Computer Networks 281 (2026) 112205

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Hybrid clustering-guided federated learning for robust intrusion detection in highly heterogeneous IoT environments

Luis Miguel García-Sáez [a,*], Sergio Ruiz-Villafranca [b,1], José Roldán-Gómez [c,1], Javier Carrillo-Mondéjar [c,1], José Luis Martínez [a,1]

[a] University of Castilla-La Mancha, C/ Investigación 2, Albacete, 02071, Spain
[b] Technical University of Madrid, Alan Turing s/n, Madrid, 28031, Spain
[c] University of Zaragoza, C. María de Luna 3, Zaragoza, 50018, Spain

ARTICLE INFO

Keywords:
Federated learning
IoT security
Intrusion detection system
Cyber threat detection
Adaptive clustering

ABSTRACT

The growing complexity and scale of Internet of Things (IoT) ecosystems have intensified the emergence of cyber threats and amplified the impact of data heterogeneity across devices. These environments are characterised by their inherent hostility, comprising resource-limited and intermittently connected devices. Consequently, this poses a considerable challenge to the stability and reliability of conventional Federated Learning (FL) approaches. Standard aggregation schemes such as FedAvg, FedProx, FedAdam, and SCAFFOLD often fail under such extreme non-Independent and Identically Distributed (non-IID) conditions, leading to unstable convergence and biased global models. This work introduces a double-clustering federated architecture for intrusion detection that coordinates training at two levels. Locally, lightweight micro-clustering organises client-side updates into consistent groups, reducing the influence of inconsistent local updates. At the server level, density-based (HDBSCAN) clustering discovers evolving families of distributionally compatible clients, allowing coordination to adapt as heterogeneity evolves over time. Clustering is stabilised across rounds through a stability-aware assignment rule. Training then proceeds via family-wise aggregation, producing one expert model per family and a global fallback model for outliers and unassigned participants. Extensive experiments on three public IoT cybersecurity datasets, X-IIoTID, RT-IoT22, and Edge-IIoTset, demonstrate the robustness of the proposed strategy across both lightweight and Deep Learning (DL) models. The architecture achieves up to 19.9% higher F1-score than standard FL methods and maintains over 90% of its peak performance even under severe non-IID conditions, while keeping runtime efficiency within ±15%. These results establish clustering-guided coordination as a practical and resilient foundation for federated intrusion detection, capable of sustaining high accuracy and stability in the most adversarial IoT environments.

1. Introduction

The proliferation of Internet of Things (IoT) devices has transformed digital ecosystems by embedding computation and communication into everyday environments. This has driven ubiquitous connectivity, automation, and data-driven services in sectors such as healthcare, transport, and smart cities [1]. With billions of devices deployed in highly distributed networks that generate large amounts of data [2], the IoT landscape has become increasingly heterogeneous. IoT devices differ in hardware, communication protocols, and software stacks, often operating in resource-constrained or unmonitored settings [3]. This diversity, combined with widespread exposure to untrusted environments, has expanded the surface of attacks and focused cyber threats that specifically target IoT infrastructures [4]. Common attacks include Denial-of-Service (DoS), eavesdropping, and botnet recruitment [5], with adversaries continuously evolving their strategies to bypass existing defences [6]. Consequently, ensuring resilience against multi-vector attacks is a key challenge for IoT networks. This highlights the importance of having robust and adaptive intrusion detection mechanisms that can protect such large-scale, heterogeneous environments.

Machine Learning (ML) has become a key component for intrusion detection in IoT environments, often outperforming traditional signature-based defences [7]. Traditional ML techniques and Deep Learning (DL) models learn discriminative patterns from network and

* Corresponding author.
E-mail addresses: luism.garcia@uclm.es (L.M. García-Sáez), sergio.rvillafranca@upm.es (S. Ruiz-Villafranca), jroldan@unizar.es (J. Roldán-Gómez), jcarrillo@unizar.es (J. Carrillo-Mondéjar), joseluis.martinez@uclm.es (J.L. Martínez).
[1] Contributing author.

https://doi.org/10.1016/j.comnet.2026.112205
Received 30 October 2025; Received in revised form 18 February 2026; Accepted 7 March 2026
Available online 14 March 2026

device behaviour data. Meanwhile, anomaly detection methods identify deviations from normal behaviour, thereby revealing stealthy or previously unseen attacks. However, most ML approaches rely on centralised data aggregation. This is particularly critical in the IoT domain, where devices continuously monitor physical environments and user activities. Therefore, raw traffic implicitly reveals sensitive behavioural patterns, daily routines, or proprietary operational secrets [8]. The transmission of such data increases the risk of intrusive profiling and conflicts with strict data privacy regulations [9]. They also struggle to cope with data imbalance, distributional drift, and the non-Independent and Identically Distributed (non-IID) nature of IoT data [10–12]. Federated Learning (FL) addresses these challenges by enabling devices to collaboratively train models while sharing only learned parameters [13]. This decentralised paradigm preserves privacy, reduces data leakage risks, and scales efficiently across heterogeneous networks [14]. In the context of IoT security, FL enables adaptive and distributed defence mechanisms. This paradigm has motivated the rise of Federated Intrusion Detection Systems (FL-IDSs) that combine collaborative learning with robust protection against evolving threats [15].

Although FL offers clear benefits for intrusion detection, it does not inherently overcome the severe heterogeneity of IoT environments [16]. As a consequence, data collected across devices exhibit strong biases, leading to inconsistent local updates and unstable global convergence. Conventional aggregation schemes such as FedAvg and FedProx assume that parameter averaging captures global data patterns [17], but this assumption breaks down under non-IID conditions. As a result, aggregated models become biased towards dominant behaviours, thereby reducing their ability to detect diverse or rare attack types [18]. In extreme cases, clients with distinct data profiles may even experience degraded local performance, exposing blind spots in network protection.

This phenomenon illustrates how naive parameter averaging can harm devices rather than assist devices at the edges of the distribution. Overcoming these limitations requires more sophisticated strategies capable of adapting to uneven participation, diverse resources, and adversarial conditions. At the same time, these strategies must enable the transfer of knowledge without overloading clients whose data differs significantly from the global consensus. As a result, FL-IDSs may struggle to provide protection across all nodes. Addressing these shortcomings requires innovative approaches that adapt to hostile conditions and exploit diversity rather than treating it merely as noise.

Building upon these limitations, we propose a FL architecture for intrusion detection that adapts clustering-based FL through an adaptive double-clustering methodology. The proposed architecture builds on existing FL clustering concepts by incorporating stabilisation mechanisms and an IDS-oriented design tailored to highly heterogeneous IoT environments. Instead of enforcing a single global model, our approach dynamically groups clients with similar data distributions into clusters. This allows the training of specialised expert models tailored to the characteristics of each group. This clustering mechanism allows clients to benefit from collaboration with peers whose patterns are more aligned with their own, while still retaining access to a fallback global model for broader generalisation. Importantly, the clustering is not static but adaptive, being continuously refined within the probabilistic space of the evolving models.

By integrating supervised learning models with this adaptive double-clustering strategy, the architecture can effectively exploit client diversity. As a consequence, detection performance improves markedly in heterogeneous IoT environments. The result is a more personalised, resilient, and scalable FL-IDS designed to operate effectively in the highly dynamic context of IoT networks. At the same time, it maintains performance comparable to existing strategies in near-IID scenarios, while preserving computational and temporal efficiency. The proposed scheme introduces minimal overheads, and in many cases none at all, when compared with existing approaches. For that, this proposal ensures that the approach remains practical and scalable.

Finally, this paper's main contributions are the following:

- We introduce a federated training architecture based on an adaptive double-clustering methodology, specifically designed to tackle heterogeneity in IoT environments. This architecture enables the creation of specialised expert models while retaining a global fallback. This design enhances both generalisation and personalisation across intrusion detection tasks.
- We deploy two complementary clustering algorithms at local and global levels to enable adaptive coordination. Each client applies lightweight MiniBatch K-Means to generate compact statistical signatures, while the server uses HDBSCAN to automatically group clients with similar distributions.
- An experimental study is conducted on three representative IoT intrusion detection datasets, X-IIoTID, Edge-IIoTset, and RT-IoT22, covering industrial and real-time scenarios. These datasets provide a comprehensive basis for assessing the performance of the proposed approach.
- We benchmark four supervised learning algorithms (Logistic Regression (LR), Support Vector Machine (SVM), Multilayer Perceptron (MLP), and Convolutional Neural Network (CNN)) within the federated setting. This evaluation demonstrates the robustness of the proposed approach across both lightweight and DL models.
- The proposed scheme is evaluated under diverse conditions of heterogeneity and near-IID scenarios, achieving up to 12% higher F1-score than standard FL methods. It maintains over 90% of its peak performance in strongly non-IID settings, with total runtime efficiency kept within $\pm$15%.

The rest of this paper is organised as Section 2 reviews related work, covering both conventional intrusion detection approaches and recent advances in FL under heterogeneous conditions. Section 3 presents the proposed hybrid architecture, detailing its adaptive double-clustering mechanism and supervised learning integration. Section 4 describes the experimental setup, including datasets, partitioning strategy, and model configurations. Section 5 reports and analyses the results, examining convergence, robustness to heterogeneity, and runtime impact. Finally, Section 6 concludes the paper and outlines promising directions for future research.

2. Related work

This section reviews previous research on intrusion detection in IoT networks and recent advances in FL to handle heterogeneity of data. It outlines some of the limitations of existing approaches while highlighting the key strengths and motivations underlying the proposed architecture.

Early intrusion detection in IoT was largely centralised, relying on ML/DL models trained on-device traffic. Proposals such as Kitsune [19], an online ensemble of autoencoders (AE) for router-class hardware, achieved efficient real-time detection but remained confined to a single observation point without cross-device collaboration. Similarly, N-BaIoT [20] introduced deep AE and a widely used dataset for botnet detection, achieving very high accuracy with few false positives. However, it still operated within a centralised paradigm. To overcome privacy and scalability constraints, FL has been adopted in intrusion detection, although often with simple global aggregation.

Lazzarini et al. [21] compared FedAvg, FedAvgM, and adaptive optimisers using a shallow Artificial Neural Network (ANN) on ToN_IoT and CICIDS2017. They found FedAvgM to perform best on both datasets (e.g. F1-score $\approx$ 0.968 on ToN_IoT and accuracy $\approx$ 0.981 on CICIDS2017), while adaptive methods lagged. Nevertheless, no explicit mechanisms for heterogeneity were considered. Campos et al. [22] studied FL-IDSs under non-IID partitions of CIC-ToN-IoT using multinomial LR, comparing FedAvg with Fed+. They found that Fed+ alleviated convergence issues and achieved higher accuracy ($\approx$ 0.888 in mixed and $\approx$ 0.904

in balanced settings, versus ≈ 0.842 and ≈ 0.870 for FedAvg). Still, no personalisation or clustering mechanisms were considered. More recently, Khraisat et al.'s PEIoT-DS on N-BaIoT [23] employed deep AE and demonstrated that FedAvgM outperformed FedAvg. It achieved this by improving accuracy (from 0.94 to 0.95), reducing false positives and converging in fewer rounds, while maintaining moderate communication costs.

While these FL-IDS approaches show the feasibility of collaborative training, they often assume that a single global model is sufficient and provide only limited means of dealing with heterogeneity. This motivates the exploration of more sophisticated, heterogeneity-aware strategies. Several heterogeneity-aware FL methods have been proposed to mitigate the limitations of simple parameter averaging. In the clustered FL literature, early foundational frameworks established the baseline for handling non-IID data through clustering. Ghosh et al. [24] proposed the Iterative Federated Clustering Algorithm (IFCA), which allows clients to self-identify their cluster by minimising local loss functions. Validated on EMNIST, IFCA demonstrated that alternating between cluster estimation and model optimisation could reduce error rates significantly compared to global baselines in light non-IID settings. From a geometric perspective, Sattler et al. [25] proposed Clustered Federated Learning (CFL). CFL employed a recursive bi-partitioning strategy based on the cosine similarity of client gradients. This method achieved high accuracy on rotated MNIST and CIFAR-10 tasks where standard FedAvg collapsed, proving particularly effective for separating non-convex objectives. Building on this line of research, more recent approaches have explored clustering-driven coordination mechanisms at scale.

Orchestra [26] introduced globally consistent client-server clustering for unsupervised FL and demonstrated robustness to non-IID settings on CIFAR-10 and CIFAR-100. This proposal reached about 0.716 linear-probe accuracy on CIFAR-10 with 100 clients and $\alpha = 0.1$, and about 0.404 on CIFAR-100, while incurring only $\leq 0.009\%$ client overhead. Bertoli et al. [27] proposed a stacked-unsupervised FL architecture for intrusion detection, combining a deep AE with an Energy-Flow Classifier (EFC). Evaluated on Bot-IoT, ToN_IoT, UNSW-NB15, and CSE-CIC-IDS-2018, the approach yielded F1-scores from about 0.46 - 0.51 with plain FedAvg to 0.78 - 0.84 across datasets by round 10. However, it still relied on a single global detector and did not explicitly personalise models for clients with divergent behaviours. Complementing these, Wei et al. proposed a personalised Bayesian FL method with Wasserstein Barycentre Aggregation (FedWBA) [28]. Tested on FMNIST, CIFAR-10, and CIFAR-100, it achieved ≈ 0.72 - 0.77 accuracy on CIFAR-10 (up to 0.769 with five Bayesian layers) and consistently lower Expected Calibration Error (ECE) (e.g. ECE ≈ 0.013 on CIFAR-10) than existing personalised baselines.

Recent studies have explored heterogeneity within IoT-specific contexts, proposing tailored solutions for these environments. Qiu et al. [29] proposed a hierarchical FL scheme with asynchronous aggregation and lightweight encryption, evaluated on Human Activity Recognition (HAR) dataset. Their method reduced the round time by about 20% compared to HierFAVG while maintaining global accuracy at ≈ 0.942, close to the 0.946 of the baseline. Zhang et al. [30] integrated clustered FL with Deep Q-Network (DQN)-based device selection on MNIST and CIFAR-10, reaching final accuracies of 0.84 and 0.44 respectively under highly non-IID splits. In addition to these results, they reported communication and latency improvements, with rounds reduced by 26–31% and latency by roughly 34% compared to random-selection baselines. Zhou et al. [31] introduced Fed-Knowledge Distillation (KF), a global-local knowledge fusion method evaluated on EMNIST, CIFAR-10, and CIFAR-100 under Dirichlet [32] non-IID partitions with $\alpha \in 1, 0.1, 0.01$. The approach consistently improved average accuracy and fairness over FedAvg. At $\alpha = 0.1$, FedKF achieved 0.85 accuracy on EMNIST, 0.56 on CIFAR-10, and 0.41 on CIFAR-100, outperforming FedAvg in all cases.

