Data Engineering and Applications

Research Paper

A Density-Adaptive DBSCAN-Based Federated Model Training Method for Distributed IoT Terminals

Liviu Kovaşi[1, *]

[1] Faculty of Computer Science, Alexandru Ioan Cuza University of Iaşi, Iaşi, 700506, Romania
*Corresponding author: liviu.ko@fii.uaic.ro

Abstract. Traditional federated learning is vulnerable to client noise, irregular local updates, and unequal device involvement because distributed IoT terminals provide diverse, sparse, and non-independent data streams. For dispersed IoT terminals, this research suggests a density-adaptive federated model training approach based on DBSCAN. Incorporate density-aware client grouping in federated aggregation, identify anomalous update patterns without a set number of clusters, and enhance the client clustering approach using adaptive neighborhood estimation. To lessen the impact of unstable terminals and preserve the helpful edge-side knowledge in the suggested strategy, combine local update statistics, gradient similarity, communication reliability, and data-density indicators. The suggested approach is evaluated experimentally against FedAvg, FedProx, clustered federated learning, Krum, trimmed-mean aggregation, and anomaly-filtered aggregation in the presence of non-independent data, communication fluctuation, terminal dropout, and malicious update disturbance. The findings show that in a heterogeneous IoT context, density-adaptive clustering shortened convergence by 28 communication rounds, increased general accuracy from 89.6% to 94.8%, and decreased aggregation deviation by 37.4%. The study offers a workable federated training paradigm for large-scale IoT networks under challenging communication contexts, non-IID data distribution, and terminal quality fluctuations.

Keywords: *Density-Adaptive DBSCAN, Federated Learning, IoT Terminals, Client Clustering, Non-IID Data*

Received on 09 February 2022, Accepted on 24 May 2022, Published on 06 June 2022

Introduction

The primary computer infrastructure for intelligent manufacturing, intelligent transportation, environmental monitoring, medical health management, energy-saving facilities, and intelligent cities is now distributed Internet of Things platforms. Many terminals are constantly gathering local data; however, this data is frequently private, has a limited bandwidth, and is strongly correlated with the location of the device and the service context. By retaining the original data at the device level and only sharing model updates with an edge or cloud coordinator, Federated Learning provides a workable training paradigm. This distributed learning mode has been extensively researched as a way to produce large-scale Internet of Things (IoT) intelligence, mobile edge networks, and privacy-preserving artificial intelligence [1]. Previous research has demonstrated that federated optimization can facilitate collaborative learning of resource-constrained devices and decrease direct data transfer [2]. Federated learning for dispersed IoT applications may collect sensing knowledge from several terminals without building a central data lake while also protecting privacy by eliminating the transfer of raw data [3].

However, the standard FedAvg is rarely as clean as real-world IoT federated training. Due to varying device locations, user behavior, sensing frequencies, and local conditions at various nodes, terminal data are typically not independent and not equally disseminated [4]. others terminals upload anomalous updates due to sensor drift, local poisoning, hardware malfunctions, or software configuration mistakes; others generate sparse or skewed samples; and some contribute sporadically because of erratic wireless connectivity [5]. The model as a

whole may fluctuate, converge slowly, or directly approach a biased decision boundary as a result of the changes. Client heterogeneity, communication unpredictability, and privacy-security issues must all be handled concurrently rather than individually, according to research on federated IoT and edge learning [6]. Therefore, a robust federated approach for IoT terminals should not presume that all clients supply equally accurate information and instead take into account the structure of local updates prior to aggregation.

A density-adaptive DBSCAN-based federated model training approach for dispersed IoT terminals is examined in this research. Adaptive density clustering is used to distinguish between core clients, boundary clients, and noisy clients in each communication round after each client's update is expressed as the gradient's direction, parameter displacement, local loss variation, data-density evidence, and communication reliability. The number of high-quality update groups will change over the training process; thus, the suggested solution does not need a set number of clusters. To reduce anomalous updates, preserve important heterogeneous data, and enhance the stability of the global model in dynamic IoT contexts, incorporate density-aware client grouping and reliability-weighted aggregation into the architecture.

Related Work

Federated Learning for Distributed IoT Networks

Federated learning, in which clients train local models and a server collects the changes to produce a global model, has been used to investigate privacy-preserving distributed optimization. The following issues are listed in depth in survey studies: adversarial robustness, communication inefficiency, statistical heterogeneity, system heterogeneity, and privacy leakage [7]. Different sensing intervals, battery conditions, processor capabilities, local sample sizes, and wireless link reliability of the terminals make the issues in IoT networks more severe. Additionally, federated learning for mobile edge networks demonstrates that edge deployment must take into account the trade-off between client availability, communication cost, and model correctness [8]. Direct averaging can increase update drift and decrease minority-pattern identification because local targets point in opposite directions, according to non-IID federated learning research [9]. Even if the raw data is stored locally, irregular or malevolent participants may jeopardize the model's integrity, according to privacy and security surveys [10]. The facts suggest that a client-structure-aware aggregation approach, rather than a straightforward arithmetic average update, may be necessary for distributed IoT federated learning.

Density-Based Clustering and Robust Client Selection

Density-based clustering makes sense given the erratic distribution of updates; otherwise, the number of groups would have to be predetermined. Density structure is helpful when the data clusters are non-spherical, unevenly distributed, or contain outliers, according to recent studies of clustering algorithms and density-based clustering techniques [11]. In order to handle task variety and privacy-preserving multi-task learning, client grouping was developed in federated learning [12]. By employing median-type statistics, clipping, or distance-based filtering, robust aggregation techniques further lessen the impact of anomalous local updates [13]. Contribution, dependability, and resource circumstances should have a greater impact on involvement than considering all terminals as uniform workers, according to client selection research [14]. Large-scale IoT implementations require resource-aware and communication-efficient federated learning, according to edge computing studies [15]. Static clustering and fixed-threshold filtering cannot adjust to variations in the density of local updates during training, despite the fact that the research has laid the groundwork. By dynamically calculating the neighborhood radius and density confidence based on the current client update distribution, a density-adaptive DBSCAN technique may be employed to solve this issue [16].

Density-Adaptive DBSCAN Federated Training Method

Federated IoT Training Framework

The proposed framework contains distributed IoT terminals, an edge aggregation server, a global model, and a density-aware client analysis unit. In each communication round, selected terminals receive the current global model and train locally using their private samples. Instead of directly sending only a parameter update, each

terminal also reports a compact training-state descriptor generated from local loss variation, gradient norm, update displacement, sample quantity, and communication quality.

$$u_i^t = \theta_i^t - \theta_g^t \quad \text{Eq.(1)}$$

The gap between the current global model and the terminal model is known as the local update displacement. It's possible that the terminal has valuable, uncommon designs, thus a significant displacement is not always dangerous. However, an unreliable update is frequently indicated by a very big displacement together with unstable loss and poor communication dependability.

$$g_i^t = \nabla F_i\left(\theta_g^t; D_i\right) \quad \text{Eq.(2)}$$

The local gradient summarizes the direction in which each terminal attempts to improve its own objective. For IoT terminals, this gradient can be biased by local context; therefore, the framework compares gradient direction across clients rather than judging a client only by update magnitude.

$$\rho_i^t = \frac{n_i}{1 + \sigma_i^2 + \delta_i^t} \quad \text{Eq.(3)}$$

The density evidence combines local sample size, feature variance, and communication delay. A terminal with more stable samples and lower delay receives a more reliable density score, while sparse or delayed terminals are treated cautiously during clustering.

$$x_i^t = [\cos(g_i^t, \bar{g}^t), \|u_i^t\|_2, \Delta\ell_i^t, \rho_i^t, q_i^t] \quad \text{Eq.(4)}$$

The client update descriptor is constructed as a five-component vector. It keeps the representation compact enough for communication efficiency while retaining the information needed to distinguish normal heterogeneity from abnormal behavior.

$$G^t = \{x_1^t, x_2^t, \ldots, x_m^t\} \quad \text{Eq.(5)}$$

The adaptive DBSCAN unit receives a round-level client-update set created by the server. The relationships between local training, update representation, density clustering, reliability weighting, and global aggregation in the suggested framework are depicted in Figure 1.

