data partitions, and computed local performance metrics. The energy metric E used to evaluate client performance was defined as:

$$E = W_{\text{prec}} \times \text{precision} + W_{\text{rec}} \times \text{recall} + (1 - \text{FPR}) \times W_{\text{fpr}} \quad (4)$$

This metric prioritizes low false positive rates while maintaining detection capability. The best local model weights and metrics were transmitted back to the server for aggregation, ensuring data privacy.

Rather than aggregating all client updates, MAFL uses a probabilistic acceptance rule based on performance metrics. The temperature parameter T controls the exploration-exploitation trade-off, decaying gradually where μ is the cooling rate:

$$T_t = T_{t-1} \times \mu \quad (5)$$

An adaptive adjustment modulates the temperature based on client performance:

$$T_{\text{new}} = T_{\text{global}} \times (1 + k) \quad (6)$$

where rejection_count tracks consecutive rejections:

$$k = e^{\frac{-1}{\text{rejection\_count}}} \quad (7)$$

As rejection count increases, the temperature increases, allowing clients with degraded performance a higher chance of participation in metric aggregation, encouraging improvement in future rounds. This dynamic adjustment maintains fairness in client participation.

The acceptance rule is defined as: (1) **Deterministic acceptance:** if $E_{\text{curr}} > E_{\text{prev}}$, accept; (2) **Probabilistic acceptance:** otherwise, accept with probability

$$P = \exp\left(\frac{E_{\text{curr}} - E_{\text{prev}}}{T}\right) \quad (8)$$

A client is accepted if $P > \text{random}(0, 1)$.

The server aggregates only accepted clients using weighted averaging:

$$\text{aggregated\_weights}[l] = \frac{\sum_i (w_i[l] \times n_i)}{\sum_i n_i} \quad (9)$$

where n_i is the number of samples for client i.

MAFL leverages exponential probability properties—notably, memoryless behavior and exponentially decreasing likelihood with increasing deviation—to govern client acceptance. These properties ensure that even clients with lower performance retain a controlled chance of participation, promoting fairness, exploration, and continuous improvement. By embedding exponential probability into the adaptive temperature mechanism, MAFL effectively balances exploration and exploitation, enhances robustness on heterogeneous Non-IID data, and improves overall model aggregation quality in federated learning for IoT environments.

C. Hierarchical Inference Strategy

During inference, the model employed a two-stage hierarchical approach to minimize computational overhead. The binary classification head first determined whether traffic was benign or malicious using a threshold of 0.5 on sigmoid-transformed logits. Only samples predicted as attacks proceeded to the multiclass head for type identification, as shown in Fig. 3.

This hierarchical inference was implemented in the forward pass using a binary mask, reducing unnecessary computations by approximately 36.1%. Benign traffic was automatically assigned class 0 (normal). The approach mirrors real-world NIDS deployment, where most network traffic is benign.

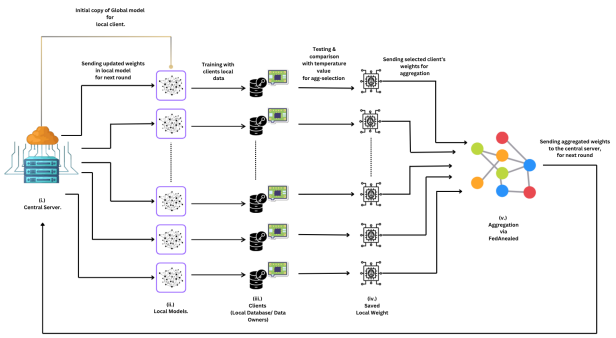

Fig. 3. Hierarchical inference process with binary filtering followed by multiclass classification.

Strategy	Acc.	Prec.	Rec.	F1
MAFL (Proposed)	**0.988**	0.988	**0.977**	0.982
FedAvg	0.980	**0.991**	**0.977**	**0.984**
FedProx	0.979	0.989	0.979	0.983
FedAdam	0.880	0.958	0.849	0.900

TABLE II
BINARY CLASSIFICATION PERFORMANCE COMPARISON

Strategy	Acc.	Prec.	Rec.	F1
MAFL (Proposed)	**0.811**	0.814	**0.796**	**0.804**
FedAvg	0.805	**0.821**	0.785	0.802
FedProx	0.747	0.770	0.747	0.758
FedAdam	0.633	0.586	0.633	0.609

TABLE III
MULTICLASS CLASSIFICATION PERFORMANCE COMPARISON

VII. RESULTS AND ANALYSIS

A. Overall Performance

The proposed Metric Annealed Federated Learning (MAFL) framework with Hierarchical (HR) model achieved 98.8% accuracy for binary classification and 81.1% accuracy for multiclass classification on UNSW-NB15 dataset, maintaining a competitive performance against both centralized and federated baselines. The model maintained a low false positive rate while providing robust scalability across distributed IoT clients.