Table 1 shows the main features of the approaches reviewed. The table is organised by research focus, following established FL taxonomies [33], to separate (i) clustered and heterogeneity-aware FL methods evaluated on standard benchmarks; (ii) personalised FL; (iii) hierarchical IoT FL applications; and (iv) IoT FL-based intrusion detection studies. Most existing FL-IDS and heterogeneity-aware methods rely on a single global model and evaluate their performance on general-purpose datasets such as CIFAR, MNIST, or HAR. These datasets are designed primarily for image or activity recognition tasks. They do not capture the specific traffic patterns and attack behaviours found in IoT networks. In addition, the number of studies employing recent and dedicated IoT intrusion datasets remains notably limited. Furthermore, several clustering-based FL methods proposed in the literature are primarily evaluated on perceptual benchmarks. These approaches often rely on computationally intensive mechanisms or static grouping assumptions. This limits their direct applicability to lightweight and dynamic intrusion-detection scenarios in IoT environments. In contrast, recent IoT datasets are larger, more diverse, and better reflect the complexity of real network traffic. Moreover, most studies report only accuracy metrics, ignoring complementary metrics such as F1-score which is critical under the strong class imbalance typical of intrusion detection. Runtime is also rarely discussed in any detail beyond isolated latency or communication figures, making it difficult to make practical comparisons.

Our proposal directly addresses these gaps. We employ Dirichlet-based non-IID partitions, where the Dirichlet concentration parameter α controls the degree of heterogeneity among clients. Smaller values represent stronger data heterogeneity across clients. The evaluation is conducted on recent IoT intrusion datasets, reporting both accuracy and F1-score to capture balanced performance under class imbalance. The proposed adaptive double-clustering architecture learns local data signatures and dynamically forms client families for training specialised expert models. It simultaneously maintains a universal backup and a stability-aware assignment to ensure consistent collaboration across heterogeneous clients. This design promotes positive transfer and distributional outlier protection, achieving a high accuracy/F1-score balance across all scenarios.

3. Proposed method

This section introduces the proposed hybrid federated architecture that integrates unsupervised double-clustering with supervised model training. It details the system design, key components, and communication workflow, explaining how the adaptive strategy enhances model alignment and resilience to client heterogeneity.

3.1. Architecture design

Based on the limitations identified in Section 2, we have designed a FL architecture that explicitly handles data heterogeneity and supports personalised intrusion detection in IoT environments. The architecture design of the proposed system and the interaction between its main components are illustrated in Fig. 1. The architecture comprises a central coordinator (server) and a set of federated clients deployed at the boundaries of the IoT network. Each client acts as a gateway that receives and processes data generated by nearby IoT devices or subnetworks, performing local training and feature extraction close to the data source. Training proceeds in communication rounds. In each round, the server distributes model parameters, and clients update them locally before returning (i) model deltas and (ii) a compact data signature, a statistical summary of their local data. With this process, raw data never leave the federated nodes, ensuring privacy preservation.

At the server, these signatures are aggregated and a global density-based clustering algorithm discovers evolving client families. Each family maintains a specialised expert model trained via supervised learning on participating clients of that family. A universal backup model is maintained in parallel to provide coverage when clients are uncertainly assigned or behave as outliers. To limit oscillations and negative transfer, a stability-aware assignment policy smooths family updates across

Table 1
Comparison of FL approaches for intrusion detection and heterogeneity-aware methods in IoT environments, organised by research focus.

Reference	Task/Domain	Model type	Aggregation scheme	Adaptivity & personalisation	Datasets	Partitioning & heterogeneity	Best performance	Performance vs baseline	Runtime impact
	Task / Model / Aggregation			*Adaptivity / Datasets / Partitioning*			*Results*		
I. Clustered & heterogeneity-aware FL (standard benchmarks / general settings)									
[26] (2022)	Vision	Unsupervised (SSL + clustering)	FedAvg + client/server clustering	Global only (globally consistent)	CIFAR-10, CIFAR-100	Non-IID (Dirichlet α: 10^5, 0.1, 10^{-3}; 10–400 clients)	C10 0.72; C100 0.40 acc. ($\alpha = 0.1$)	+2–5% vs. baselines	≈ 0% (clustering ≤ 0.009%)
[30] (2024)	IoT	Supervised (CNN)	FedAvg (clustered) + DQN-based device selection	Per-cluster (dynamic DQN); no personalisation	MNIST, CIFAR-10	Non-IID (σ: 0.5–1.0; 100 clients)	MNIST 0.84; CIFAR-10 0.44 acc.	+5–17% vs. random-selection	-34% latency vs. FL-Random
[31] (2024)	IoT	Supervised (CNN + KD)	FedKF	Global only	EMNIST, CIFAR-10, CIFAR-100	Non-IID (Dirichlet α: 0.01–1; 20 clients)	EMNIST 0.85; C10 0.56; C100 0.41 acc. ($\alpha = 0.1$)	+1–3% vs. FedAvg	Not reported
II. Personalised FL									
[28] (2025)	Vision	Supervised (Bayesian FL)	Wasserstein barycenter (Bayesian)	Personalised (FedWBA)	MNIST, FMNIST, CIFAR-10, CIFAR-100	Non-IID (50–200 clients)	MNIST 0.98; FMNIST 0.93; C10 0.73; C100 0.64 acc.	+0–1% vs. best baseline	Not reported
III. Hierarchical / system-level FL in IoT (non-IDS applications)									
[29] (2025)	IoT (HAR)	Supervised (not defined)	Hierarchical (edge–cloud), asynchronous	Global only; async hierarchical	HAR	Non-IID (4–20 clients)	HAR 0.94 acc.	−0.4% vs. HierFAVG	-20% round time vs. HierFAVG
IV. Federated intrusion detection systems (IoT-IDS / network IDS)									
[22] (2022)	IoT IDS	Supervised (multinomial LR)	FedAvg, Fed+	Global only (entropy-based)	CIC-ToN_IoT	IID & non-IID (10 clients)	0.87–0.90 acc. (CIC-ToN_IoT)	+0.01–5.3% vs. FedAvg	Not reported
[21] (2023)	IoT IDS	Supervised (ANN)	FedAvg, FedAvgM, FedAdam, FedAdagrad	Global only	ToN_IoT, CICIDS2017	IID (4 clients)	ToN_IoT 0.97–0.98; CICIDS2017 0.98 acc.	−0.5–0.8% vs. central	Not reported
[27] (2023)	Network IDS (incl. IoT)	Unsupervised (stacked AE + EFC)	FedAvg (standard)	Global only	Bot-IoT, ToN_IoT, UNSW-NB15, CSE-CIC-IDS-2018	Non-IID cross-silo (4 clients)	0.78–0.84 F1-score (across datasets)	+27–38% vs. FedAvg (plain)	Not reported
[23] (2025)	IoT IDS	Supervised (deep AE)	FedAvg, FedAvgM	Global only	N-BaIoT	Non-IID (9 clients)	N-BaIoT 0.95 acc.	+1.0% vs. FedAvg	Not reported
Our proposal (2025)	IoT IDS	Hybrid (unsupervised double layer clustering + supervised)	FedAvg, FedProx, FedAdam, SCAFFOLD, **Our hybrid scheme**	Per-cluster experts (dynamic) + global backup; stability-aware assignment	X-IIoTID, RT-IoT22, Edge-IIoTset	Non-IID (Dirichlet α: 0.1–50; 10–200 clients)	X-IIoTID 0.958/0.957; RT-IoT22 0.989/0.985; Edge-IIoTset 0.912/0.912 (acc/F1; $\alpha = 0.1$)	+0.04–0.55 F1-score improvement vs. baselines	−15.3–12.7% total time vs. baselines

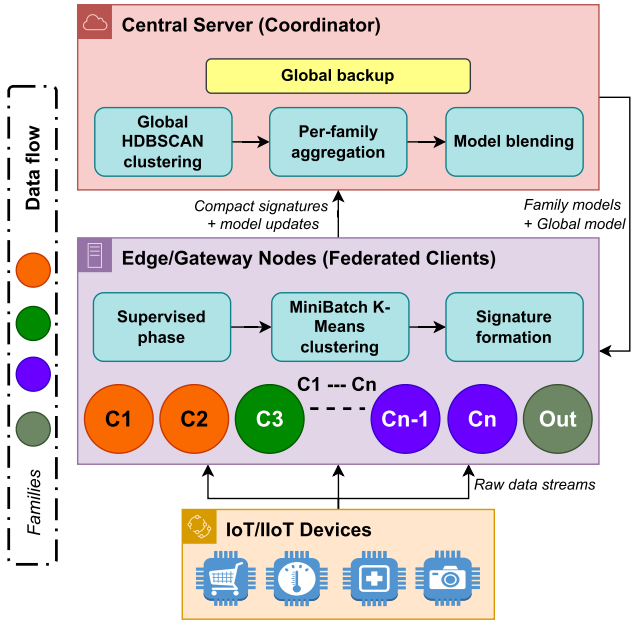

Fig. 1. Workflow of the proposed double-clustering architecture.

rounds. Model aggregation is performed per family, weighted by participation and local sample size, while the backup model is updated using all client contributions. New or ambiguous clients initially rely on the backup model and are migrated once sufficient evidence supports their inclusion in a family. The exchanged signatures are lightweight and privacy-preserving, and the architecture supports partial participation. This design personalises detection for distributionally aligned groups and protects atypical clients in non-IID IoT environments.

3.2. Adaptive double-clustering strategy

IoT clients exhibit a persistent non-IID structure and drift. Rather than forcing a single global model, we first identify the underlying structure and then aggregate data within homogeneous groups. We achieve this by using local micro-clustering to form lightweight signatures, followed by global density-based clustering that yields families and explicit noise. Aggregation is then scoped per family, with a backup model for atypical clients.

Local micro-clustering and client signatures Each client i compresses its local data distribution into a compact signature. In the first round (t=1), clustering is applied to the normalised raw features, since no trained model is yet available. From $t\geq 2$, the process shifts to a predictive space derived from the current local model. Class-probability vectors for probabilistic models or calibrated decision scores for margin-based ones, scaled to ensure comparability across classes. Clustering in this output space groups samples on which the model exhibits similar behaviour rather than purely feature-based similarity. This makes clusters more stable under heterogeneous data and feature-scale differences. As local models evolve, their micro-clusters evolve as well, capturing behaviourally homogeneous regions that better reflect the client's local data dynamics.

MiniBatch K-Means (MBK) [34] is selected for its balance of efficiency, stability, and interpretability in resource-constrained devices. It processes data incrementally with limited memory, supports warm starts between rounds. It also produces per-cluster statistics, which are centroids and diagonal variances, that are compact to transmit and easy to compare on the server. Compared with alternative clustering methods, MBK offers a trade-off between efficiency and robustness. It avoids the costly covariance updates of Expectation-Maximisation Gaussian Mixture Models (EM/GMM) [35], the pairwise distance computations of k-medoids [36], and the parameter sensitivity of density-based methods such as DBSCAN [37]. Unlike the classical K-Means [38] or its K-Means++ [39] variant, it updates centroids incrementally on small batches, greatly reducing memory and latency without loss of accuracy. Regarding computational complexity, MBK operates with linear cost $O(b \cdot k \cdot d)$, whereas alternatives like EM/GMM and DBSCAN have cubic ($O(Nd^3)$) and quadratic ($O(N^2)$) costs, respectively [40]. These properties make MBK a practical choice for on-device micro-clustering, combining simplicity, stability, and scalability.

Adaptive k and signature construction The number of local micro-clusters $m_i^{(t)}$ is adaptive but capped by M, the maximum number of clusters allowed per client, to keep memory and communication bounded. For each candidate k, MBK computes the mean squared distance within the cluster, denoted $\bar{J}_k^{(t)}$, also referred to as inertia. It represents the average squared Euclidean distance between samples and their assigned centroid. We increase k until the relative improvement in compactness becomes negligible, according to Eq. (1):

$$\Delta_k^{(t)} = \frac{\bar{J}_{k-1}^{(t)} - \bar{J}_k^{(t)}}{\bar{J}_{k-1}^{(t)}} < \eta \quad \Rightarrow \quad m_i^{(t)} = k, \tag{1}$$

where η is a small threshold that controls the minimum required improvement. If no smaller k satisfies this condition, we set $m_i^{(t)}=M$. This rule avoids the need for additional validation metrics and is naturally adaptable to local data complexity. From round $t\geq 3$, MBK is warm-started at the previous value $m_i^{(t-1)}$, allowing clusters to evolve with the data while preventing over-segmentation. Very small clusters can be discarded when their relative weight π_{ij} falls below a minimum threshold $\pi_{\min}$. From the final clustering, each client builds a compact signature, as defined in Eq. (2):

$$S_i = \{(\pi_{ij}, \mu_{ij}, v_{ij})\}_{j=1}^{m_i}, \tag{2}$$

where π_{ij} is the cluster weight, μ_{ij} the centroid, and $v_{ij} \in \mathbb{R}^d_{\geq 0}$ a diagonal variance vector regularised with a small floor to stabilise tiny clusters. Only this signature and the local model update are sent to the server, never the raw data or labels.

Server-side feature engineering for clustering Once the client updates are received, the server transforms each local micro-cluster into a rich feature vector. Each centroid μ_{ij} is first normalised to unit length so that all vectors share a comparable scale, independent of the client or feature magnitudes. Then two scalar descriptors of the corresponding variance vector v_{ij}, which are standard deviation, and maximum, are appended. This compact representation preserves both positional and dispersion information. It also stabilises the subsequent global clustering, making the process less sensitive to heterogeneous feature ranges across clients. After this transformation, the server holds one rich vector per client micro-cluster, ready for global grouping.

Global discovery of families and noise The server then applies HDBSCAN [41] to the set of rich vectors produced by all clients. We selected HDBSCAN over alternative density-based methods like DBSCAN or OPTICS primarily due to its ability to handle varying density clusters [42]. Under severe non-IID conditions, client updates produce clusters with drastically different densities driven by the imbalance in local sample sizes and class distributions. Standard DBSCAN, which relies on a single global density threshold (ϵ), typically fails to simultaneously capture both dense and sparse client families in such conditions. Furthermore, unlike partition-based methods (e.g. K-Means) that force every point into a cluster, HDBSCAN explicitly labels ambiguous or outlier points as noise [43]. This property allows the system to route outlier clients to the global fallback model rather than corrupting the specialised family experts. When the feature space is too sparse to form stable clusters, the server retries with a smaller min_cluster_size. If discovery still fails, a coarse K-Means++ over client-level signatures is used as a fallback to maintain progress. The result is a set of families of clients with similar distributions, plus a noise group labelled -1 representing outliers.