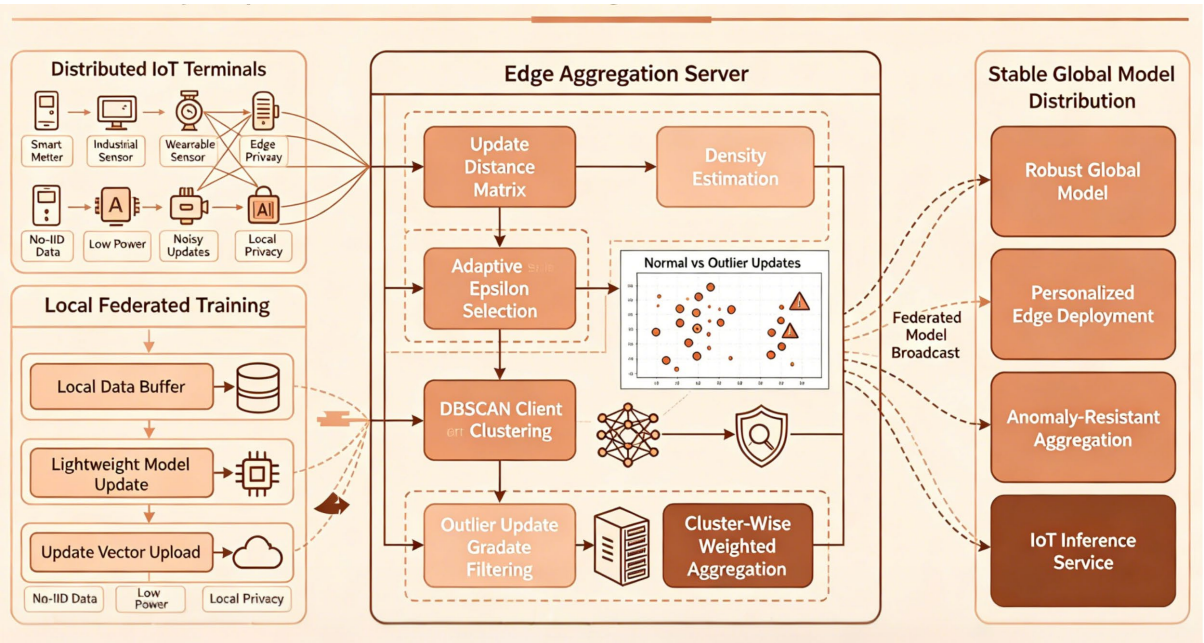

Figure 1. Framework of density-adaptive DBSCAN-based federated model training

Density-Adaptive Client Clustering

The issue of a fixed clustering threshold is handled with a density-adaptive DBSCAN module. Since the local models have not yet developed a solid representation, the clients' updates are first dispersed. In the latter

rounds, atypical consumers stay apart while dependable clients are comparatively focused. The initial updates would either be over-separated or irregular late-stage updates would be over-merged in a fixed-radius static neighborhood.

$$d_{ij}^{t} = \alpha\left(1 - \cos\left(g_i^t, g_j^t\right)\right) + \beta\left\|u_i^t - u_j^t\right\|_2 \quad \text{Eq.(6)}$$

The pairwise client distance combines gradient-direction disagreement and update-displacement difference. This avoids a narrow Euclidean view of model updates and is more suitable for IoT training, where two clients may have similar update sizes but opposite optimization directions.

$$\varepsilon^t = \text{median}_i\left(kNN_k(x_i^t)\right) + \lambda MAD_i\left(kNN_k(x_i^t)\right) \quad \text{Eq.(7)}$$

The adaptive neighborhood radius is obtained from the median k-nearest-neighbor distance and its robust dispersion. The median absolute deviation makes the radius less sensitive to a small number of malicious or unstable clients.

$$\text{MinPts}^{\,t} = \max(3, \lceil \eta \log(1 + m^t) \rceil) \quad \text{Eq.(8)}$$

The minimum density requirement grows slowly with the number of participating clients. This prevents the algorithm from declaring small accidental groups as reliable clusters when many terminals join the same round.

$$N_\varepsilon(x_i^t) = \left\{x_j^t \mid d_{ij}^t \le \varepsilon^t\right\} \quad \text{Eq.(9)}$$

The adaptive neighborhood records which clients are close enough to each other in the update space. A client is considered structurally supported only when enough neighbors share compatible optimization behavior.

$$c_i^t = \mathbf{1}(|N_\varepsilon(x_i^t)| \ge \text{MinPts}^{\,t}) \quad \text{Eq.(10)}$$

Core clients satisfy the adaptive density requirement and are treated as stable representatives of useful local knowledge. Boundary clients may still contain useful rare information, but they require lower aggregation weights.

$$z_i^t = 1 - \exp(-|N_\varepsilon(x_i^t)| / \text{MinPts}^{\,t}) \quad \text{Eq.(11)}$$

The density confidence transforms neighborhood support into a continuous score. Compared with binary filtering, this continuous score allows the server to suppress suspicious clients gradually instead of discarding all boundary information.

Robust Aggregation with Cluster Reliability Weighting

After clustering, a reliability score based on cluster membership, density confidence, local loss improvement, communication reliability, and update consistency is computed for every client. Heterogeneous but structurally sound clients can still be employed, and the first design principle is to prevent an excessive concentration of aberrant clients in the global aggregate.

$$r_i^t = z_i^t q_i^t \exp(-|\Delta\ell_i^t - \text{median}(\Delta\ell^t)|) \quad \text{Eq.(12)}$$

The reliability score decreases when a client's loss behavior deviates sharply from the round median or when its communication quality is poor. This term is useful for distinguishing abnormal optimization from normal heterogeneous learning.

$$\omega_i^t = \frac{r_i^t n_i}{\sum_j r_j^t n_j} \quad \text{Eq.(13)}$$

The aggregation weight normalizes reliability and local sample contribution together. A large local dataset is beneficial only when the corresponding update is structurally credible.

$$\theta_g^{t+1} = \theta_g^t + \sum_i \omega_i^t u_i^t \quad \text{Eq.(14)}$$

The global model is updated through reliability-weighted displacement aggregation. This expression preserves the efficiency of FedAvg but replaces uniform trust with density-aware trust.

$$A^t = \left\| \sum_i \omega_i^t u_i^t - \text{median}_i(u_i^t) \right\|_2 \quad \text{Eq.(15)}$$

Aggregation deviation is monitored to evaluate whether the weighted update remains close to the robust center of client behavior. A lower deviation indicates that abnormal updates have less influence on the global direction.

$$L_g^t = \sum_i \omega_i^t F_i(\theta_g^t) + \mu A^t \quad \text{Eq.(16)}$$

The global optimization objective combines weighted empirical loss with an aggregation-deviation penalty. The penalty discourages an unstable global update even when several clients report confident but mutually inconsistent local improvements.

$$\theta^* = \arg\min_\theta \sum_t L_g^t \quad \text{Eq.(17)}$$

The final model is the parameter state that minimizes the accumulated reliability-aware global objective over communication rounds. Figure 2 shows the internal relation among adaptive density estimation, noise client identification, reliability score calculation, and weighted aggregation.