B. Experimental Comparison

In contrast to traditional FL aggregation strategies such as FedAvg, FedProx, and FedAdam, MAFL demonstrated faster convergence and better generalization under non-IID data conditions.

For binary classification, MAFL improved F1-score by 3–4% over FedAvg and achieved lower communication overhead due to adaptive client selection as demonstrated in Table II.

For multiclass, MAFL achieved a 5–6% gain in precision compared to FedProx, indicating enhanced feature discrimination via the annealing-based aggregation as shown in Table III.

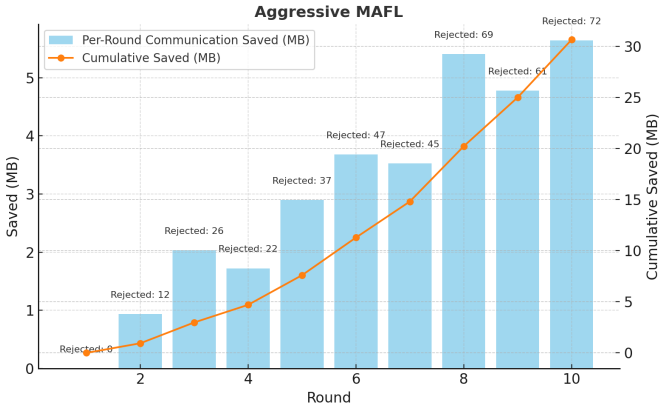

Fig. 4. Communication Overhead Reduction

Method	Acc.	Prec.	Rec.	F1	Params
HR-MAFL	**0.811**	**0.814**	**0.795**	**0.804**	∼14K
CNN-FL [26]	0.744	0.787	0.770	0.778	90K
FcNN [27]	0.690	0.690	0.643	0.639	81K
RFF+KA [28]	0.769	0.752	0.769	0.754	5K
DNN [29]	0.987	0.943	0.960	0.954	35K
ANN [30]	0.756	0.799	0.756	0.765	30K
Enhanced CNN [31]	0.997	0.993	0.994	0.990	29K

TABLE IV
MULTICLASS CLASSIFICATION MODEL COMPARISON

C. Communication Overhead Reduction

Figure 4 illustrates the effectiveness of the proposed MAFL framework in reducing communication overhead compared to FedAvg over ten rounds with 100 clients using UNSW-NB15 dataset. Using the number of parameters transferred as the metric, MAFL achieved significant communication savings through its annealed client selection and adaptive aggregation strategy. With a higher cooling rate ($\mu = 0.009$) promoting aggressive rejection of low-performing clients, the total transmitted parameters per round decreased consistently and overall approximately 60% reduction was achieved. Despite the reduction, the global model's performance remained stable, confirming that MAFL effectively minimizes communication costs without compromising model accuracy.

D. Comparison with Existing Work

Compared to prior FL-based IDS frameworks such as FedGroup, FLForest, and DAFL, the proposed MAFL achieves competitive accuracy with substantially lower communication and computational overhead. Its adaptive metric-annealed aggregation selectively accepts high-quality client updates, reducing bandwidth use while maintaining model stability. With only $\approx$ 14K parameters, MAFL attains 98.8% and 81.1% accuracy for binary and multiclass tasks, respectively—demonstrating superior efficiency for IoT edge deployment. In contrast models that achieves higher performance contains large number of parameters to be trained. Table V and Table IV illustrates the above informations.