Stable family assignment Each client inherits a family label through a majority vote over the labels of its own micro-clusters, excluding those

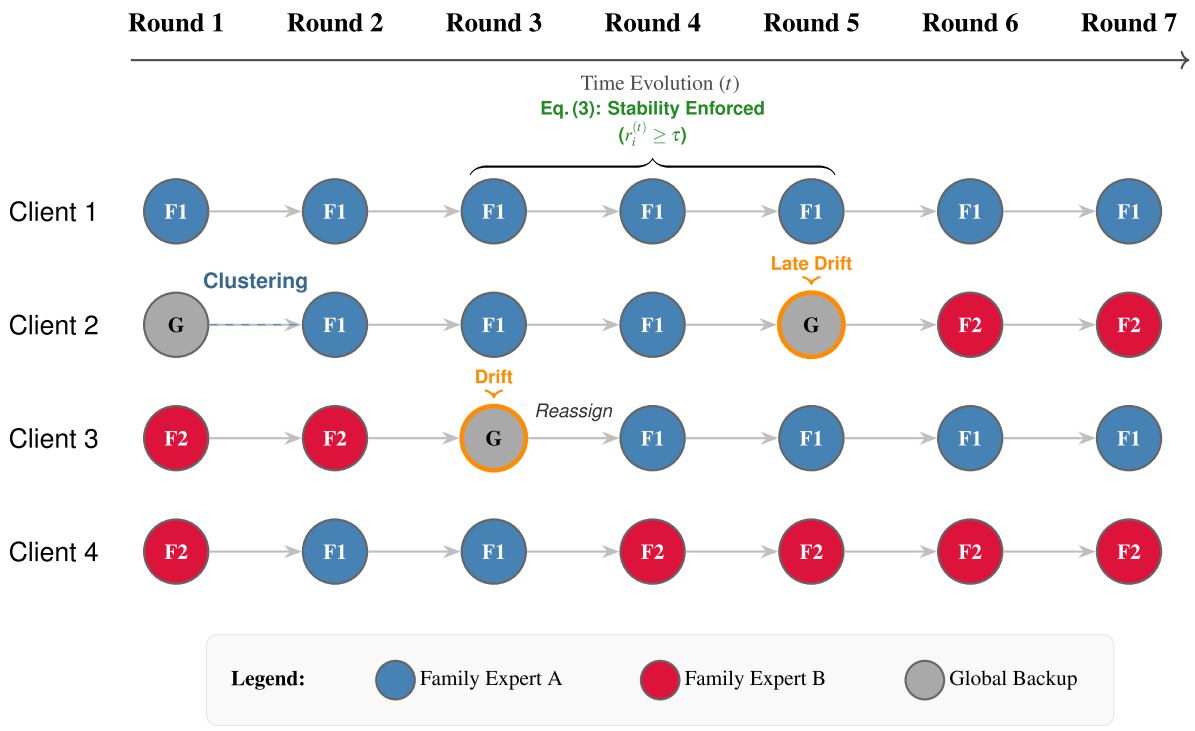

Fig. 2. Family dynamics over 7 rounds: (i) clustering-based assignment to family experts, (ii) stability enforced by Eq. (3) to prevent oscillations, and (iii) drift-driven reassignment, including cross-family migration.

classified as noise. To prevent unstable oscillations between families, the server applies a stability rule. If a client already belonged to a family in the previous round and a sufficient fraction of its current micro-clusters still support that family, the assignment is retained. Otherwise, the new majority label is adopted. Formally, let $r_i^{(t)}$ denote the proportion of micro-clusters of client i that remain associated with its previous family. The client preserves its label whenever the stability condition in Eq. (3) is satisfied:

$$r_i^{(t)} \geq \tau, \tag{3}$$

where τ is a configurable stability threshold that controls the inertia of family assignments. Larger values of τ enforce stronger consistency across rounds, while smaller ones allow faster adaptation to distributional drift. Clients whose micro-clusters are entirely labelled as noise are placed in the noise group. This hysteresis mechanism smooths short-term fluctuations, protecting against transient cluster splits and small local drifts that could cause unnecessary reassignments.

Family-wise aggregation and universal fallback Supervised training is performed locally on each client. After each round, the server aggregates the resulting model updates for each discovered family, using weights proportional to the number of local examples. This produces a specialised expert for each family. In parallel, the server aggregates the updates from all clients to refresh a universal fallback model, which serves as a global reference for unassigned or atypical devices. To smooth out temporal fluctuations between rounds, each family expert is blended with its previous version using a decaying factor, as defined in Eq. (4):

$$\lambda_t = \max\left(0.1,\ 0.3 \times 0.95^{t-1}\right), \tag{4}$$

which progressively reduces the influence of older parameters while maintaining continuity for small or slowly evolving families. In the next round, each client receives the parameters of their assigned family expert. If their assignment is uncertain or labelled as noise, they receive the parameters of the fallback model instead. This routing strategy prevents cross-family contamination and guarantees that every client obtains a stable and valid model, even when cluster membership fluctuates.

Fig. 2 shows client membership evolution over seven communication rounds, contrasting decisions between family experts and the global backup model. It shows that clients with consistent behaviour remain in the same family throughout the rounds (e.g. C1), in line with the stability threshold (Eq. (3)). The figure also illustrates how the method adapts when a client no longer aligns with its current family. When the local predictions drift away from the family pattern, the assignment is re-evaluated and updated to a more suitable family (e.g. C4 at rounds 2 and 4). Furthermore, clients may migrate to the global model prior to this transition (e.g. C3 at round 3).

Finally, Algorithm 1 summarises the complete training cycle, showing how the proposed components interact across rounds to achieve adaptive and stable FL.

4. Experimental setup

This section describes the experimental setup used to evaluate the proposed architecture. It details the datasets, preprocessing steps, and heterogeneity modelling procedures, as well as the learning models and baseline configurations employed for comparison. The section also specifies the evaluation metrics and parameter settings within the FL environment.

4.1. Datasets and data preprocessing

Public datasets are essential for benchmarking IDSs in IoT. They provide standardised traffic traces and labelled attack scenarios for reproducible evaluation. These datasets capture network or device-level activity under both normal and malicious conditions, covering diverse protocols and attack types. We selected three representative public datasets to evaluate the proposed architecture, X-IIoTID, RT-IoT22, and Edge-IIoTset, as they are relatively recent and well-documented. Moreover, they encompass a wide variety of IoT and IIoT network conditions, protocols, and attack scenarios, providing a balanced testbed to analyse the behaviour of the proposed method under heterogeneous and

Algorithm 1: FL cycle with adaptive double-clustering and family-wise aggregation.

Input: Client datasets $\{D_i\}_{i=1}^{N}$, maximum local clusters M, stability threshold τ, inertia threshold η
Output: Family experts $\{\theta_f\}$ and global fallback model θ_g

for *each round* $t = 1, 2, \ldots, T$ **do**
  `// --- Client-side phase ---`
  for *each client* i *in parallel* **do**
    Train local model $\theta_i^{(t)}$ for E epochs on D_i
    Extract embedding space $z_i^{(t)}(x)$ (raw features if $t = 1$, predictive space otherwise)
    Run MiniBatch K-Means with adaptive $m_i^{(t)} \leq M$ using inertia criterion η
    Compute signature $S_i = \{(\pi_{ij}, \mu_{ij}, v_{ij})\}_{j=1}^{m_i^{(t)}}$
    Send $(\theta_i^{(t)}, S_i)$ to server
  `// --- Server-side phase ---`
  Transform each (μ_{ij}, v_{ij}) into a rich vector r_{ij} (normalised centroid + variance stats)
  Apply HDBSCAN to $\{r_{ij}\}$ to discover client families $\{\mathcal{F}_f^{(t)}\}$ and noise label -1
  Assign each client to family $f_i^{(t)}$ by majority vote over its micro-clusters
  Retain previous family if stability ratio $r_i^{(t)} \geq \tau$ else update label
  `// --- Aggregation phase ---`
  for *each family* $\mathcal{F}_f^{(t)}$ **do**
    Aggregate models: $\theta_f^{(t+1)} \leftarrow \sum_{i \in \mathcal{F}_f^{(t)}} w_i \theta_i^{(t)}$
    Blend with previous expert using decay λ_t
  Aggregate all clients into fallback model $\theta_g^{(t+1)}$
  `// --- Dispatch ---`
  Send $\theta_{f_i^{(t+1)}}$ to each client i (or θ_g if $f_i^{(t+1)} = -1$)

non-IID settings. The main characteristics of these datasets are summarised in Table 2, as well as their class distribution.

X-IIoTID This dataset captures IIoT traffic across multiple communication layers and technologies, covering both control and application domains. It offers a rich variety of statistical flow features, protocol attributes, and metadata describing normal and attack behaviours. Its technology-agnostic and device-independent design enables evaluating model generalisability across diverse industrial setups and unseen traffic patterns. X-IIoTID contains ten traffic classes, including benign operations and nine types of cyberattacks that target availability, integrity, and reconnaissance processes [44].

RT-IoT22 This dataset records network traffic generated in a real-time IoT testbed integrating sensors, actuators, and embedded controllers under both benign and malicious conditions. It includes twelve classes corresponding to normal activity and various attacks, such as DoS, spoofing, scanning, and probing. RT-IoT22 stands out for its ability to reproduce time-sensitive and event-driven communication, where timing constraints directly impact network behaviour. Consequently, it provides a valuable benchmark for evaluating the adaptability and robustness of intrusion detection models operating in dynamic IoT environments [45].

Edge-IIoTset Edge-IIoTset combines realistic IoT and IIoT network traces obtained from heterogeneous devices and multi-protocol communication scenarios, including MQTT, Modbus/TCP, and HTTP. It provides a unified data format compatible with both centralised and FL experiments, enriched with metadata describing device roles, protocol types, and traffic context. The dataset comprises fifteen traffic classes, covering benign flows and a broad range of network and application layer attacks. It is particularly suitable for analysing distributional heterogeneity and studying client clustering dynamics in FL-IDSs [46].

Data preprocessing A dedicated preprocessing pipeline was applied and individually adapted to each dataset to account for their specific formats and feature definitions while ensuring overall compatibility. All datasets were cleaned to remove redundant or non-informative columns, such as duplicated identifiers or repeated timestamps, as well as entries with missing labels. Categorical fields were binarised, and invalid or infinite values were corrected by median imputation or replaced with the nearest finite value. Non-numeric or textual attributes (e.g. IP addresses, protocol names) were discarded to yield purely numerical feature spaces. All numerical features were scaled to the range $[0, 1]$ using min-max normalisation to stabilise the clustering and learning procedures. Label fields were unified under a multiclass taxonomy where benign samples were grouped into a single "Benign" class and attack types were standardised across datasets.

4.2. Data partitioning and heterogeneity modelling

To emulate realistic non-IID conditions among clients, the datasets were partitioned following a probabilistic allocation scheme based on the Dirichlet distribution [47]. This approach is commonly used in FL to generate controllable levels of heterogeneity while preserving class coverage across clients [48]. Given K clients and C classes, the per-class proportions for each client are drawn from a Dirichlet distribution $\mathrm{Dir}(\alpha)$, where the concentration parameter $\alpha > 0$ controls how unevenly data are distributed. High values of α yield nearly IID partitions with balanced class proportions, while small values produce strongly skewed, non-IID partitions dominated by few classes.

Formally, for each class c, we sample proportions as defined in Eq. (5):

$$(p_{1c}, p_{2c}, \ldots, p_{Kc}) \sim \mathrm{Dirichlet}(\alpha, \ldots, \alpha), \tag{5}$$

and assign to each client k a fraction p_{kc} of the available samples from class c. This procedure does not guarantee that every client receives all classes. It ensures that, collectively, the federation retains complete class coverage while providing flexible control over the degree of distributional imbalance. To analyse the sensitivity of the proposed architecture to client heterogeneity, we explored a range of α values: $\{50, 20, 10, 5, 1, 0.6, 0.3, 0.1\}$.

A minimum allocation threshold was enforced so that each client contained at least 100 samples, avoiding underpopulated participants. Within each client, data were randomly split into training and testing subsets with an 80/20 ratio, maintaining class distribution in both sets. This guarantees that each client contributes meaningfully to both the clustering and supervised learning stages. The partitioning strategy covers a wide range of heterogeneity scenarios, allowing a thorough evaluation of the architecture's capability to adapt to data imbalance and distributional shift among clients.

Fig. 3 illustrates examples of the resulting data partitions for the X-IIoTID and RT-IoT22 datasets under different values of α. Each heatmap shows the proportion of samples per class assigned to each client. As α decreases, class concentration becomes increasingly uneven, leading to more heterogeneous and non-overlapping distributions. For moderate heterogeneity (α=1.0), most clients still preserve coverage of all classes, although proportions vary noticeably. Under highly skewed conditions (α=0.1), more than half of the clients lack at least one class entirely, highlighting strong distributional imbalance. This visual evidence confirms how the Dirichlet-based allocation effectively controls the degree of non-IID behaviour across the federation.

4.3. Baselines and learning models

In order to benchmark the proposed architecture, we adopted four representative aggregation strategies widely used in FL: FedAvg, FedProx, FedAdam, and SCAFFOLD. These methods provide complementary

Table 2
Technical overview and class distribution of the datasets used for experimental evaluation. Values are reported as class proportions in the range [0,1].

Dataset	Instances	Features	Classes	Type	Class distribution (proportion)
X-IIoTID	820,834	63	10	IIoT	Normal (0.5142), RDoS (Ransomware-driven DoS) (0.1720), Reconnaissance (0.1545), Weaponisation (0.0819), Lateral movement (0.0388), Exfiltration (0.0273), Tampering (0.0064), C&C (Command and Control) (0.0033), Exploitation (0.0014), Crypto-ransomware (0.0002).
RT-IoT22	123,117	85	12	IoT	DoS SYN Hping (0.7689), ThingSpeak traffic (0.0659), ARP poisoning (0.0629), MQTT publish (0.0337), Nmap UDP scan (0.0210), Nmap XMAS tree scan (0.0163), Nmap OS detection (0.0162), Nmap TCP scan (0.0081), DoS Slowloris (0.0043), Wipro bulb traffic (0.0021), Metasploit brute-force SSH (0.0003), Nmap FIN scan (0.0002).
Edge-IIoTset	201,393	97	15	IoT/IIoT	Normal (0.7143), DDoS UDP (0.0628), DDoS ICMP (0.0360), Password (0.0266), SQL injection (0.0264), DDoS TCP (0.0261), Vulnerability scanner (0.0260), DDoS HTTP (0.0255), Uploading (0.0193), Backdoor (0.0125), Port scanning (0.0105), XSS (Cross-site scripting) (0.0081), Ransomware (0.0051), Fingerprinting (0.0006), MITM (Man-in-the-middle) (0.0002).