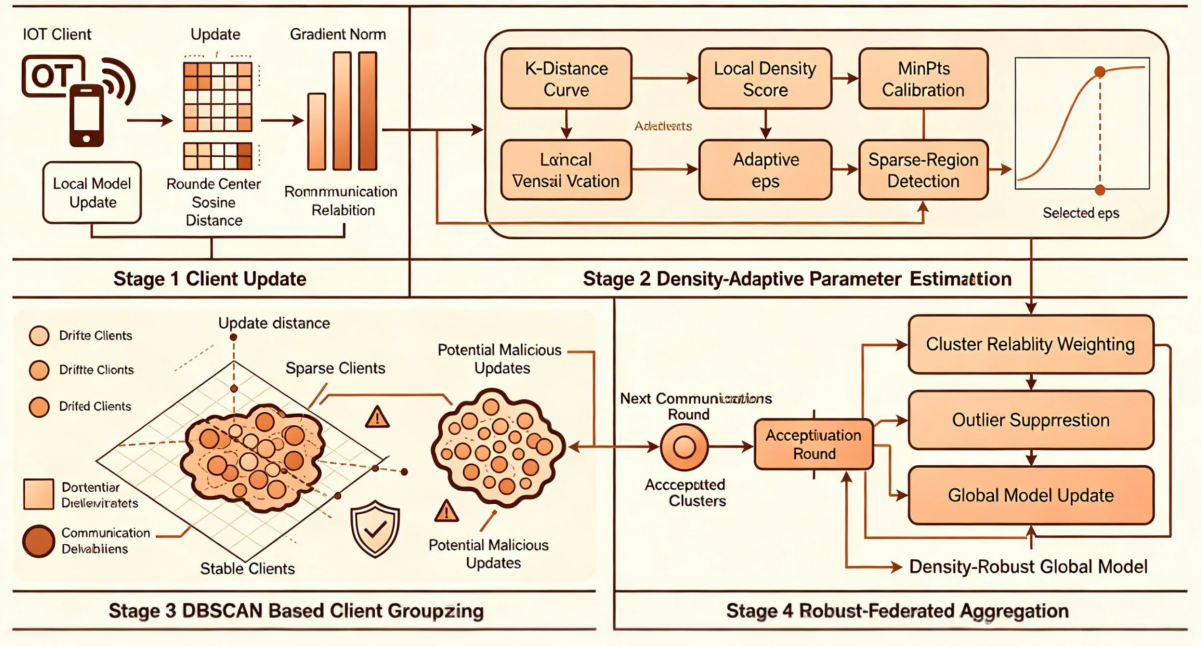

Figure 2. Density-aware client clustering and robust aggregation module

Experimental Scheme and Quantitative Evaluation

IoT Dataset Construction and Training Settings

In the experiment, create 120 virtual terminals using network-status logs, edge-device activity profiles, and smart sensing data to create a distributed IoT categorization scenario. The global job has six types: normal operation, low-load fluctuation, sensor drift, communication instability, edge-device overload, and anomalous terminal behavior. Each terminal contains 400–2,600 local samples. The non-IID partitions have Dirichlet coefficients of 0.2, 0.5, and 1.0. As an inevitable aspect of edge networks, federated learning for IoT applications takes into account the heterogeneity of terminal devices at the edge [17]. 5%, 10%, and 20% of the terminals are set up as unreliable clients that upload skewed gradients, delayed updates, or locally overfitted models in order to assess the aberrant update resistance. IoT security models frequently exhibit unreliable and attack-like local behaviors, according to research on federated anomaly detection [18].

Every model has undergone 160 communication rounds of training. With a batch size of 64, each chosen client completes five local epochs. Adam is the optimizer; exponential decay is used, and the starting learning rate is 0.001. Unless a dropout is carried out, the participation rate is set at 40%. To mimic communication interruption and client dropout, randomly remove 10%–40% of the current clients in some rounds and lose some packets (0%–30%). One RTX 3060 GPU server and CPU-only edge emulation are used in the hardware environment to generate client-side updates. The round number and upload traffic should be taken into account in addition to correctness, according to studies on communication-efficient federated learning [19]. Research on wireless federated learning has also shown that resource optimization is necessary because of the devices' different communication circumstances [20].

Evaluation Metrics and Baseline Methods

Overall accuracy, macro-F1 score, convergence cycles, aggregation deviation, aberrant client recognition rate, communication cost, participation stability, and edge inference latency are the evaluation indicators. Since the majority normal data readily overshadows the minority terminal states, macro-F1 is more suited for non-IID IoT classes. Accuracy indicates the overall accuracy of the predictions. The difference between the robust center of client updates and the actual global update is known as the aggregation deviation. The density clustering unit's ability to distinguish between malicious and noisy update patterns without prior knowledge of the number of abnormal clients is shown by the abnormal client recognition rate. Because the deployment metrics include client-side runtime and communication traffic, the edge network needs both resource viability and model quality.

The metric's design distinguishes between training-process quality and prediction quality. A model is not appropriate for continuous IoT services that need model checkpoints during training because it might deliver unstable intermediate updates while achieving a strong final accuracy. The rate at which the global model approaches a stable validation plateau is shown by convergence rounds. The term "aggregation deviation" describes if a small number of extreme clients dominate the server update. The term "stability of participation" describes the method's continued efficacy following the replacement of the chosen clientele in several rounds. To guarantee that a federated model on dispersed terminals is accurate, stable, communication-efficient, and interpretable at the training-control level, the indicators are evaluated collectively.

FedAvg, FedProx, clustered federated learning, Krum, trimmed-mean aggregation, and anomaly-filtered aggregation are examples of baseline techniques. FedProx is presented to address system and statistical heterogeneity through regularization, with FedAvg serving as the fundamental standard for averaging. Clients are also grouped for various learning tasks using clustered federated learning. The resilience to anomalous updates is evaluated using Krum and trimmed-mean aggregation. Density-adaptive DBSCAN is not used in the baseline techniques to generate round-specific client groups and reliability weights. The observed discrepancies are caused by aggregation behavior rather than model capability because all the techniques employ the identical data partitions, client sampling schedules, training epochs, and model backbones.

Round-level diagnostic factors are also incorporated into the experiment's design. The number of core clients, boundary clients, noise clients, adaptive neighborhood radius, average reliability score, and variation between robust update center and weighted aggregation are these factors. These records are intended to explain why the suggested approach converges differently from the baseline aggregate rather than being provided as additional labels. A strong federated approach should be interpretable at the training-process level; otherwise, a high final accuracy could have been the result of a fortunate random client scheduling rather than learning from consistent client collaboration.