Method	Acc.	Prec.	Rec.	F1	Params
HR-MAFL	**0.988**	**0.988**	**0.977**	**0.982**	∼14K
FedPG [21]	0.819	0.828	0.934	0.878	16K
FedPE [21]	0.822	0.810	0.951	0.875	16K
FedPPID [22]	0.920	0.925	0.918	0.921	200K
Hybrid DL-FL [23]	0.971	0.985	0.963	0.974	120K
Unsupervised [24]	0.774	0.712	0.845	0.773	20K
FTTF [25]	0.998	0.996	0.995	0.997	254K

TABLE V
BINARY CLASSIFICATION MODEL COMPARISON

VIII. CONCLUSION AND FUTURE DIRECTIONS

This study introduced Metric Annealed Federated Learning (MAFL), an efficient and privacy-preserving framework for intrusion detection in IoT edge environments. By integrating a hierarchical CNN–BiLSTM client model with a metric-driven annealing aggregation mechanism, MAFL effectively reduced communication overhead and improved model robustness under non-IID data distributions. Experimental results on the UNSW-NB15 dataset demonstrated competitive detection accuracy—98.8% for binary and 81.1% for multiclass classification—while maintaining a lightweight parameter footprint of approximately 14K, confirming the framework's suitability for deployment on resource-constrained IoT devices.

Future work will focus on extending MAFL to handle cross-silo and heterogeneous data environments with stronger privacy guarantees, such as differential privacy and homomorphic encryption. Additionally, optimizing client selection policies through reinforcement learning and expanding evaluation to real-world IoT scenarios will further enhance scalability and practicality. The integration of adaptive compression and asynchronous aggregation strategies also remains a promising direction to minimize latency and further reduce communication costs in large-scale federated networks.

REFERENCES

[1] Vishwakarma, A. K., Chaurasia, S., & Kumar, K. (2024). Internet of things technology, research, and challenges: a survey. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-024-19278-6

[2] Itransition. (2025, February 28). Industrial IoT: market trends & use case statistics. *Online*. https://www.itransition.com/iot/industrial

[3] Victor, H., Kobayashi, S., & Yamauchi, T. (2023). Analyzing post-injection attacker activities in IoT devices: A comprehensive log analysis approach. In *Eleventh International Symposium on Computing and Networking Workshops (CANDARW)* (pp. 292–297). Matsue, Japan. https://doi.org/10.1109/CANDARW60564.2023.00055

[4] Liu, H., & Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. *Applied Sciences*, 9(20), 4396. https://doi.org/10.3390/app9204396

[5] Kim, C., Jang, M., Seo, S., Park, K., & Kang, P. (2021). Intrusion detection based on sequential information preserving log embedding methods and anomaly detection algorithms. *IEEE Access*, 9, 58088–58101. https://doi.org/10.1109/ACCESS.2021.3071763

[6] Lu, Z., Pan, H., Dai, Y., Si, X., & Zhang, Y. (2024). Federated learning with non-IID data: A survey. *IEEE Internet of Things Journal*, 11(11), 19188–19209. https://doi.org/10.1109/JIOT.2024.3376548

[7] Hard, A. S., Rao, K., Mathews, R., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C., & Ramage, D. (2018). Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*. https://arxiv.org/abs/1811.03604

[8] Alamleh, A., et al. (2023). Federated learning for IoMT applications: A standardization and benchmarking framework of intrusion detection systems. *IEEE Journal of Biomedical and Health Informatics*, 27(2), 878–887. https://doi.org/10.1109/JBHI.2022.3167256

[9] Li, J., Tong, X., Liu, J., & Cheng, L. (2023). An efficient federated learning system for network intrusion detection. *IEEE Systems Journal*, 17(2), 2455–2464. https://doi.org/10.1109/JSYST.2023.3236995

[10] McMahan, B., Moore, E., Ramage, D., Hampson, S., & Agüera y Arcas, B. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics* (pp. 1273–1282). PMLR. http://proceedings.mlr.press/v54/mcmahan17a.html

[11] Zeng, Y., Gu, H., Wei, W., & Guo, Y. (2019). Deep-Full-Range: A deep learning based network encrypted traffic classification and intrusion detection framework. *IEEE Access*, 7, 45182–45190. https://doi.org/10.1109/ACCESS.2019.2908G

[12] Merlino, V., & Allegra, D. (2024). Energy-based approach for attack detection in IoT devices: A survey. *Internet of Things*, 27, 101306. https://doi.org/10.1016/j.iot.2024.101306

[13] Xiang, H., Zhang, X., Xu, X., Beheshti, A., Qi, L., Hong, Y., & Dou, W. (2024). Federated learning-based anomaly detection with isolation forest in the IoT-Edge continuum. *ACM Transactions on Multimedia Computing, Communications, and Applications*. https://doi.org/10.1145/3702995