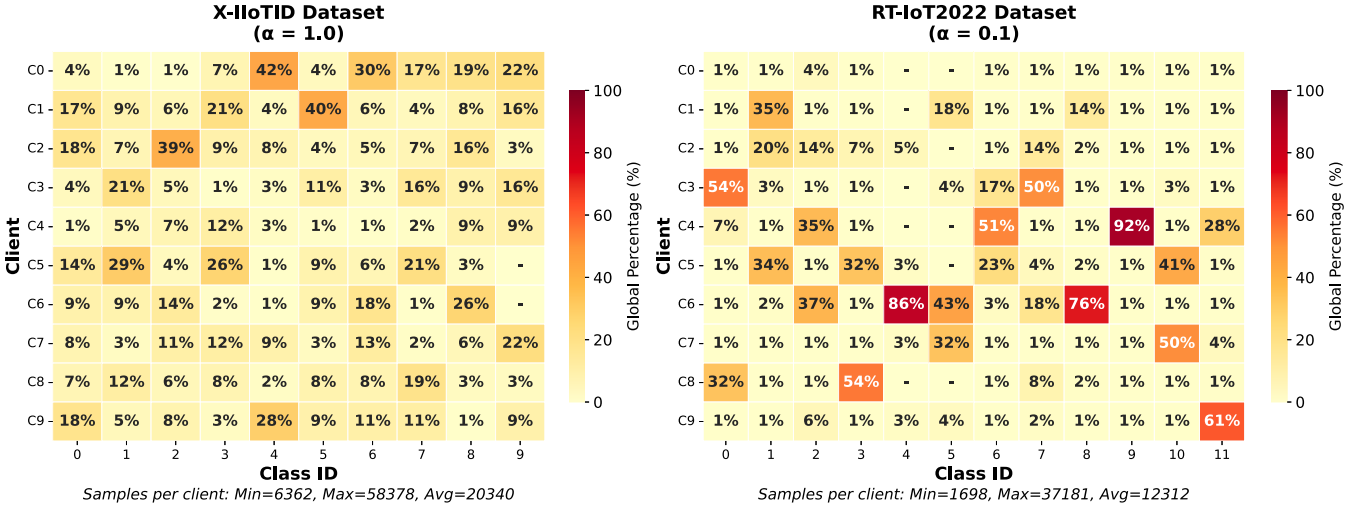

Fig. 3. Example of data partitioning across clients for different α values in X-IIoTID and RT-IoT22 datasets. Each heatmap represents the class composition of a subset of clients (C_0-C_9) obtained from Dirichlet sampling, where darker colours indicate higher class proportions within each client. The columns correspond to class identifiers, while rows represent individual clients. The numbers inside each cell show the percentage of samples from the corresponding class in that client.

perspectives on global optimisation, allowing us to analyse the robustness of our approach alongside our proposed strategy under different coordination dynamics.

FedAvg FedAvg [49] represents the canonical aggregation strategy in FL. After each round of local training, client model updates are averaged proportionally to the number of samples per client, resulting in a new global model. Despite its simplicity, FedAvg remains a strong baseline and is widely adopted due to its efficiency and stability under moderate non-IID conditions.

FedProx FedProx [50] extends FedAvg by adding a proximal term to the local objective, penalising large deviations from the global model during local updates. This modification reduces the negative impact of data heterogeneity and unbalanced client participation. It is therefore particularly suitable for non-IID federations such as those considered in this work.

FedAdam FedAdam [51] adapts the Adam optimiser to the federated setting by maintaining first- and second-moment estimates of global updates on the server. This enables faster and more stable convergence, particularly in scenarios involving noisy or inconsistent local updates. It represents a strong adaptive baseline for comparison by combining the adaptivity of Adam with the communication efficiency of FedAvg. *SCAFFOLD* SCAFFOLD [52] addresses the issue of client drift in heterogeneous settings by introducing control variates to correct local updates. By explicitly estimating the update direction variance between the client and the server, it ensures that local optimisation steps remain aligned with the global objective. This mechanism makes it a critical baseline for evaluating performance in strongly non-IID environments.

For the supervised learning component, we selected four models widely adopted in intrusion detection: LR, SVM, MLP, and CNN. LR and SVM serve as strong linear baselines for tabular and network flow data, offering interpretable decision boundaries and low computational cost [53]. In contrast, MLP and CNN have been shown to provide higher representational capacity and have been increasingly applied to intrusion detection tasks. These algorithms have been demonstrated to capture complex feature interactions despite the non-spatial nature of the data [54]. The integration of both traditional and deep models facilitates a balanced evaluation across varying levels of model complexity. This, in turn, enables a clearer assessment of how aggregation and clustering behave under different learning capacities within the federated framework.

4.4. Evaluation metrics

To assess the performance of the model in multiclass intrusion detection, we use metrics derived from the confusion matrix, which is built from the model's predictions on network traffic instances. The matrix comprises four key indicators: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). These indicators form the basis for the analysis of classification performance in federated environments. In this study, these indicators are defined as follows:

- TP: Positive instances correctly identified. Network flows belonging to a specific attack type that are correctly classified by the model.
- TN: Negative instances correctly identified. Benign flows correctly classified as non-attacks.
- FP: Negative instances incorrectly classified as positive. Benign or unrelated flows misclassified as belonging to an attack type.
- FN: Positive instances incorrectly classified as negative. Flows that belong to an attack type but are misclassified as benign or another attack.

From these indicators, several performance metrics are derived: precision, recall, F1-score, and accuracy.

Precision This metric, given by Eq. (6), measures the proportion of positive predictions that are correct.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6}$$

Recall This metric is defined in Eq. (7) and quantifies the model's ability to correctly identify all positive instances, reflecting its detection coverage.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

F1-score The F1-score, defined in Eq. (8), represents the harmonic mean of precision and recall, providing a balanced measure between false alarms and missed detections. It is particularly valuable for heterogeneous datasets or distributions where class imbalance is present.

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

Accuracy Accuracy, given by Eq. (9), indicates the overall proportion of correctly classified samples. Although widely used, this metric can be misleading in highly imbalanced datasets where dominant classes bias the outcome.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{9}$$

In this work, both the F1-score and accuracy are adopted as the main performance metrics. Accuracy provides an intuitive overview of overall model correctness and allows comparison across algorithms, while the F1-score offers a more robust evaluation under heterogeneous and imbalanced conditions. Precision reduces false alarms by quantifying how many detected attacks are indeed true, and recall maximises detection by measuring how many actual attacks are correctly identified. Considering that data in FL-IDSs for IoT and IIoT are often non-IID, with varying device behaviours and exposure to attacks, a combination of F1-score and accuracy is employed. For multiclass evaluation, the reported F1-score corresponds to the weighted F1-score, computed as the average of class-wise F1-scores weighted by their respective supports. This aggregation accounts for the pronounced class imbalance typically observed in intrusion-detection datasets and provides a representative per-sample summary of detection performance [55]. Additionally, the runtime is reported as the average total execution time of the FL process after all training rounds. It provides an overall indicator of computational efficiency.

4.5. Parameter settings

Following the evaluation criteria described above, the experimental setup was configured to ensure reproducibility and comparability across algorithms. The experiments were conducted using α=0.3 and α=0.1. Each experiment was executed for 10 federated rounds, with 3 local epochs per round, which represents a commonly adopted middle ground for comparative evaluation [56]. This setting offers a reasonable balance between convergence and communication cost, which is particularly relevant in resource-constrained IoT environments [57]. Moreover, recent FL-based intrusion-detection studies . To maintain consistency, 15 clients were simulated for LR and MLP models, and 10 clients for SVM and CNN. The MiniBatch K-Means used in the client-side clustering stage employed a batch size of 256, with three initialisations per run to stabilise convergence.

The local optimisers were selected according to each model's characteristics. LR and SVM used the invscaling learning rate schedule, while MLP employed a constant learning rate of 0.001. For CNN, an adaptive learning rate was used, adjusted dynamically during training to account for local convergence stability. In the federated aggregation schemes, FedProx used a proximal coefficient of μ=0.1, whereas FedAdam followed the settings η=0.01, β_1=0.9, β_2=0.99, and τ=10^{-9}.

Regarding our adaptive stabilisation factor, tests were conducted using values between 0.15 and 0.5. Lower values (0.2-0.3) were found to perform better under small α (strongly non-IID conditions), while higher values (0.4-0.5) were more suitable for larger α, where local distributions are more homogeneous and the clustering structure more stable. This behaviour confirms that higher stabilisation facilitates broader exploration of global clusters when client data distributions are more aligned.

The neural network models were configured as follows. The MLP comprised three hidden layers with 128, 64, and 32 neurons, respectively, each followed by ReLU activations. The CNN consisted of two convolutional layers. The first with 32 filters (kernel size 3, stride 1, padding 1) followed by batch normalisation, ReLU activation, and 2×2 max pooling. The second with 64 filters under the same configuration. These architectures were chosen to balance representational capacity and computational efficiency on tabular IoT traffic data.

Table 3
Centralised performance on Edge-IIoTset, RT-IoT22, and X-IIoTID datasets (upper-bound reference for comparison).

Algorithm	Centralised Performance Reference					
	Edge-IIoTset		RT-IoT22		X-IIoTID	
	Acc.	F1-score	Acc.	F1-score	Acc.	F1-score
LR	0.9298	0.9180	0.9368	0.9242	0.9467	0.9408
SVM	0.9298	0.9164	0.9818	0.9818	0.9485	0.9427
MLP	0.9401	0.9304	0.9959	0.9957	0.9855	0.9856
CNN	0.9463	0.9414	0.9959	0.9958	0.9841	0.9839

5. Results and discussion

This section analyses the performance of the proposed approach compared with standard federated strategies. It focusses on convergence, heterogeneity robustness, and efficiency (runtime and bandwidth), alongside an evaluation of clustering stability and cohesion.

5.1. Performance evaluation across rounds

In order to provide a robust upper-bound reference and contextualise the forthcoming federated evaluation, we first report the final performance achieved under a fully centralised training setting. To ensure a fair comparison, these models were trained using the exact same hyperparameter configuration as in the federated experiments. Furthermore, the training process was set to 30 epochs, representing the equivalent effective training load of the federated approach (3 local epochs × 10 communication rounds). This baseline allows us to measure the gap introduced by decentralisation and the integration of the proposed double-clustering strategy, as well as the impact of heterogeneous client distributions. Table 3 summarises the accuracy and F1-score obtained when trained centrally on the Edge-IIoTset, RT-IoT22, and X-IIoTID datasets.

In terms of federated performance, the accuracy and F1-score metrics reported represent the aggregated performance across all clients using their respective assigned models. Tables 4, 5, and 6 detail the convergence and performance achieved on Edge-IIoTset, RT-IoT22, and X-IIoTID at rounds 2, 5, and 8. These tables compare our adaptive double-clustering strategy with FedAvg, FedProx, FedAdam, and

Table 4
Edge-IIoTset - Aggregated performance evolution at rounds 2, 5, and 8 ($\alpha = 0.3$) for each algorithm and strategy.

Algorithm	Strategy	Round					
		2		5		8	
		Acc.	F1-score	Acc.	F1-score	Acc.	F1-score
LR	**Our approach**	0.7213	0.6986	0.8581	0.8478	0.8987	0.8886
	FedAvg	0.8347	0.8395	0.8291	0.8354	0.8593	0.8734
	FedProx	0.8329	0.8156	0.9058	0.8948	0.8724	0.8632
	FedAdam	0.6946	0.6441	0.6873	0.6510	0.7382	0.7139
	SCAFFOLD	0.6648	0.6573	0.7819	0.7764	0.8624	0.8551
SVM	**Our approach**	0.8533	0.8424	0.8846	0.8826	0.9137	0.9037
	FedAvg	0.8661	0.8548	0.9043	0.9029	0.9361	0.9313
	FedProx	0.8248	0.8123	0.9470	0.9436	0.9090	0.9066
	FedAdam	0.7634	0.7249	0.8589	0.8282	0.8565	0.8279
	SCAFFOLD	0.7741	0.7555	0.7878	0.7754	0.7692	0.7591
MLP	**Our approach**	0.8608	0.8453	0.8695	0.8668	0.9211	0.9177
	FedAvg	0.5826	0.5117	0.6203	0.5570	0.6667	0.6247
	FedProx	0.6367	0.5651	0.7042	0.6321	0.6858	0.6048
	FedAdam	0.6906	0.6085	0.6763	0.6115	0.7452	0.6740
	SCAFFOLD	0.2498	0.2421	0.7342	0.7291	0.5951	0.5796
CNN	**Our approach**	0.8926	0.8869	0.9275	0.9244	0.9227	0.9178
	FedAvg	0.8130	0.8081	0.7114	0.7103	0.8839	0.8900
	FedProx	0.7959	0.7925	0.8851	0.8798	0.7747	0.7735
	FedAdam	0.8193	0.7983	0.6582	0.6199	0.5964	0.5634
	SCAFFOLD	0.7234	0.6934	0.8733	0.8689	0.8239	0.8141

Table 5
RT-IoT22 - Aggregated performance evolution at rounds 2, 5, and 8 ($\alpha = 0.3$) for each algorithm and strategy.

Algorithm	Strategy	Round					
		2		5		8	
		Acc.	F1-score	Acc.	F1-score	Acc.	F1-score
LR	**Our approach**	0.9069	0.8910	0.9408	0.9300	0.9211	0.9116
	FedAvg	0.8268	0.7961	0.8110	0.7957	0.9573	0.9512
	FedProx	0.8274	0.8012	0.9230	0.9119	0.9014	0.8915
	FedAdam	0.8413	0.8082	0.8520	0.8334	0.7930	0.7522
	SCAFFOLD	0.8909	0.8849	0.9145	0.9262	0.8800	0.8872
SVM	**Our approach**	0.9513	0.9395	0.9437	0.9319	0.9594	0.9507
	FedAvg	0.8325	0.8133	0.9627	0.9586	0.9488	0.9462
	FedProx	0.9112	0.9011	0.8809	0.8719	0.9378	0.9323
	FedAdam	0.8866	0.8754	0.9301	0.9227	0.8912	0.8767
	SCAFFOLD	0.8525	0.8458	0.9243	0.9178	0.8905	0.8959
MLP	**Our approach**	0.8756	0.8585	0.9398	0.9384	0.9614	0.9644
	FedAvg	0.8787	0.8069	0.8932	0.8293	0.9196	0.8482
	FedProx	0.7998	0.7228	0.9118	0.8431	0.8779	0.8071
	FedAdam	0.8912	0.8714	0.9182	0.9281	0.9312	0.9201
	SCAFFOLD	0.6720	0.6348	0.6523	0.6261	0.7152	0.7070
CNN	**Our approach**	0.9459	0.9449	0.9648	0.9649	0.9717	0.9739
	FedAvg	0.9153	0.9034	0.8809	0.9185	0.9599	0.9634
	FedProx	0.7421	0.7474	0.9014	0.8885	0.9223	0.9205
	FedAdam	0.5417	0.5618	0.7509	0.7794	0.7925	0.8074
	SCAFFOLD	0.8543	0.8539	0.9289	0.9347	0.8931	0.8994

SCAFFOLD under $\alpha = 0.3$. The findings reveal enhancements for the proposed approach, particularly in the middle and late stages of training. Across the three datasets, our method achieves more stable trajectories throughout the training process. It also achieves higher accuracy/F1-score combinations, particularly for MLP and CNN models, which are typically the most sensitive to heterogeneous client distributions. In contrast, standard strategies such as FedAvg, FedProx, and FedAdam exhibit pronounced variability across rounds, and in several cases experience substantial performance drops under non-IID conditions. Notably, SCAFFOLD mitigates some instability, yet it remains clearly below the performance reached by our scheme.