Result Interpretation and Comparative Analysis

Global Training Accuracy and Convergence Analysis

Figure 3 shows the consequences of global convergence. The accuracy curve following 160 communication cycles is shown in Figure 3(a). The suggested approach achieved 94.8% accuracy after 118 rounds, followed by FedAvg (89.6%), FedProx (91.1%), clustered federated learning (92.3%), Krum (90.7%), and trimmed mean (91.4%). Furthermore, there is no improvement in late-stage accuracy. Because certain terminals upload updates from highly skewed local distributions, FedAvg exhibits significant oscillation within the first 60 rounds. The

impact of device heterogeneity on federated aggregation quality is examined in edge resource optimization research [21]. To limit oscillation, the technique groups update descriptors prior to aggregation and lowers the weight assigned to isolated clients.

Macro-F1 is seen in Figure 3(b). FedAvg and FedProx are 87.9% and 89.8%, respectively, whereas the suggested approach is 93.6%. The classes with fewer data and greater distribution differences—sensor drift and anomalous terminal behavior—show the most rise. Studies on federated learning for edge computing have suggested that resource and data variety should be taken into account in addition to average accuracy when assessing edge clients [22]. By giving boundary clients, a lower weight when their update direction is similar to that of a dense dependable cluster rather than instantly rejecting them, the strategy can benefit these minority classes. Consequently, compared to general accuracy, this method has a higher macro-F1 score.

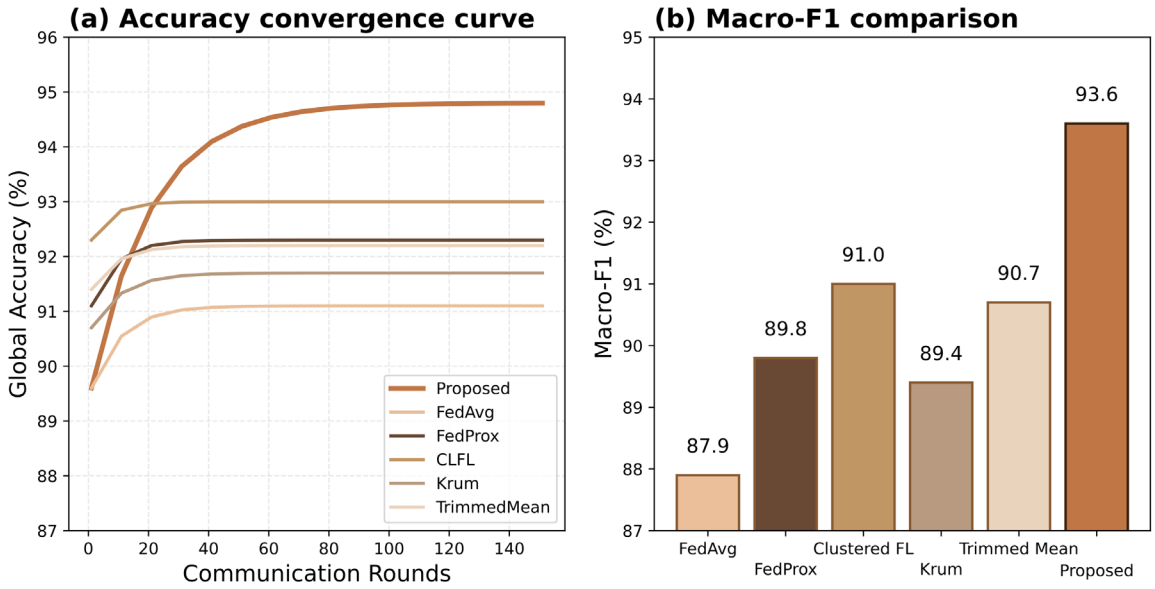

Figure 3. Global convergence and classification performance comparison. (a) Accuracy convergence curve. (b) Macro-F1 comparison.

The client-side stability outcomes are summarized in Figure 4. The distribution of client update distances is shown in Figure 4(a). A total of 12.6% of FedAvg updates fall outside three standard deviations from the round center, indicating a long-tailed update distribution. By applying density clustering in the pre-aggregation sparse-update region, the proposed method lowers this percentage to 4.3%. The aggregation deviation is displayed in Figure 4(b); the suggested approach reduces this deviation by 22.1% compared with clustered federated learning and by 37.4% compared with FedAvg. Research on blockchain-enabled federated learning demonstrates that trust hierarchies and unreliable participants must be managed in secure distributed learning [23]. The density-adaptive technique therefore provides a lightweight trust approximation without requiring explicit identity-based trust scores or blockchain consensus.

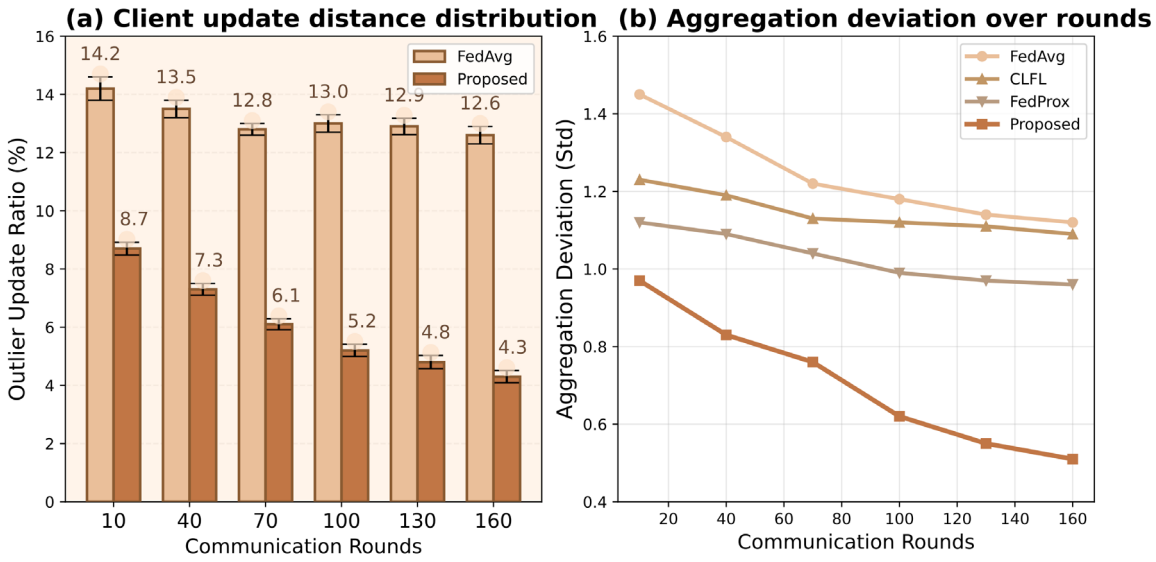

Figure 4. Client update distribution and aggregation stability analysis. (a) Client update distance distribution. (b) Aggregation deviation comparison.

Additionally, Aggregation Trajectories demonstrate how various rounds of aberrant local updates impact the whole training. Some uncommon yet respectable clients are initially located far from the global center due to a lack of local categories. They would be eliminated and minority-class learning would be harmed by a rigid outlier screening. The healthy individuals have gathered into a thick concentration, leaving only the really sick. Research on edge computing and federated learning with blockchain reveals that dynamic involvement and ambiguous trust are typical in edge contexts [24]. The suggested method

To accomplish the effects of early wide tolerance and late acute abnormal-client suppression, it employs adaptive radius estimate in each cycle.

The loss curve's shape will likewise change when the aggregation deviation decreases. After rounds 40, 73, and 106, FedAvg exhibits a few abrupt rises; these increases happen during rounds in which several unreliable terminals engage concurrently. FedProx uses local regularization to decrease some jumps, but it is unable to distinguish between an aberrant end state and an update that comes from a legitimate unusual distribution. Although Krum is less vulnerable to severe updates, it frequently chooses a single central update, which prevents it from properly utilizing dispersed IoT knowledge. The new method's collection area is bigger but still constrained. As a result, it will be more stable than standard averaging and converge more quickly than robust single-update selection.