[14] Zhang, Y., Suleiman, B., & Alibasa, M. (2023). FedGroup: A federated learning approach for anomaly detection in IoT environments. In *Lecture Notes in Computer Science*. https://doi.org/10.1007/978-3-031-34776-4_7

[15] Konečný, J. (2016). Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*. https://arxiv.org/abs/1610.05492

[16] Khan, L. U., Saad, W., Han, Z., Hossain, E., & Hong, C. S. (2021). Federated learning for Internet of Things: Recent advances, taxonomy, and open challenges. *IEEE Communications Surveys Tutorials*, 23(3), 1759–1799. https://doi.org/10.1109/COMST.2021.3090430

[17] Baqer, M. (2025). Energy-efficient federated learning for Internet of Things: Leveraging in-network processing and hierarchical clustering. *Future Internet*, 17(1), 4. https://doi.org/10.3390/fi17010004

[18] McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & Agüera y Arcas, B. (2023). Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*. https://doi.org/10.48550/arXiv.1602.05629

[19] Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S., & McMahan, H. B. (2021). Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*. https://doi.org/10.48550/arXiv.2003.00295

[20] Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., & Smith, V. (2020). Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*. https://doi.org/10.48550/arXiv.1812.06127

[21] Nguyen, T.-A., He, J., Le, L. T., Bao, W., & Tran, N. H. (2023). Federated PCA on Grassmann manifold for anomaly detection in IoT networks. In *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications* (pp. 1–10). https://doi.org/10.1109/INFOCOM53939.2023.10229026

[22] Al Amro, S. (2025). Securing Internet of Things devices with federated learning: A privacy-preserving approach for distributed intrusion detection. *Computers, Materials Continua*, 83(3), 4623–4658. https://doi.org/10.32604/cmc.2025.063734

[23] Baidar, R., Maric, S., & Abbas, R. (2025). Hybrid deep learning-federated learning powered intrusion detection system for IoT/5G advanced edge computing network. *arXiv preprint arXiv:2509.15555*. https://doi.org/10.48550/arXiv.2509.15555

[24] Gourceyraud, M., Ben Salem, R., Neal, C., Cuppens, F., & Boulahia Cuppens, N. (2025). Federated intrusion detection system based on unsupervised machine learning. *arXiv preprint arXiv:2503.22065*. https://doi.org/10.48550/arXiv.2503.22065

[25] Elaziz, M. A., Fares, I. A., Dahou, A., & Shrahili, M. (2025). Federated learning framework for IoT intrusion detection using tab transformer and nature-inspired hyperparameter optimization. *Frontiers in Big Data*, 8. https://doi.org/10.3389/fdata.2025.1526480

[26] Wang, L., & Wu, C. (2025). An intrusion detection method based on federated deep learning in complex networks. In L. Chen & A. B. M. Zain (Eds.), *Fourth International Conference on Electronic Information Engineering and Data Processing (EIEDP 2025)* (Vol. 13574, p. 1357405). International Society for Optics and Photonics. https://doi.org/10.1117/12.3067459

[27] Bierbrauer, D. A., Coffey, S. M., Willeke, M. R., Beggs, J. D., & Bastian, N. (2023). Data-efficient federated learning for edge network intrusion detection. *SSRN*. https://ssrn.com/abstract=4926239

[28] Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., & Lloret, J. (2019). Shallow neural network with kernel approximation for prediction problems in highly demanding data networks. *Expert Systems with Applications*, 124, 196–208. https://doi.org/10.1016/j.eswa.2019.01.063

[29] Abhijit, C. S., Jerusha, Y. A., Ibrahim, S. P. S., & Varadharajan, V. (2025). Federated transfer learning for rare attack class detection in network intrusion detection systems. *Scientific Reports*, 15(1). https://doi.org/10.1038/s41598-025-02068-x

[30] Kasongo, S. M., & Sun, Y. (2020). Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. *Journal of Big Data*, 7(105). https://doi.org/10.1186/s40537-020-00379-6

[31] Deshmukh, A., De La Rosa, P. E., Rodriguez, R. V., & Dasari, S. (2025). Enhancing privacy in IoT-enabled digital infrastructure: Evaluating federated learning for intrusion and fraud detection. *Sensors*, 25(10). https://doi.org/10.3390/s25103043