Table 6
X-IIoTID - Aggregated performance evolution at rounds 2, 5, and 8 ($\alpha = 0.3$) for each algorithm and strategy.

Algorithm	Strategy	Round					
		2		5		8	
		Acc.	F1-score	Acc.	F1-score	Acc.	F1-score
LR	**Our approach**	0.9430	0.9392	0.9294	0.9263	0.9397	0.9365
	FedAvg	0.8590	0.8688	0.8730	0.8634	0.8798	0.8787
	FedProx	0.8331	0.8126	0.8899	0.8918	0.9254	0.9297
	FedAdam	0.7338	0.7912	0.8010	0.8355	0.6792	0.7554
	SCAFFOLD	0.8729	0.8715	0.8788	0.8697	0.8641	0.8736
SVM	**Our approach**	0.9421	0.9369	0.9361	0.9348	0.9470	0.9445
	FedAvg	0.8989	0.9026	0.9365	0.9356	0.9250	0.9287
	FedProx	0.8644	0.8739	0.9171	0.9266	0.9249	0.9279
	FedAdam	0.5316	0.5906	0.5989	0.6682	0.6857	0.7539
	SCAFFOLD	0.5877	0.5898	0.9084	0.9164	0.7638	0.7646
MLP	**Our approach**	0.8439	0.8468	0.8940	0.9090	0.9390	0.9457
	FedAvg	0.5722	0.4908	0.5834	0.4964	0.4958	0.4021
	FedProx	0.5146	0.4247	0.4767	0.3988	0.4484	0.3392
	FedAdam	0.5446	0.4571	0.5050	0.3918	0.4748	0.3919
	SCAFFOLD	0.7187	0.6454	0.6026	0.6175	0.6482	0.6479
CNN	**Our approach**	0.9442	0.9447	0.9663	0.9680	0.9602	0.9621
	FedAvg	0.8214	0.8024	0.8854	0.9033	0.9344	0.9506
	FedProx	0.7823	0.7618	0.8850	0.8791	0.9533	0.9571
	FedAdam	0.6241	0.5494	0.7209	0.7060	0.8167	0.8026
	SCAFFOLD	0.7570	0.7839	0.9322	0.9410	0.7595	0.7557

Table 7
Comparison of final aggregated performance (Round 10) on Edge-IIoTset, RT-IoT22, and X-IIoTID datasets ($\alpha = 0.3$) for each algorithm and strategy.

Algorithm	Strategy	Final Performance (Round 10)					
		Edge-IIoTset		RT-IoT22		X-IIoTID	
		Acc.	F1-score	Acc.	F1-score	Acc.	F1-score
LR	**Our approach**	**0.9168**	**0.9096**	**0.9270**	**0.9172**	**0.9265**	**0.9234**
	FedAvg	0.8514	0.8602	0.8990	0.9100	0.9076	0.9086
	FedProx	0.8901	0.8783	0.8550	0.8402	0.9007	0.9044
	FedAdam	0.7411	0.7210	0.8991	0.8792	0.8273	0.8576
	SCAFFOLD	0.8299	0.8167	0.9191	0.9125	0.8840	0.8924
SVM	**Our approach**	**0.9293**	**0.9240**	**0.9634**	**0.9576**	**0.9577**	**0.9552**
	FedAvg	0.9095	0.9023	0.9571	0.9547	0.9331	0.9374
	FedProx	0.8923	0.8826	0.9081	0.9025	0.9486	0.9455
	FedAdam	0.8859	0.8646	0.8953	0.8918	0.7210	0.7827
	SCAFFOLD	0.8786	0.8648	0.8995	0.9100	0.8990	0.9009
MLP	**Our approach**	**0.9193**	**0.9152**	**0.9748**	**0.9764**	**0.9423**	**0.9507**
	FedAvg	0.7549	0.7055	0.8439	0.7699	0.5413	0.4520
	FedProx	0.7779	0.6787	0.9335	0.8708	0.5554	0.4618
	FedAdam	0.6915	0.6276	0.8998	0.8874	0.5954	0.5238
	SCAFFOLD	0.7872	0.7543	0.8095	0.8150	0.6378	0.6269
CNN	**Our approach**	**0.9383**	**0.9353**	**0.9859**	**0.9858**	**0.9738**	**0.9748**
	FedAvg	0.8445	0.8449	0.8474	0.8450	0.8839	0.9005
	FedProx	0.8726	0.8719	0.9303	0.9333	0.8160	0.8503
	FedAdam	0.5332	0.5089	0.7559	0.7467	0.8330	0.8434
	SCAFFOLD	0.8270	0.8225	0.8722	0.8766	0.8761	0.8825

Table 7 presents the final F1-score and accuracy achieved at Round 10. The results demonstrate improvements in both metrics for the proposed approach across all datasets and models. For linear models (LR and SVM), the proposed strategy reaches final F1-scores above 0.91 and 0.92 in the RT-IoT22 and X-IIoTID datasets, respectively. This confirms that distribution-aware grouping benefits even models with lower capacity. However, the advantage is most critical for DL models (MLP and CNN), where baselines struggle significantly under non-IID conditions ($\alpha = 0.3$). In X-IIoTID, for instance, FedAvg and FedProx drop to F1-scores as low as 0.45–0.46 for MLP. By contrast, our approach maintains a performance of 0.95. A similar pattern is observed for CNNs in Edge-IIoTset. FedAdam exhibits severe instability (F1-score $\approx$ 0.50), whereas

Table 8
Edge-IIoTset - Aggregated performance evolution at rounds 2, 5, and 8 ($\alpha = 0.1$) for each algorithm and strategy.

Algorithm	Strategy	Round					
		2		5		8	
		Acc.	F1-score	Acc.	F1-score	Acc.	F1-score
LR	**Our approach**	0.6548	0.6309	0.8019	0.7999	0.8710	0.8663
	FedAvg	0.8480	0.8480	0.7529	0.7529	0.8641	0.8641
	FedProx	0.8017	0.7900	0.9012	0.8993	0.8491	0.8489
	FedAdam	0.6063	0.5587	0.5752	0.5231	0.7351	0.7010
	SCAFFOLD	0.4262	0.4239	0.7030	0.7128	0.8855	0.8864
SVM	**Our approach**	0.5680	0.5760	0.8196	0.8147	0.8691	0.8847
	FedAvg	0.8522	0.8404	0.8526	0.8552	0.8530	0.8597
	FedProx	0.6515	0.6321	0.8857	0.8837	0.8831	0.8813
	FedAdam	0.7945	0.7616	0.6880	0.6524	0.7672	0.7558
	SCAFFOLD	0.8383	0.8252	0.7771	0.7787	0.7685	0.7633
MLP	**Our approach**	0.7884	0.7787	0.8913	0.8918	0.8067	0.8091
	FedAvg	0.7101	0.6621	0.7412	0.7199	0.6888	0.6377
	FedProx	0.7238	0.6770	0.6641	0.6234	0.7856	0.7478
	FedAdam	0.6089	0.5563	0.5214	0.4628	0.8483	0.8208
	SCAFFOLD	0.2768	0.2565	0.6563	0.6497	0.5139	0.4975
CNN	**Our approach**	0.8548	0.8492	0.9165	0.9083	0.9148	0.9101
	FedAvg	0.8934	0.8895	0.8856	0.8799	0.9624	0.9611
	FedProx	0.6151	0.6071	0.7970	0.7908	0.8848	0.8821
	FedAdam	0.5726	0.5286	0.6911	0.6704	0.6434	0.6362
	SCAFFOLD	0.7359	0.7295	0.8518	0.8435	0.8692	0.8634

Table 9
RT-IoT22 - Aggregated performance evolution at rounds 2, 5, and 8 ($\alpha = 0.1$) for each algorithm and strategy.

Algorithm	Strategy	Round					
		2		5		8	
		Acc.	F1-score	Acc.	F1-score	Acc.	F1-score
LR	**Our approach**	0.9431	0.9351	0.9672	0.9601	0.9695	0.9614
	FedAvg	0.8359	0.8132	0.8187	0.7984	0.6764	0.6579
	FedProx	0.7591	0.7532	0.8604	0.8450	0.7030	0.6601
	FedAdam	0.7069	0.7408	0.7664	0.7947	0.8209	0.8547
	SCAFFOLD	0.7975	0.8047	0.9372	0.9390	0.7577	0.7637
SVM	**Our approach**	0.9856	0.9799	0.9864	0.9816	0.9792	0.9775
	FedAvg	0.8001	0.8018	0.9164	0.9039	0.8375	0.8499
	FedProx	0.7504	0.7303	0.8737	0.8670	0.8897	0.8818
	FedAdam	0.8222	0.8219	0.8224	0.8109	0.8561	0.8514
	SCAFFOLD	0.3885	0.4745	0.3394	0.3838	0.3871	0.4576
MLP	**Our approach**	0.6832	0.6699	0.8504	0.8419	0.9264	0.9273
	FedAvg	0.6068	0.5463	0.7480	0.7095	0.8258	0.8021
	FedProx	0.7931	0.7383	0.7649	0.6404	0.7650	0.7269
	FedAdam	0.6872	0.6394	0.8568	0.8327	0.7066	0.6831
	SCAFFOLD	0.7252	0.6934	0.2388	0.2219	0.8543	0.8320
CNN	**Our approach**	0.9065	0.9038	0.9339	0.9487	0.8918	0.8917
	FedAvg	0.8385	0.8078	0.9033	0.9000	0.9690	0.9715
	FedProx	0.8328	0.8334	0.8674	0.8545	0.8963	0.8993
	FedAdam	0.4086	0.4068	0.5681	0.5620	0.6004	0.6010
	SCAFFOLD	0.3850	0.3789	0.8487	0.8530	0.8511	0.8552

our method achieves 0.93, preserving model utility even when standard aggregation leads to collapse.

From an optimisation perspective, the instability observed in FedAdam can be attributed to its reliance on global moment estimation. Under highly non-IID regimes, client updates exhibit large directional dispersion. As a result, the server-side momentum accumulates incompatible descent signals instead of a consistent trend, amplifying oscillations. Our approach reduces the variance and directional inconsistency of the aggregated update, yielding smoother optimisation dynamics across rounds. Moreover, explicit outlier handling prevents atypical updates from contaminating the family experts.

Table 10
X-IIoTID - Aggregated performance evolution at rounds 2, 5, and 8 ($\alpha = 0.1$) for each algorithm and strategy.

Algorithm	Strategy	Round					
		2		5		8	
		Acc.	F1-score	Acc.	F1-score	Acc.	F1-score
LR	**Our approach**	0.9399	0.9354	0.9100	0.9127	0.9312	0.9404
	FedAvg	0.7522	0.7843	0.8152	0.8414	0.8917	0.9047
	FedProx	0.8219	0.8501	0.9007	0.9073	0.8784	0.9001
	FedAdam	0.5827	0.6628	0.6623	0.7532	0.7406	0.8066
	SCAFFOLD	0.4254	0.4701	0.4204	0.4408	0.6873	0.7170
SVM	**Our approach**	0.9266	0.9214	0.9484	0.9465	0.9648	0.9639
	FedAvg	0.7832	0.8141	0.8506	0.8668	0.8738	0.8836
	FedProx	0.8002	0.8318	0.8964	0.9066	0.8952	0.9121
	FedAdam	0.5647	0.6633	0.6711	0.7470	0.6918	0.7643
	SCAFFOLD	0.6135	0.6588	0.6651	0.6785	0.7849	0.8386
MLP	**Our approach**	0.8325	0.8236	0.8797	0.8894	0.8913	0.9146
	FedAvg	0.3911	0.3353	0.3026	0.2394	0.5329	0.4764
	FedProx	0.5257	0.4666	0.5176	0.4597	0.6465	0.5752
	FedAdam	0.3045	0.1825	0.4152	0.3561	0.5381	0.4746
	SCAFFOLD	0.5561	0.5463	0.4447	0.4609	0.3317	0.2600
CNN	**Our approach**	0.6807	0.7327	0.7812	0.7688	0.8238	0.8148
	FedAvg	0.8482	0.8692	0.9071	0.9076	0.8669	0.8709
	FedProx	0.5168	0.4751	0.8310	0.8247	0.9169	0.9276
	FedAdam	0.4169	0.3736	0.5207	0.4746	0.6164	0.6061
	SCAFFOLD	0.5104	0.5220	0.2436	0.2628	0.4598	0.3873

Tables 8, 9, and 10 detail the convergence and performance achieved on Edge-IIoTset, RT-IoT22, and X-IIoTID at Rounds 2, 5, and 8 under $\alpha = 0.1$. This setting represents a highly heterogeneous (strongly non-IID) scenario, in which aggregation strategies are expected to experience larger gradient conflicts and higher variability across rounds.

The results show that the proposed family-wise aggregation provides a more reliable convergence trajectory under extreme heterogeneity. From the middle rounds onwards, it delivers a strong balance between accuracy and F1-score. In contrast, standard approaches remain sensitive to round-to-round instability and exhibit performance variations across datasets. In RT-IoT22, our method achieves consistently high performance across LR, SVM, MLP, and CNN from round 2 onwards, whereas FedAvg and FedProx show significant performance degradation for certain models (e.g. LR at round 8). In Edge-IIoTset and X-IIoTID, several baselines display pronounced fluctuations across rounds, reflecting the difficulty of maintaining stable optimisation under severe non-IID partitions. Although certain baseline configurations may achieve higher values at specific intermediate checkpoints (notably for CNN in Edge-IIoTset and X-IIoTID), these gains are not consistently sustained across rounds. Notably, SCAFFOLD mitigates instability in some cases, but it can also become erratic under $\alpha = 0.1$, particularly for margin-based models, where performance can collapse at early and mid rounds.

Table 11 reports the final accuracy and F1-score at Round 10 for $\alpha = 0.1$. The proposed approach achieves the best final performance across all datasets and model families, confirming that the stability observed in the intermediate checkpoints translates into superior end-state detectors. The gains are especially pronounced for DL models, where baselines struggle most under severe non-IID conditions. For instance, in X-IIoTID with MLP, our method reaches an F1-score of 0.9535, whereas FedAvg and FedProx remain at 0.3993 and 0.4576, respectively. Similarly, in RT-IoT22 with MLP, the proposed strategy attains an F1-score of 0.9784, while SCAFFOLD drops to 0.4503. These results reinforce that clustering-based aggregation is particularly effective at protecting higher-capacity models from the divergence and noise amplification that can undermine conventional federated optimisers in highly adversarial IoT settings.