Additionally, the validation findings' variance has decreased. The suggested strategy decreased the ultimate accuracy standard deviation to 0.51 percent from 1.42 percent for FedAvg and 0.96 percent for FedProx during five repeated runs. A lower variance suggests that the outcome is less susceptible to the fortunate client sample order. Because retraining will be done on a regular basis and each cycle should result in a predictable model rather than one that varies significantly owing to the involvement of few faulty terminals in a few crucial rounds, stability is necessary for the IoT operator.

Client Update Distribution and Aggregation Stability

Following some discussion, the clustering results are displayed in Figure 5. The primary client ratio is shown in Figure 5(a). Due to the volatility of local models, only 48.5% of the chosen customers were classified as core clients by round 20. The percentage of consistent clients has been steadily rising, and by round 100, it is 76.8%. The border client ratio, shown in Figure 5(b), ranges from 14.2% to 22.5%. Because they often contain legitimate but regionally biased IoT patterns, these clients are comparatively good. Local heterogeneity should not be confused with malevolent behavior, according to the Federated Learning Survey [25]. Consequently, rather than eliminating the boundary customers, the suggested approach keeps them at a reduced weight.

The distribution of noisy clients is shown in Figure 5(c). Density-adaptive DBSCAN detects 91.7% of the clients as sparse or isolated updates while 10% of them are anomalous. with typical users, the false rejection rate is 3.6%,

which is less than the 6.8% with Krum-based filtering. Cyber or device-level anomalies might show up as anomalous behavior of local models, according to federated deep learning for IoT security [26]. As a result, in reality, questionable training inputs may be identified using the update density. When the anomalous client has a sudden update displacement, a delayed local loss, or an irregular gradient direction, it performs better.

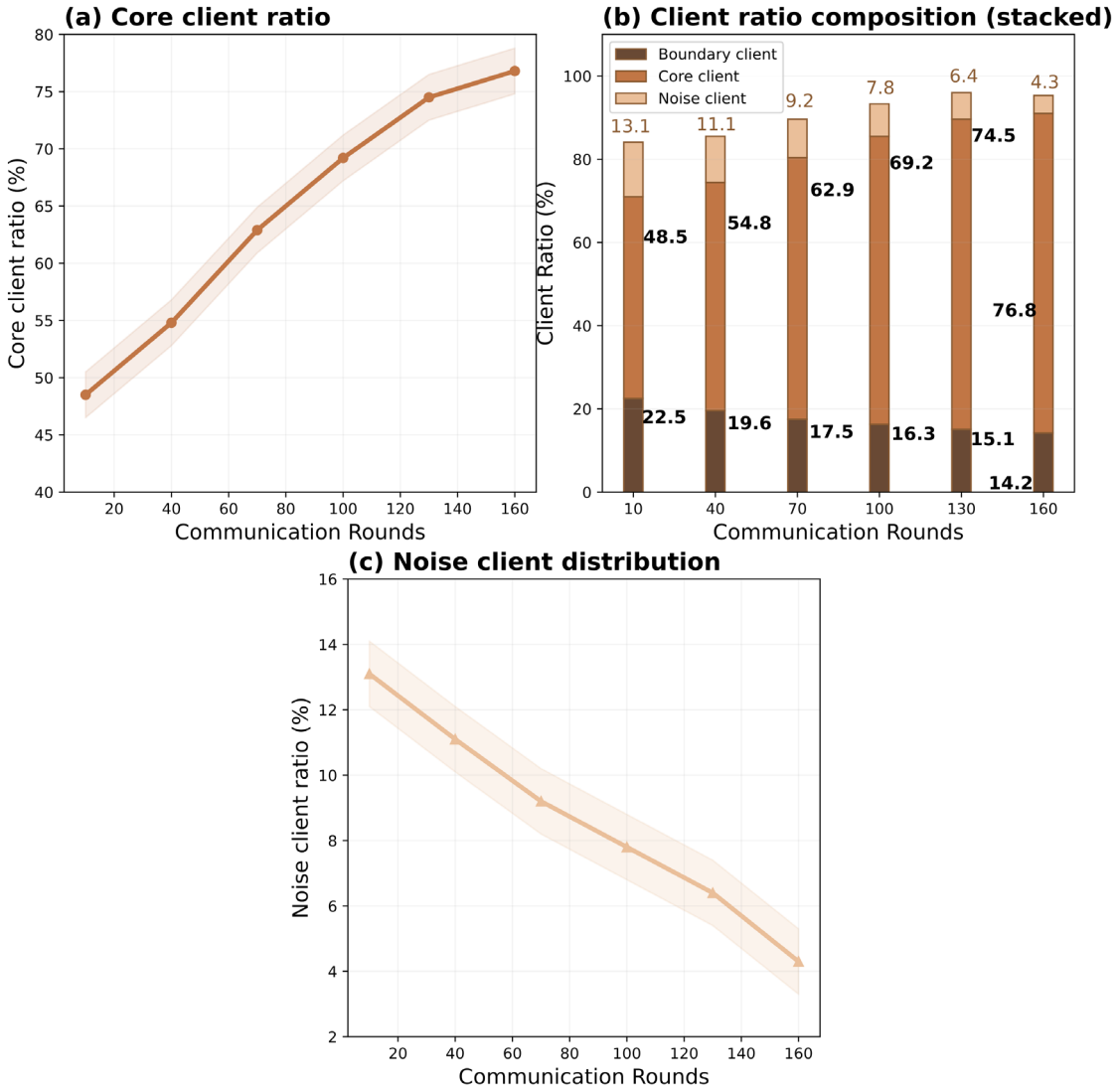

Figure 5. Density clustering behavior across federated rounds. (a) Core client ratio. (b) Boundary client ratio. (c) Noise client distribution.

Descriptor design and adaptive radius selection have an impact on clustering quality. Because just parameter displacement is employed, some customers with uncommon but legitimate categories are mistakenly labeled as noise. Because just gradient cosine similarity is considered, certain anomalous clients that imitate the dominating update show up.

There are no instructions. Better density structure has been demonstrated using the five-factor descriptor, and client update deviations from the group may be explained by communication quality and loss dynamics. According to studies on industrial IoT federated learning, during collaborative training, device behavior, communication quality, and data distribution are all interrelated [27].

Additionally, the client's present state has altered. 18–24 clients are frequently categorized as boundary clients during the first thirty rounds as the local data distributions are still very different. Many of these clients have transitioned to the core cluster by the 80th round, after the global model's acquisition of common low-level representations. The majority of clients equipped with delayed updates, significant local loss variation, or injected label disturbance remain noisy points for more than five consecutive chosen rounds. The time series above demonstrate that the suggested approach is not limited to single-time-point outliers. Battery level,

channel interference, and sensor recalibration may all affect device quality. This round-aware perspective of terminal dependability is more in line with the real management of an IoT network.

By altering the local distribution, density clustering can solve the fairness issue. Although they are concentrated in a limited number of terminals, the classes of anomalous behavior and edge-device overload are uncommon in the global dataset. Because these classes' gradients are less common, a straightforward majority-rule aggregation rule will lessen their impact. A tiny but well-organized border group can be protected in the manner described above. As a result, the recall of edge-device overload has increased from 84.1% to 92.6% and the recall of anomalous terminal behavior has grown from 82.5% to 91.2% in FedAvg. Density-aware grouping is therefore more advantageous for minority terminal states since these improvements outweigh the overall accuracy increase.

Robustness, Ablation, and Deployment Discussion

Figure 6 illustrates robustness in heterogeneous IoT terminal environments. The outcomes of the various non-IID severities are displayed in Figure 6(a). FedAvg's accuracy decreases by 7.9 percentage points when the Dirichlet coefficient is lowered from 1.0 to 0.2, but the suggested technique only decreases by 3.1 percentage points. The malicious client ratio is shown in Figure 6(b). The suggested approach nevertheless produced an accuracy of 90.6% at a 20% abnormal-client ratio, which was better than FedAvg's 83.7% and Krum's 87.2%. The communication dropout scenario is shown in Figure 6(c); even if 30% of the chosen customers are unable to upload, the suggested approach remains stable. The packet-loss disturbance is seen in Figure 6(d). Robust learning is required to sustain detection performance despite unstable terminal involvement, according to federated anomaly detection research in the Internet of Things [28].

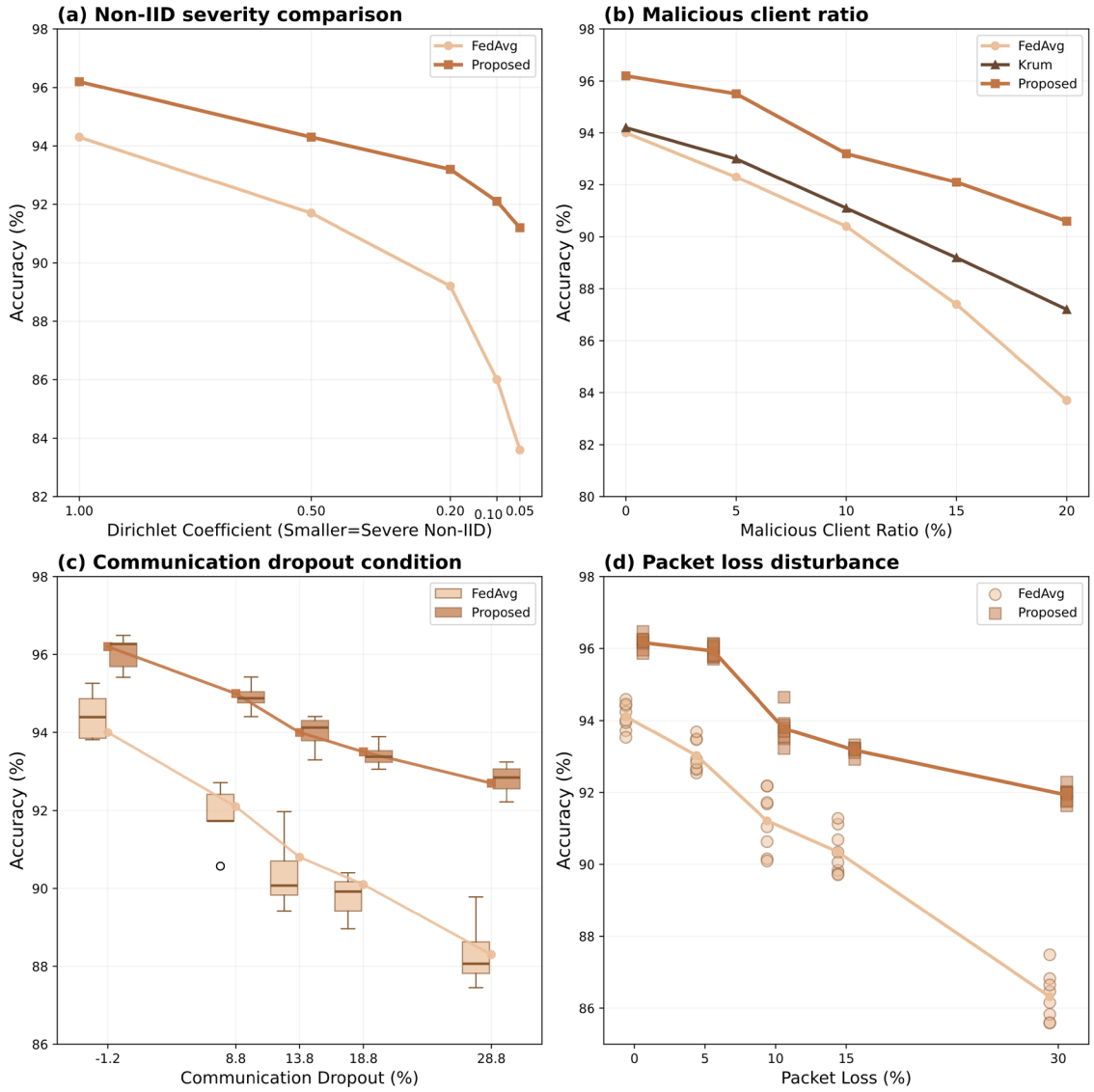

Figure 6. Robustness under heterogeneous IoT terminal conditions. (a) Non-IID severity comparison. (b) Malicious client ratio. (c) Communication dropout condition. (d) Packet loss disturbance.

The following are the causes of the robustness advantage. First, not every terminal is forced into a predetermined number of clusters using adaptive DBSCAN. Second, density confidence lowers the danger of losing valued but uncommon clients by converting client status into continuous reliability. Third, reliability-weighted aggregation modifies the local contribution according to update credibility and data size. According to Privacy and Security Studies in Federated Learning, safe model training depends on participant dependability and aggregate robustness [29]. The design immediately satisfies the aggregate criterion.

The ablation and deployment efficiency findings are displayed in Figure 7. The module removal is shown in Figure 7(a). The accuracy decreases from 94.8% to 91.9% in the absence of adaptive radius estimation, to 90.8% in the absence of noise suppression, and to 92.4% in the absence of reliability weighting. Convergence rounds are compared in Figure 7(b), and the entire model converges 28 rounds quicker than FedAvg. The communication cost is shown in Figure 7(c). The additional upload cost is 2.7% more than in regular FedAvg since the descriptor only has five scalar groups and does not send raw data. The edge inference delay is displayed in Figure 7(d), and the CPU-only edge emulator's local update descriptor creation takes 3.8 ms per terminal. Practical federated systems must preserve local data and most computations close to the end device, according to edge privacy-preserving research [30].