Finally, comparing these results with the centralised baseline shows that the proposed architecture narrows the performance gap associated with distributed learning. In most scenarios, our method retains more

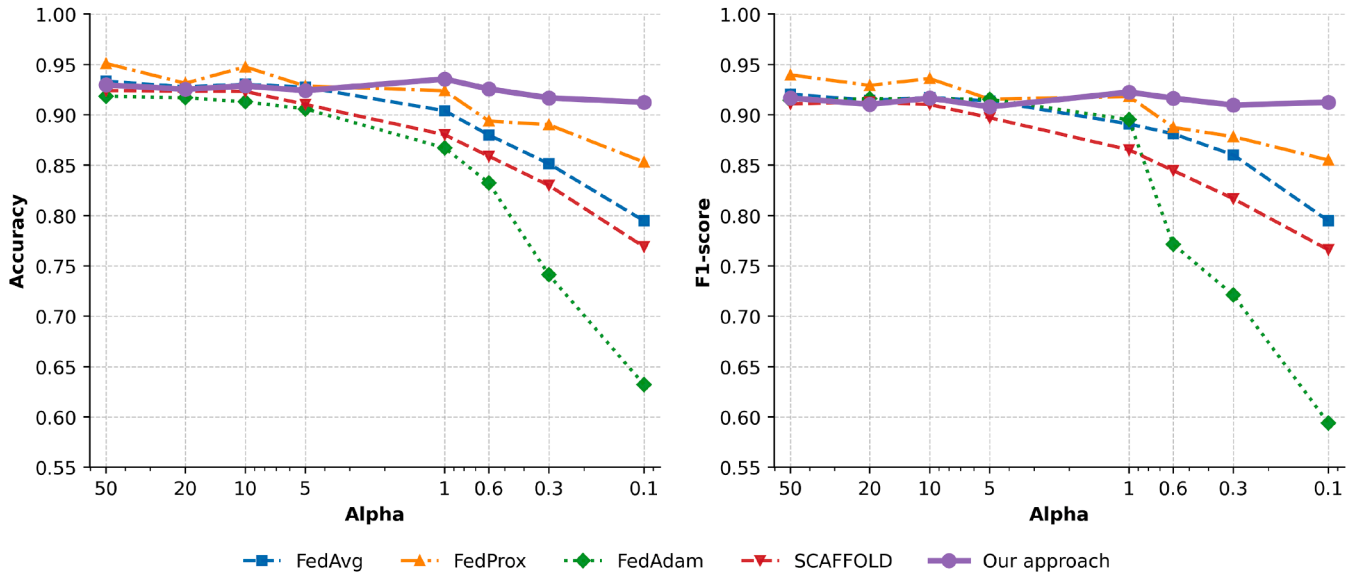

Fig. 4. Final aggregated accuracy and F1-score versus Dirichlet α on Edge-IIoTset using the LR model (10 rounds)..

Table 11
Comparison of final aggregated performance (Round 10) on Edge-IIoTset, RT-IoT22, and X-IIoTID datasets ($\alpha = 0.1$) for each algorithm and strategy.

Algorithm	Strategy	Final Performance (Round 10)					
		Edge-IIoTset		RT-IoT22		X-IIoTID	
		Acc.	F1-score	Acc.	F1-score	Acc.	F1-score
LR	**Our approach**	**0.9124**	**0.9124**	**0.9760**	**0.9700**	**0.9294**	**0.9392**
	FedAvg	0.7947	0.7947	0.9026	0.8927	0.8518	0.8734
	FedProx	0.8530	0.8550	0.8317	0.8075	0.8833	0.8895
	FedAdam	0.6321	0.5938	0.8508	0.8855	0.7539	0.8134
	SCAFFOLD	0.7689	0.7659	0.8940	0.9094	0.8231	0.8573
SVM	**Our approach**	**0.9168**	**0.9127**	**0.9885**	**0.9848**	**0.9578**	**0.9573**
	FedAvg	0.8721	0.8781	0.9513	0.9524	0.8999	0.9194
	FedProx	0.8299	0.8334	0.8621	0.8764	0.9041	0.9088
	FedAdam	0.7723	0.7540	0.8852	0.8763	0.6652	0.7596
	SCAFFOLD	0.7652	0.7555	0.7974	0.8613	0.7984	0.8354
MLP	**Our approach**	**0.8318**	**0.8241**	**0.9760**	**0.9784**	**0.9350**	**0.9535**
	FedAvg	0.6730	0.6274	0.5012	0.4381	0.4467	0.3993
	FedProx	0.6472	0.6036	0.6576	0.6097	0.4957	0.4576
	FedAdam	0.5697	0.5203	0.7059	0.6649	0.5689	0.5200
	SCAFFOLD	0.7098	0.6808	0.4321	0.4503	0.5890	0.5486
CNN	**Our approach**	**0.8817**	**0.8783**	**0.9505**	**0.9484**	**0.8622**	**0.8912**
	FedAvg	0.7354	0.7299	0.8058	0.8193	0.7214	0.7281
	FedProx	0.8436	0.8429	0.8883	0.8865	0.7853	0.8260
	FedAdam	0.4508	0.4374	0.6764	0.6738	0.6627	0.6056
	SCAFFOLD	0.8107	0.8037	0.8110	0.8117	0.4442	0.3997

Table 12
Relative variation in total runtime (Our vs. reference). Negative values indicate shorter runtime (better).

Dataset	Algorithm	vs. FedAvg	vs. FedProx	vs. FedAdam	vs. SCAFFOLD
X-IIoTID	LR	−0.9%	+1.5%	−2.4%	−3.4%
	SVM	−0.3%	+2.3%	+5.6%	−3.7%
	MLP	+5.3%	−1.8%	+0.4%	−2.3%
	CNN	+1.6%	+8.0%	+4.3%	−1.4%
RT-IoT22	LR	+12.2%	+10.1%	+11.8%	+7.1%
	SVM	**+12.7%**	+11.9%	+12.6%	+5.4%
	MLP	+9.3%	+11.2%	+12.0%	−6.1%
	CNN	−2.6%	+10.5%	−4.6%	+3.4%
Edge-IIoTset	LR	+1.3%	+2.0%	−0.7%	+0.5%
	SVM	+9.6%	+1.8%	−4.2%	+8.2%
	MLP	+9.8%	+6.4%	+12.4%	+1.9%
	CNN	+1.7%	−14.6%	**−15.3%**	−4.8%

than 95% of the centralised model's performance in terms of both accuracy and F1-score. In addition, a few configurations even surpass the centralised reference. For instance, on X-IIoTID with SVM, the federated model achieves a higher F1-score than the centralised baseline (0.9552 vs 0.9427). Similar behaviour is observed on Edge-IIoTset for SVM. This phenomenon can be attributed to the specialised nature of the double-clustering mechanism. A centralised model is typically forced to optimise a single decision boundary across conflicting non-IID distributions, often resulting in generalisation noise. In contrast, the proposed architecture functions as a distributed ensemble, where decision boundaries are inherently tailored to specific sub-distributions. Rather than being compromised by a global average, the system transforms data heterogeneity from a liability into a structural advantage.

5.2. Sensitivity to data heterogeneity

To evaluate the robustness of the proposed system under different conditions of non-IID data, we analysed how varying the Dirichlet parameter α affects model performance. Figs. 4-6 show the evolution of accuracy and F1-score across all datasets as α decreases from 50 to 0.1, using the strategies described in Section 4.3. Specifically, Fig. 4 utilises LR as a lightweight baseline, whereas Figs. 5 and 6 employ MLP and CNN, respectively, as more complex DL models.

Across all datasets, our approach exhibits superior stability and accuracy as heterogeneity increases. When α drops below 1, the baselines experience a sharp degradation in both metrics. This is particularly notable in adaptive strategies such as FedAdam and SCAFFOLD, which struggle to estimate stable gradients or control variates when local distributions become extremely skewed. FedProx partially mitigates this effect through its proximal term, but its effectiveness decreases under extreme heterogeneity. In contrast, our method sustains gradual performance decay, maintaining over 90% of its peak F1-score even at α=0.1. These results demonstrate that aligning aggregation based on behavioural similarity, through local micro-clustering and family-wise grouping, can effectively mitigate the adverse effects of distributional skew. This alignment helps prevent negative transfer across different clients.

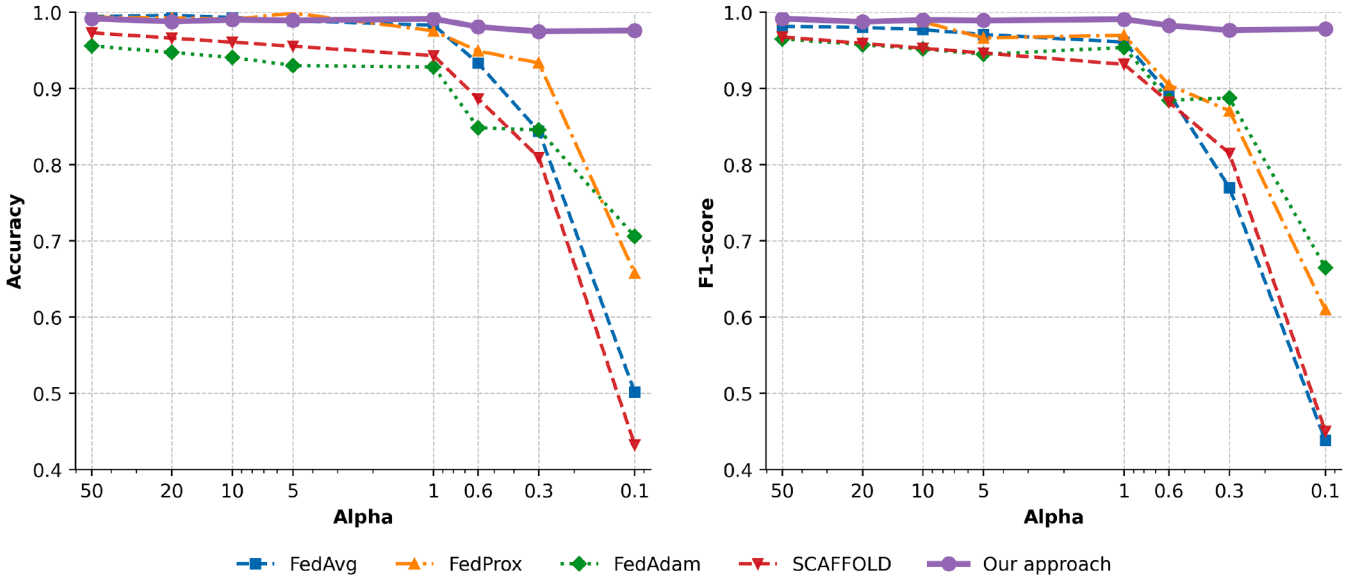

Fig. 5. Final aggregated accuracy and F1-score versus Dirichlet α on RT-IoT22 using the MLP model (10 rounds).

Table 13
Per-round training time, transition time, and client upload payload using the MLP model on Edge-IIoTset. Upload denotes the average per-client effective payload sent to the server per round.

Method	Nodes	t_{train} [s]	t_{trans} [s]	Upload [KB/rnd]
Our approach	15	1.2989	0.0947	83.84
	50	0.5841	0.1069	83.88
	100	0.2873	0.1102	83.94
	200	0.1243	0.1510	84.00
FedAvg	15	1.3106	0.0105	82.81
	50	0.6368	0.0243	82.81
	100	0.2845	0.0359	82.81
	200	0.1298	0.0612	82.81
FedProx	15	1.7814	0.0096	82.81
	50	0.6738	0.0231	82.81
	100	0.3585	0.0385	82.81
	200	0.1816	0.0593	82.81
FedAdam	15	1.2934	0.0104	82.81
	50	0.5949	0.0219	82.81
	100	0.2884	0.0343	82.81
	200	0.1344	0.0600	82.81
SCAFFOLD	15	1.5911	0.0214	165.62
	50	0.6639	0.0385	165.62
	100	0.3307	0.0574	165.62
	200	0.1587	0.1053	165.62

Table 14
Comparative analysis of total bandwidth consumption for all evaluated algorithms and strategies on the Edge-IIoTset dataset.

Model	Nodes	Total bandwidth (MB)				
		Our approach	FedAvg	FedProx	FedAdam	SCAFFOLD
LR	10	9.04	8.70	8.68	8.72	12.77
	20	15.83	15.08	15.10	15.08	22.84
	50	36.34	34.27	34.27	34.29	53.00
	100	70.59	66.29	66.42	66.31	103.35
	200	138.87	130.80	130.97	131.05	203.95
SVM	10	8.95	8.72	8.68	8.75	12.81
	20	15.75	14.97	14.99	14.95	22.86
	50	36.22	34.20	34.08	34.06	52.97
	100	70.36	66.19	66.25	66.21	103.20
	200	138.45	130.32	130.46	129.86	203.80
MLP	10	7.38	7.51	7.32	7.42	10.63
	20	11.77	11.91	11.95	11.89	18.62
	50	25.31	24.81	24.77	24.77	38.00
	100	48.34	46.26	46.10	46.14	70.95
	200	95.92	89.63	89.32	89.72	136.36
CNN	10	8.93	9.02	9.40	9.33	13.76
	20	13.90	13.88	13.32	13.38	21.45
	50	26.49	27.26	26.40	26.26	42.09
	100	48.59	48.19	48.55	48.28	76.42
	200	95.65	92.00	92.34	91.77	144.60

The dataset-wise analysis further confirms this trend. On the Edge-IIoTset dataset, performance degradation in FedAvg and FedAdam becomes evident from α<5, with both losing over 20–25% of their peak F1-score as heterogeneity increases. SCAFFOLD also degrades noticeably in the most heterogeneous setting. In contrast, our approach maintains around 93–95% of its maximum performance, showing strong tolerance to distributional imbalance. For RT-IoT22, the decline in baseline strategies starts later, near α=1, and intensifies under higher heterogeneity. Notably, at α=0.1, standard FedAvg and the drift-correction mechanism of SCAFFOLD collapse, with F1-scores dropping below 0.5. FedAdam retains slightly better robustness ($\approx$ 0.65) but still significantly underperforms compared to our approach, which preserves about 97% of its best F1-score (0.96). In X-IIoTID, where the initial performance of all methods is close to saturation, degradation remains modest until α<0.5. However, at α=0.1, both SCAFFOLD and FedAdam crash to 0.40–0.45, while FedAvg manages to retain an F1-score of $\approx$ 0.85. Only our method demonstrates consistent stability, sustaining more than 95% of its original accuracy across the entire range.

Overall, the increasing gap between our method and the baselines for smaller α values confirms that neither standard averaging nor conventional drift-correction techniques (like SCAFFOLD) are sufficient to cope with the extreme divergence of IoT traffic. The consistent detection performance observed across the three datasets highlights the adaptability of the proposed architecture. This consistency further demonstrates its robustness, even under highly imbalanced and non-stationary IoT environments.