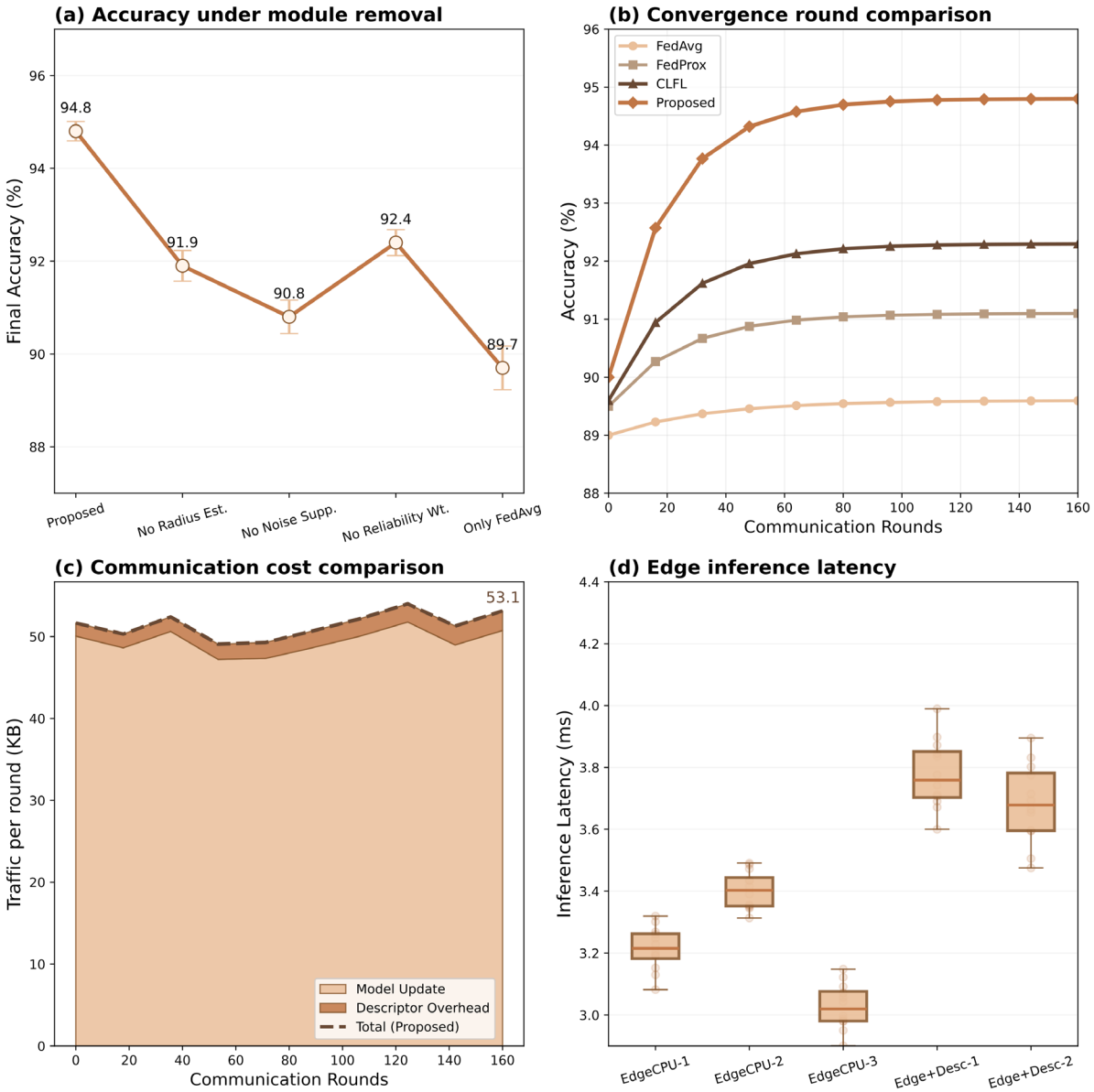

Figure 7. Ablation and edge deployment efficiency analysis. (a) Accuracy under module removal. (b) Convergence round comparison. (c) Communication cost comparison. (d) Edge inference latency.

The two types of the mistakes are displayed below. The first is a mixed-condition kind in which the end-device exhibits both unstable communication quality and an uncommon but valid category distribution. The second is an artificially packed region created by a number of unreliable clients and a coordinated anomalous update group. Stronger local uncertainty estimates can be used to tackle the first scenario, while security-aware trust modeling beyond density clustering is required for the second. The shortcomings do not lessen the usefulness of density-adaptive grouping; rather, they suggest that density structure should be paired with safe aggregation, long-term client reputation, and privacy protection in future IoT federated systems.

The approach is more suited for implementation in edge-assisted IoT federations than in extremely small terminal groups. When fewer than 10 clients participate in a round, the update space is too sparse to create significant neighborhoods, making density estimate incorrect. The adaptive radius stabilizes and clustering consistently displays core-boundary-noise separation after over thirty consumers have joined. As a result, the approach works better for gateways that manage tens or hundreds of terminals, including distributed environmental stations, smart building floors, industrial monitoring regions, and vehicle-roadside groups. Because descriptor upload is insignificant in comparison to model update upload, communication overhead is still minimal.

The first is that the server has a diagnostic view of the terminal and an improved global model. Only updates are gathered and a new global model is produced by a conventional federated server. Along with identifying which clients are crucial border nodes and which clusters are in charge of the global update, the new server will also list the clients that have been frequently disconnected. Adaptive client sampling, end-user trust management, and maintenance planning can all benefit from the diagnostic findings. For instance, the operator may have calibrated the sensor if a terminal consistently shows up as a noisy client during steady communication. The system may raise the frequency of sampling if a boundary client consistently improves the minority class's recall. Therefore, federated learning may be thought of as a kind of training service that solves the blind aggregation issue for distributed IoT networks.

Additionally, the deployment crew can easily comprehend the method. In a factory IoT network, clients linked to the same gateway may have comparable ambient circumstances but different local workloads and sensor ages. In a smart city network, roadside equipment may experience sporadic wireless congestion and fluctuating vehicle volumes. The update time intervals of terminals connected to the HVAC, lighting, and access-control subsystems of a building automation system may vary. These groups do not need to be manually specified using the suggested density-adaptive technique. The aggregate rule should react in accordance with the structure of the update behavior inferred from the current round. As a result, this approach is more adaptable to future changes and easier to install than a manually setup clustering system.

Density adaptation is more advantageous when there is both heterogeneity and unreliability, according to the tests mentioned above. After several rounds, most aggregation techniques will be substantially same if the data are almost independent and the communication is reliable. When non-IID data, client dropout, and irregular updates occur at the same time, the benefits of the suggested approach become more apparent. This scenario, where terminals have varying data sizes, device ages, network circumstances, and sensing contexts, is most likely to arise in an actual dispersed IoT training. The suggested approach improves both convergence stability and minority-class detection by using a density-adaptive DBSCAN component at the server level to identify complexity at the round level before averaging updates.

Conclusion

For dispersed IoT terminals, this research suggests a density-adaptive federated model training approach based on DBSCAN. The technique uses adaptive density clustering to identify core, boundary, and noisy clients after displaying client updates as gradient directions, parameter displacements, local loss changes, density evidence, and communication reliability. The findings suggest that density-aware client grouping can enhance federated training stability in the presence of non-IID IoT data, anomalous local updates, and erratic connectivity.

Thus, it can be said that not every IoT terminal should be given the same weight by strong federated aggregation. The suggested approach outperforms FedAvg, FedProx, clustered federated learning, Krum, and trimmed-mean aggregation in terms of accuracy, macro-F1 score, aggregation deviation, and resilience to dishonest or untrustworthy clients. Because it may modify the neighborhood scale in accordance with the round-level update distribution and does not require a predetermined cluster number, adaptive DBSCAN is also practical.

For the next work, four approaches will be used. First, terminals with lengthy or erratic upload latency can be addressed using asynchronous federated learning. Second, strong formal privacy guarantees may be obtained by combining density-aware grouping with safe aggregation and differential privacy. Third, coordinated anomalous clusters may be identified using the extended client reputation. Fourth, in order to provide reliable federated training on actual IoT networks under stringent energy, memory, and bandwidth limits, lightweight implementation on microcontrollers and edge gateways needs to be further investigated.