5.3. Operational efficiency and structural dynamics

To assess the computational cost of the proposed architecture, Table 12 reports the relative variation in total execution time with respect to the baseline strategies. A negative percentage indicates a

Table 15
Structure and loyalty metrics using the MLP model (15 clients) on Edge-IIoTset, grouped by heterogeneity factor (α) and stability threshold (τ).

Heterogeneity (α)	Stability threshold	Structure and cohesion			Individual loyalty		
		Final families	Peer consistency [%]	Avg. noise [%]	Client persistence [%]	Perfect clients [%]	Avg. stay [rounds]
50	0.5	2	39.82	0.00	72.59	0.00	3.24
	0.3	3	43.77	0.00	77.04	0.00	3.47
	0.1	2	90.61	0.00	96.30	66.67	8.33
1	0.5	3	28.20	0.00	63.70	0.00	2.44
	0.3	4	33.08	0.00	68.89	6.67	3.21
	0.1	4	74.52	0.00	88.89	20.00	5.67
0.1	0.5	5	27.50	3.33	52.59	0.00	2.18
	0.3	6	29.96	2.00	54.81	0.00	2.24
	0.1	5	33.99	3.33	64.44	0.00	2.62

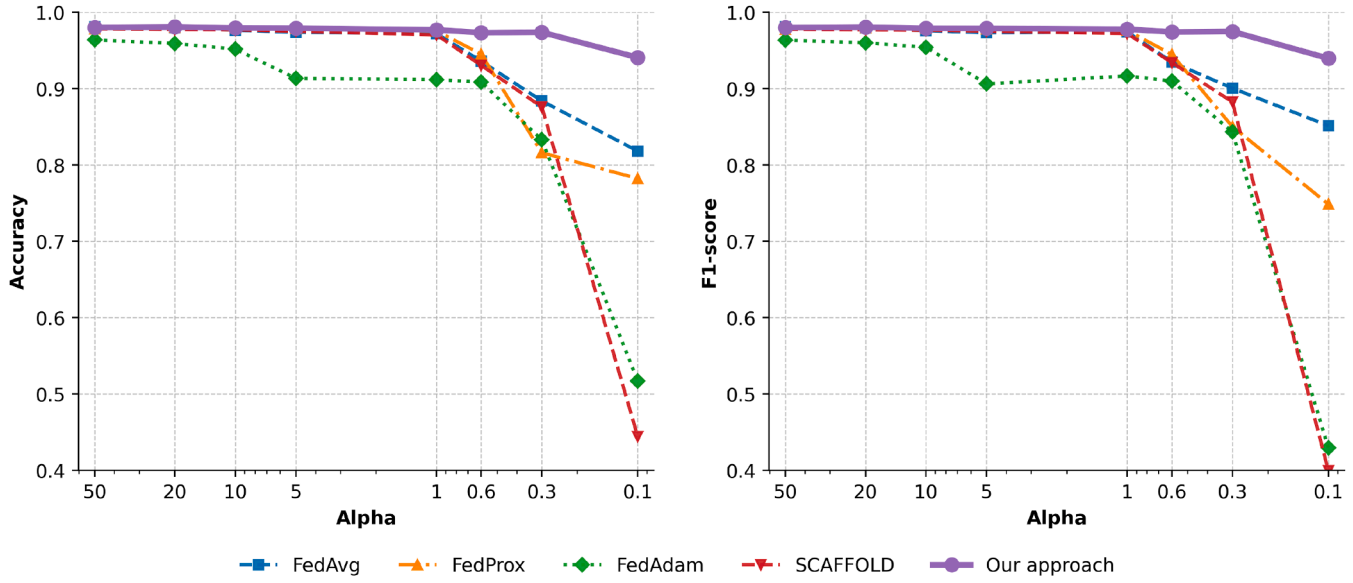

Fig. 6. Final aggregated accuracy and F1-score versus Dirichlet α on X-IIoTID using the CNN model (10 rounds).

reduction in runtime, while a positive value denotes additional overhead introduced by the clustering and adaptive aggregation processes. The analysis was conducted under identical hardware and communication conditions to ensure comparability across all methods.

Overall, the runtime impact remains moderate, with a global range between −15.3% and +12.7%. On X-IIoTID, execution times are largely comparable across strategies, with deviations within ±5%. In RT-IoT22, the proposed method shows an average increase of about 10% due to the higher number of client families formed in this more heterogeneous setting, which slightly extends the clustering stage. Conversely, in Edge-IIoTset, where the client groups stabilise more quickly, runtime reductions of up to 15% are observed, especially for CNN models. This improvement stems from the reduced number of local updates required for convergence once stable families are established.

These results confirm that the additional processing introduced by our mechanism does not result in significant computational overhead. In fact, in several cases, it accelerates convergence by improving model alignment and reducing oscillations between rounds. A notable finding is the competitive operational profile compared against SCAFFOLD. Our architecture achieves total execution times that are largely comparable to this baseline, and frequently lower, particularly in X-IIoTID (reductions between 1.4% and 3.7%) and for MLP in RT-IoT22 (reduction of 6.1%). This indicates that the overhead of computing control variates in SCAFFOLD is often equivalent to, or exceeds, the cost of our clustering mechanism. Consequently, the proposed strategy enhances detection without imposing a heavier computational overhead.

To provide a granular analysis of scalability and resource consumption, Table 13 details the system performance across varying network size (15 to 200 nodes). Two key metrics are reported alongside training time (t_{train}). The transition time (t_{trans}), defined as the latency from the moment a client completes local training until it receives the updated global parameters (encompassing local clustering, server-side clustering, aggregation, and distribution); and the upload payload, which measures the average effective volume of useful data transmitted by each client to the server per round.

In terms of local computation, the proposed approach maintains a training efficiency comparable to lightweight baselines like FedAvg ($t_{\text{train}} \approx 0.12$ s at 200 nodes), avoiding the additional computational overhead observed in FedProx and SCAFFOLD. Specifically, FedProx incurs extra cost by computing an additional proximal regularisation term, whereas SCAFFOLD adds latency through the extra arithmetic operations required to correct local gradients.

Regarding transition latency, baselines such as FedAvg, FedProx, and FedAdam exhibit similar and minimal overheads. This is expected, as their server-side operations are limited to standard parameter averaging or efficient momentum updates. In contrast, SCAFFOLD presents an intermediate cost (e.g. 0.1053 s at 200 nodes), driven by the additional aggregation of global control variates. Our approach records a slightly higher latency (0.1510 s) due to the execution of the clustering algorithm. However, this absolute cost remains negligible for practical deployment. Furthermore, the latency scales efficiently as the network grows. This stability indicates that the computational complexity of the density-based clustering remains low relative to the aggregation process,

ensuring that the addition of new clients does not create a computational bottleneck. For instance, the transition time for FedAvg increases by over 480% when scaling from 15 to 200 nodes. In contrast, our approach exhibits a modest increase of around 60% over the same range. This confirms that the overhead is largely fixed and robust to network expansion.

Crucially, the analysis of the upload payload reveals a significant advantage in bandwidth efficiency. SCAFFOLD requires doubling the data transmitted per client (approx. 165 KB/round) to exchange control variates. Conversely, our signature-based coordination incurs a marginal overhead of approximately 1.3% (≈ 84 KB/round) compared to FedAvg, regardless of network size.

Extending this analysis to total bandwidth consumption, Table 14 shows the total amount of data transferred during the entire training lifecycle for varying client sizes. The results confirm that the proposed architecture maintains a lean communication profile, achieving similar efficiency as lightweight baselines such as FedAvg. In contrast, SCAFFOLD's overhead increases drastically at scale. With 200 clients using the LR model, SCAFFOLD consumes over 203 MB, whereas our method consumes roughly 138 MB, an increase of almost 50%. This stems from the structural nature of the exchanged data. SCAFFOLD requires transmitting high-dimensional control variates alongside model parameters doubling the payload. Our approach only exchanges compact behavioural signatures, making it a far more scalable and bandwidth-efficient solution for resource-constrained IoT environments.

Finally, to validate the adaptive nature of the proposed architecture, Table 15 analyses the internal dynamics of the clustering mechanism. This evaluation uses the MLP model on Edge-IIoTset, under varying heterogeneity levels (α) and stability thresholds (τ). To interpret these results, we report four key metrics: *Peer Consistency*, which measures the stability of a client's neighbourhood regardless of cluster switching; *Average Noise*, indicating the percentage of outliers that are not assigned to families; *Client Persistence*, which tracks the probability of a client remaining in its mapped family across rounds; and *Perfect Clients*, the fraction of nodes that never switch clusters during the entire training process.

The results demonstrate a clear correlation between data distribution and structural behaviour. In IID-like scenarios (α=50), the system converges towards a highly stable configuration with minimal fragmentation (2–3 families). Notably, with a permissive threshold (τ=0.1), client persistence reaches 96.30%, with 66.67% of clients identified as "Perfect Clients", indicating that the algorithm locks onto the underlying homogeneity. Conversely, under extreme heterogeneity (α=0.1), the system automatically adapts by expanding to 5–6 families to isolate divergent behaviours. While this naturally reduces persistence, the stability threshold (τ) acts as an effective stabiliser. By reducing τ we force the algorithm to prioritise historical stability over immediate changes. For instance, at α=1, reducing the threshold from 0.5 to 0.1 increases peer consistency from 28.20% to 74.52% and extends the average client stay from 2.44 to 5.67 rounds.

5.4. Security and privacy implications

Threat model and structural resilience Our experimental evaluation follows standard benchmarking protocols for FL intrusion detection [58] and assumes a benign and honest environment. In this setting, all participants strictly adhere to the training protocol, in order to isolate the effects of statistical heterogeneity from external interference. However, the proposed architecture introduces structural defences against adversarial behaviours such as poisoning or Byzantine attacks [59].

The shift to predictive space clustering (from round $t \geq 2$) implies that poisoning attacks would alter the statistical behaviour of the model's predictions. This deviation results in a shift in the client's spectral signature, causing HDBSCAN to isolate the malicious actor as a micro-cluster or label it as noise [60]. This prevents legitimate *Family Experts* from being contaminated. Even if an anomaly infiltrates a family, the history-anchored aggregation rule (Eq. (4)) introduces inertia. Small per-round deviations are attenuated and substantial model drift requires either (i) sustained participation across many rounds and/or (ii) controlling a large fraction of clients. This condition ensures that malicious updates remain high-density and do not form a separate cluster [61]. Compared to high-magnitude model poisoning, the attack surface is reduced. Nevertheless, a massive and coordinated poisoning campaign (Sybil attack [62]) could form a high-density cluster that the algorithm misidentifies as a valid family. In such cases, specific defence mechanisms beyond unsupervised clustering would be required.

Privacy implications The transmission of signatures introduces a trade-off between personalisation and privacy. We mitigate operational data leakage by abstracting signatures into the predictive space, masking raw traffic attributes. Additionally, the aggregation strategy inherently protects against micro-targeting in small clusters by the aforementioned temporal inertia. The model weights mask individual contributions, even in the case where a cluster temporarily shrinks to a single participant. However, sophisticated Gradient Inversion Attacks (GIA) utilising auxiliary data could potentially exploit residual leakage from the shared family models. The integration of techniques such as Differential Privacy (DP) would help to mitigate these specific risks [63]. However, such measures introduce additional computational overhead and architectural complexity.

Operational stability and parameter sensitivity A common concern with density-based clustering is its sensitivity to hyperparameters, specifically *min_cluster_size*. However, in our architecture, this parameter controls the granularity of personalisation rather than system stability. In IID regimes, the algorithm consolidates participants into fewer, broader families, as observed in our results. In scenarios where the density structure cannot be resolved, the system automatically triggers a fallback to coarse K-Means++ clustering. This guarantees continuity even when optimal density-based personalisation is not feasible.

6. Conclusions and future work

This study demonstrates that integrating adaptive clustering into the federated learning process significantly enhances intrusion detection performance in heterogeneous IoT and IIoT environments. Across the three evaluated datasets, the proposed approach consistently outperforms FedAvg, FedProx, and FedAdam, achieving F1-score average improvements around 8-12% and maintaining over 90% of its peak performance even under high heterogeneity (α=0.1). These gains confirm that structured, behaviour-aware aggregation improves both convergence stability and detection reliability.

Furthermore, the method achieves this performance with only moderate computational cost. The runtime analysis shows that the additional clustering and adaptive aggregation introduce a variation of less than ±15% compared with standard schemes, and in several cases even reduce total execution time due to faster convergence. This balance between efficiency and robustness highlights the practical viability of the approach for real-world deployments. This is particularly important in large-scale or resource-constrained IoT systems, where communication and heterogeneity remain major challenges.

While the proposed architecture demonstrates strong adaptability and stability, it has certain limitations. The system still relies on a central coordinator for clustering and aggregation, and lacks advanced privacy mechanisms to protect shared model statistics. Furthermore, the issue of resilience against malicious or unreliable participants has not yet been explicitly addressed. The following future research paths are therefore aimed at improving the privacy, scalability and robustness of the federated process:

Privacy and security enhancement Although the current design preserves data locality, future research should incorporate stronger privacy-preserving mechanisms, such as DP or homomorphic encryption [64]. This would ensure the protection of model statistics during exchange, and prevent both inference and poisoning attacks.

Dynamic and decentralised coordination The architecture currently relies on a central coordinator for clustering and aggregation. Exploring decentralised or peer-to-peer variants [65] could remove this single point of failure and enhance resilience, particularly in ad-hoc or intermittently connected networks.

Adaptive optimisation and meta-learning The observed relationship between the Dirichlet parameter α and the stability factor suggests that adaptive or meta-learning strategies [66] could be introduced to tune clustering thresholds and family assignment automatically based on observed data dynamics.

Multimodal and temporal learning Extending the current framework to handle multimodal data [67], such as tabular, temporal, and signal-based features, could broaden its applicability to industrial IoT scenarios and improve robustness against complex or evolving attack patterns.

CRediT authorship contribution statement

Luis Miguel García-Sáez: Writing – review & editing, Writing – original draft, Visualization, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Sergio Ruiz-Villafranca:** Writing – review & editing, Validation, Supervision, Resources, Data curation, Conceptualization; **José Roldán-Gómez:** Writing – review & editing, Validation, Supervision, Resources, Methodology, Conceptualization; **Javier Carrillo-Mondéjar:** Writing – original draft, Visualization, Validation, Supervision, Resources; **José Luis Martínez:** Writing – review & editing, Validation, Supervision.