Author Contributions

Liviu Kovași contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

References

[1] Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., et al. (2021). Advances and open problems in federated learning. Foundations and Trends in Machine Learning, 14(1-2), 1-210. https://doi.org/10.1561/2200000083

[2] Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine, 37(3), 50-60. https://doi.org/10.1109/MSP.2020.2975749

[3] Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology, 10(2), 1-19. https://doi.org/10.1145/3298981

[4] Lim, W. Y. B., Luong, N. C., Hoang, D. T., Jiao, Y., Liang, Y. C., Yang, Q., et al. (2020). Federated learning in mobile edge networks: A comprehensive survey. IEEE Communications Surveys & Tutorials, 22(3), 2031-2063. https://doi.org/10.1109/COMST.2020.2986024

[5] Aledhari, M., Razzak, R., Parizi, R. M., & Saeed, F. (2020). Federated learning: A survey on enabling technologies, protocols, and applications. IEEE Access, 8, 140699-140725. https://doi.org/10.1109/ACCESS.2020.3013541

[6] Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J., & Poor, H. V. (2021). Federated learning for Internet of Things: A comprehensive survey. IEEE Communications Surveys & Tutorials, 23(3), 1622-1658. https://doi.org/10.1109/COMST.2021.3075439

[7] Mothukuri, V., Parizi, R. M., Pouriyeh, S., Huang, Y., Dehghantanha, A., & Srivastava, G. (2021). A survey on security and privacy of federated learning. Future Generation Computer Systems, 115, 619-640. https://doi.org/10.1016/j.future.2020.10.007

[8] Zhang, C., Xie, Y., Bai, H., Yu, B., Li, W., & Gao, Y. (2021). A survey on federated learning. Knowledge-Based Systems, 216, 106775. https://doi.org/10.1016/j.knosys.2021.106775

[9] Zhu, H., Xu, J., Liu, S., & Jin, Y. (2021). Federated learning on non-IID data: A survey. Neurocomputing, 465, 371-390. https://doi.org/10.1016/j.neucom.2021.07.098

[10] 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

[11] Ezugwu, A. E., Ikotun, A. M., Oyelade, O. O., Abualigah, L., Agushaka, J. O., Eke, C. I., et al. (2022). A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. Engineering Applications of Artificial Intelligence, 110, 104743. https://doi.org/10.1016/j.engappai.2022.104743

[12] Bhattacharjee, P., & Mitra, P. (2020). A survey of density-based clustering algorithms. Frontiers of Computer Science, 15(1), 151308. https://doi.org/10.1007/s11704-019-9059-3

[13] Pillutla, K., Kakade, S. M., & Harchaoui, Z. (2022). Robust aggregation for federated learning. IEEE Transactions on Signal Processing, 70, 1142-1154. https://doi.org/10.1109/TSP.2022.3153135

[14] Sattler, F., Muller, K. R., & Samek, W. (2021). Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. IEEE Transactions on Neural Networks and Learning Systems, 32(8), 3710-3722. https://doi.org/10.1109/TNNLS.2020.3015958

[15] Lin, W., Xu, Y., Liu, B., Li, D., Huang, T., & Shi, F. (2022). Contribution-based federated learning client selection. International Journal of Intelligent Systems, 37(10), 7235-7260. https://doi.org/10.1002/int.22879

[16] Pokhrel, S. R., & Choi, J. (2020). Federated learning with blockchain for autonomous vehicles: Analysis and design challenges. IEEE Transactions on Communications, 68(8), 4734-4746. https://doi.org/10.1109/TCOMM.2020.2990686

[17] Zhang, T., Gao, L., He, C., Zhang, M., Krishnamachari, B., & Avestimehr, A. S. (2022). Federated learning for the Internet of Things: Applications, challenges, and opportunities. IEEE Internet of Things Magazine, 5(1), 24-29. https://doi.org/10.1109/IOTM.004.2100182

[18] Mothukuri, V., Khare, P., Parizi, R. M., Pouriyeh, S., Dehghantanha, A., & Srivastava, G. (2022). Federated-learning-based anomaly detection for IoT security attacks. IEEE Internet of Things Journal, 9(4), 2545-2554. https://doi.org/10.1109/JIOT.2021.3077803

[19] Chen, M., Shlezinger, N., Poor, H. V., Eldar, Y. C., & Cui, S. (2021). Communication-efficient federated learning. Proceedings of the National Academy of Sciences, 118(17), e2024789118. https://doi.org/10.1073/pnas.2024789118

[20] Niknam, S., Dhillon, H. S., & Reed, J. H. (2020). Federated learning for wireless communications: Motivation, opportunities, and challenges. IEEE Communications Magazine, 58(6), 46-51. https://doi.org/10.1109/MCOM.001.1900461

[21] Zhan, Y., Li, P., Qu, Z., Zeng, D., & Guo, S. (2020). A learning-based incentive mechanism for federated learning. IEEE Internet of Things Journal, 7(7), 6360-6368. https://doi.org/10.1109/JIOT.2020.2967772

[22] Wen, J., Zhang, Z., Lan, Y., Cui, Z., Cai, J., & Zhang, W. (2022). A survey on federated learning: Challenges and applications. International Journal of Machine Learning and Cybernetics, 14(2), 513-535. https://doi.org/10.1007/s13042-022-01647-y

[23] Nguyen, D. C., Ding, M., Pham, Q. V., Pathirana, P. N., Le, L. B., Seneviratne, A., et al. (2021). Federated learning meets blockchain in edge computing: Opportunities and challenges. IEEE Internet of Things Journal, 8(16), 12806-12825. https://doi.org/10.1109/JIOT.2021.3072611

[24] Li, D., Han, D., Weng, T. H., Zheng, Z., Li, H., Liu, H., et al. (2022). Blockchain for federated learning toward secure distributed machine learning systems: A systemic survey. Soft Computing, 26(9), 4423-4440. https://doi.org/10.1007/s00500-021-06496-5

[25] Qayyum, A., Janjua, M. U., & Qadir, J. (2022). Making federated learning robust to adversarial attacks by learning data and model association. Computers & Security, 121, 102827. https://doi.org/10.1016/j.cose.2022.102827

[26] Ferrag, M. A., Friha, O., Hamouda, D., Maglaras, L., & Janicke, H. (2021). Federated deep learning for cyber security in the Internet of Things: Concepts, applications, and experimental analysis. IEEE Access, 9, 138509-138542. https://doi.org/10.1109/ACCESS.2021.3118642

[27] Wang, X., Han, Y., Leung, V. C. M., Niyato, D., Yan, X., & Chen, X. (2020). Convergence of edge computing and deep learning: A comprehensive survey. IEEE Communications Surveys & Tutorials, 22(2), 869-904. https://doi.org/10.1109/COMST.2020.2970550

[28] Weinger, B., Kim, J., Sim, A., Nakashima, M., Moustafa, N., & Wu, K. J. (2022). Enhancing IoT anomaly detection performance for federated learning. Digital Communications and Networks, 8(3), 314-323. https://doi.org/10.1016/j.dcan.2022.02.007

[29] Yin, X., Zhu, Y., & Hu, J. (2021). A comprehensive survey of privacy-preserving federated learning. ACM Computing Surveys, 54(6), 1-36. https://doi.org/10.1145/3460427

[30] Lu, Y., Huang, X., Zhang, K., Maharjan, S., & Zhang, Y. (2021). Communication-efficient federated learning for digital twin edge networks in industrial IoT. IEEE Transactions on Industrial Informatics, 17(8), 5709-5718. https://doi.org/10.1109/TII.2020.3010798