Data availability

Public datasets were used for this work

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work has been funded by the University of Castilla-La Mancha through the predoctoral 2024-UNIVERS-12844, supported by the European Social Fund Plus (ESF+), by the Regional Government of Castilla-La Mancha (JCCM) through the project SBPLY/21/180501/000195, and through the R&D project PID2024-158682OB-C32, funded by the MCIN and the European Regional Development Fund: “a way of making Europe”. This work has also been partially supported by PID2022-142332OA-I00, TED2021-131115A-I00, and PID2023-151467OA-I00, funded by MCIN/AEI/10.13039/501100011033, by the Recovery, Transformation and Resilience Plan funds, by the European Union (NextGenerationEU/PRTR), and the National Cybersecurity Institute of Spain (INCIBE). This work is also supported by the Department of University, Industry, and Innovation of the Government of Aragon under the Strategic Projects Program for Research Groups (DisCo research group, ref. T21-23R).

References

[1] J. Wang, M. K. Lim, C. Wang, M.-L. Tseng, The evolution of the internet of things (IoT) over the past 20 years, Computers Industr. Eng. 155 (2021) 107174. https://www.sciencedirect.com/science/article/pii/S0360835221000784. https://doi.org/10.1016/j.cie.2021.107174

[2] Y. B. Zikria, R. Ali, M. K. Afzal, S. W. Kim, Next-generation internet of things (IoT): opportunities, challenges, and solutions, Sensors 21 (4) (2021) 1174. Publisher: Multidisciplinary Digital Publishing Institute, https://doi.org/10.3390/s21041174

[3] P. K. Sadhu, V. P. Yanambaka, A. Abdelgawad, Internet of things: security and solutions survey, Sensors 22 (19) (2022) 7433. Publisher: Multidisciplinary Digital Publishing Institute, https://doi.org/10.3390/s22197433

[4] A. Srhir, T. Mazri, M. Benbrahim, Security in the IoT: state-of-the-art, issues, solutions, and challenges, Int. J. Adv. Computer Sci. Appl. (IJACSA) 14 (5) (2023). Publisher: The Science and Information (SAI) Organization Limited, https://thesai.org/Publications/ViewPaper?Volume=14&Issue=5&Code=IJACSA&SerialNo=7. https://doi.org/10.14569/IJACSA.2023.0140507

[5] A. Borys, A. Kamruzzaman, H. N. Thakur, J. C. Brickley, M. L. Ali, K. Thakur, An evaluation of IoT DDos cryptojacking malware and mirai botnet, in: 2022 IEEE World AI IoT Congress (AIIoT), 2022, pp. 725–729. https://doi.org/10.1109/AIIoT54504.2022.9817163

[6] E. Bout, V. Loscri, A. Gallais, Evolution of IoT security: the era of smart attacks, IEEE Internet of Things Magazine 5 (1) (2022) 108–113. https://ieeexplore.ieee.org/abstract/document/9773117. https://doi.org/10.1109/IOTM.001.2100183

[7] M. Schrötter, A. Niemann, B. Schnor, A comparison of neural-network-based intrusion detection against signature-based detection in IoT networks, Information 15 (3) (2024) 164. Publisher: Multidisciplinary Digital Publishing Institute, https://doi.org/10.3390/info15030164

[8] M. Aqeel, F. Ali, M. W. Iqbal, T. A. Rana, M. Arif, M. R. Auwul, et al., A review of security and privacy concerns in the internet of things (IoT), J. Sensors 2022 (1) (2022) 5724168. https://doi.org/10.1155/2022/5724168

[9] B. Liu, M. Ding, S. Shaham, W. Rahayu, F. Farokhi, Z. Lin, When machine learning meets privacy: a survey and outlook, ACM Comput. Surv. 54 (2) (2021) 31:1–31:36. https://doi.org/10.1145/3436755

[10] A. Samuel, G. N. Edegbe, A systematic review of centralized and decentralized machine learning models: security concerns, defenses and future directions, NIPES - Journal of Science and Technology Research 6 (4) (2024). https://journals.nipes.org/index.php/njstr/article/view/1047. https://doi.org/10.5281/zenodo.14681449

[11] S. Wang, L. L. Minku, X. Yao, A systematic study of online class imbalance learning with concept drift, IEEE Trans. Neural Netw. Learn. Syst. 29 (10) (2018) 4802–4821. https://ieeexplore.ieee.org/abstract/document/8246564. https://doi.org/10.1109/TNNLS.2017.2771290

[12] Z. Lu, H. Pan, Y. Dai, X. Si, Y. Zhang, Federated learning with non-IID data: a survey, IEEE Internet Things J. 11 (11) (2024) 19188–19209. https://ieeexplore.ieee.org/abstract/document/10468591. https://doi.org/10.1109/JIOT.2024.3376548

[13] J. Wen, Z. Zhang, Y. Lan, Z. Cui, J. Cai, W. Zhang, A survey on federated learning: challenges and applications, Int. J. Mach. Learn. Cybern. 14 (2) (2023) 513–535. https://doi.org/10.1007/s13042-022-01647-y

[14] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, H. Vincent Poor, Federated learning for internet of things: a comprehensive survey, IEEE Commun. Surv. Tutor. 23 (3) (2021) 1622–1658. https://ieeexplore.ieee.org/abstract/document/9415623. https://doi.org/10.1109/COMST.2021.3075439

[15] J. L. Hernandez-Ramos, G. Karopoulos, E. Chatzoglou, V. Kouliaridis, E. Marmol, A. Gonzalez-Vidal, G. Kambourakis, Intrusion detection based on federated learning: a systematic review, ACM Comput. Surv. 57 (12) (2025) 309:1–309:65. https://doi.org/10.1145/3731596

[16] M. Ye, X. Fang, B. Du, P. C. Yuen, D. Tao, Heterogeneous federated learning: state-of-the-art and research challenges, ACM Comput. Surv. 56 (3) (2023) 79:1–79:44. https://doi.org/10.1145/3625558

[17] J. Wang, Q. Liu, H. Liang, G. Joshi, H. V. Poor, Tackling the objective inconsistency problem in heterogeneous federated optimization, in: Advances in Neural Information Processing Systems, 33, Curran Associates, Inc., 2020, pp. 7611–7623. https://proceedings.neurips.cc/paper/2020/hash/564127c03caab942e503ee6f810f54fd-Abstract.html.

[18] T.-M. H. Hsu, H. Qi, M. Brown, Measuring the Effects of Non-Identical Data Distribution for Federated Visual Classification, 2019, https://doi.org/10.48550/arXiv.1909.06335

[19] Y. Mirsky, T. Doitshman, Y. Elovici, A. Shabtai, Kitsune: an ensemble of autoencoders for online network intrusion detection, in: Proceedings 2018 Network and Distributed System Security Symposium, Internet Society, San Diego, CA, 2018. https://www.ndss-symposium.org/wp-content/uploads/2018/02/ndss2018_03A-3_Mirsky_paper.pdf. https://doi.org/10.14722/ndss.2018.23204

[20] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, D. Breitenbacher, A. Shabtai, Y. Elovici, N-BaIoT: Network-based Detection of IoT Botnet Attacks Using Deep Autoencoders, IEEE Pervasive Comput. 17 (3) (2018) 12–22. http://arxiv.org/abs/1805.03409. https://doi.org/10.1109/MPRV.2018.03367731

[21] R. Lazzarini, H. Tianfield, V. Charissis, Federated learning for IoT intrusion detection, AI 4 (3) (2023) 509–530. Publisher: Multidisciplinary Digital Publishing Institute, https://doi.org/10.3390/ai4030028

[22] E. M. Campos, P. F. Saura, A. González-Vidal, J. L. Hernández-Ramos, J. B. Bernabé, G. Baldini, A. Skarmeta, Evaluating federated learning for intrusion detection in internet of things: review and challenges, Comput. Networks 203 (2022) 108661. https://www.sciencedirect.com/science/article/pii/S1389128621005405. https://doi.org/10.1016/j.comnet.2021.108661

[23] A. Khraisat, A. Alazab, M. Alazab, A. Obeidat, S. Singh, T. Jan, Federated learning for intrusion detection in IoT environments: a privacy-preserving strategy, Discover Internet of Things 5 (1) (2025) 72. https://doi.org/10.1007/s43926-025-00169-7

[24] A. Ghosh, J. Chung, D. Yin, K. Ramchandran, An efficient framework for clustered federated learning, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20, Curran Associates Inc., Red Hook, NY, USA, 2020, pp. 19586–19597. https://dl.acm.org/doi/10.5555/3495724.3497367.

[25] F. Sattler, K.-R. Müller, W. Samek, Clustered federated learning: model-agnostic distributed multitask optimization under privacy constraints, IEEE Trans. Neural Netw. Learn. Syst. 32 (8) (2021) 3710–3722. https://ieeexplore.ieee.org/document/9174890. https://doi.org/10.1109/TNNLS.2020.3015958

[26] E. S. Lubana, C. I. Tang, F. Kawsar, R. P. Dick, A. Mathur, Orchestra: Unsupervised Federated Learning via Globally Consistent Clustering, 2022, https://doi.org/10.48550/arXiv.2205.11506

[27] B. G. de Carvalho, L. A. P. Junior, A. L. d. Santos, O. Saotome, Generalizing intrusion detection for heterogeneous networks: A stacked-unsupervised federated learning approach, Computers Secur. 127 (2023) 103106. arXiv:2209.00721, https://doi.org/10.1016/j.cose.2023.103106

[28] T. Wei, B. Mei, J. Lyu, R. Zhang, F. Zhou, Y. Sun, Personalized Bayesian Federated Learning with Wasserstein Barycenter Aggregation, 2025, https://doi.org/10.48550/arXiv.2505.14161

[29] C. Qiu, Z. Wu, H. Wang, Q. Yang, Y. Wang, C. Su, Hierarchical aggregation for federated learning in heterogeneous IoT scenarios: enhancing privacy and communication efficiency, Future Internet 17 (1) (2025) 18. Publisher: Multidisciplinary Digital Publishing Institute, https://doi.org/10.3390/fi17010018

[30] H. Zhang, Z. Li, S. Xi, X. Zhao, J. Liu, P. Zhang, Heterogeneity-aware device selection for clustered federated learning in IoT, Peer-to-Peer Netw. Appl. 18 (1) (2024) 29. https://doi.org/10.1007/s12083-024-01869-7

[31] X. Zhou, X. Lei, C. Yang, Y. Shi, X. Zhang, J. Shi, Handling data heterogeneity for IoT devices in federated learning: a knowledge fusion approach, IEEE Internet Things J. 11 (5) (2024) 8090–8104. https://ieeexplore.ieee.org/document/10265259. https://doi.org/10.1109/JIOT.2023.3319986

[32] N. Ebrahimi, E. S. Soofi, S. A. Zhao, Information measures of Dirichlet distribution with applications, Appl. Stoch. Models Bus. Ind. 27 (2) (2011) 131–150. https://doi.org/10.1002/asmb.870

[33] H. Ratnayake, L. Chen, X. Ding, A review of federated learning: taxonomy, privacy and future directions, J. Intell. Inf. Syst. 61 (3) (2023) 923–949. https://doi.org/10.1007/s10844-023-00797-x

[34] D. Sculley, Web-scale k-means clustering, in: Proceedings of the 19th International Conference on World Wide Web, WWW '10, Association for Computing Machinery, New York, NY, USA, 2010, pp. 1177–1178. https://doi.org/10.1145/1772690.1772862

[35] A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc. Ser. B (Methodological) 39 (1) (1977) 1–22. https://doi.org/10.1111/j.2517-6161.1977.tb01600.x

[36] J. E. Gentle, Review of finding groups in data: an introduction to cluster analysis, Biometrics 47 (2) (1991) 788. Publisher: International Biometric Society, https://doi.org/10.2307/2532178

[37] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96, AAAI Press, Portland, Oregon, 1996, pp. 226–231.

[38] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics, 5.1, University of California Press, 1967, pp. 281–298. https://projecteuclid.org/ebooks/berkeley-symposium-on-mathematical-statistics-and-probability/Proceedings-of-the-Fifth-Berkeley-Symposium-on-Mathematical-Statistics-and/chapter/Some-methods-for-classification-and-analysis-of-multivariate-observations/bsmsp/1200512992.

[39] D. Arthur, S. Vassilvitskii, k-means++: the advantages of careful seeding, in: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '07, Society for Industrial and Applied Mathematics, USA, 2007, pp. 1027–1035.

[40] S. Firdaus, M. A. Uddin, A survey on clustering algorithms and complexity analysis, Int. J. Computer Sci. Issues (IJCSI) 12 (2) (2015) 62–85.

[41] R. J. G. B. Campello, D. Moulavi, J. Sander, Density-based clustering based on hierarchical density estimates, in: J. Pei, V. S. Tseng, L. Cao, H. Motoda, G. Xu (Eds.), Advances in Knowledge Discovery and Data Mining, Springer, Berlin, Heidelberg, 2013, pp. 160–172. https://doi.org/10.1007/978-3-642-37456-2_14

[42] A. A. Bushra, G. Yi, Comparative analysis review of pioneering DBSCAN and successive density-based clustering algorithms, IEEE Access 9 (2021) 87918–87935. https://ieeexplore.ieee.org/abstract/document/9453785. https://doi.org/10.1109/ACCESS.2021.3089036

[43] M. Fuchs, W. Höpken, Clustering, in: R. Egger (Ed.), Applied Data Science in Tourism: Interdisciplinary Approaches, Methodologies, and Applications, Springer International Publishing, Cham, 2022, pp. 129–149. https://doi.org/10.1007/978-3-030-88389-8_8

[44] M. Al-Hawawreh, E. Sitnikova, N. Aboutorab, X-IIoTID: a connectivity-agnostic and device-agnostic intrusion data set for industrial internet of things, IEEE Internet Things J. 9 (5) (2022) 3962–3977. https://ieeexplore.ieee.org/document/9504604. https://doi.org/10.1109/JIOT.2021.3102056

[45] B. S. Sharmila, R. Nagapadma, Quantized autoencoder (QAE) intrusion detection system for anomaly detection in resource-constrained IoT devices using RT-IoT2022 dataset, Cybersecurity 6 (1) (2023) 41. https://doi.org/10.1186/s42400-023-00178-5

[46] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, H. Janicke, Edge-IIoTset: a new comprehensive realistic cyber security dataset of IoT and IIot applications for centralized and federated learning, IEEE Access 10 (2022) 40281–40306. https://ieeexplore.ieee.org/document/9751703. https://doi.org/10.1109/ACCESS.2022.3165809

[47] N. Balakrishnan, V. B. Nevzorov, Dirichlet distribution, in: A Primer on Statistical Distributions, John Wiley & Sons, Ltd, 2003, pp. 269–276. https://doi.org/10.1002/0471722227.ch27

[48] S. Chatterjee, L. Pattnaik, S. Satpathy, P. K. Tripathy, Federated threat detection with Dirichlet distribution, in: A. Rocha, F. Moreira, S. N. Mohanty, S. Hu (Eds.), Integrating Advanced Technologies for Enhanced Security and Efficiency, Springer Nature Switzerland, Cham, 2025, pp. 63–77. https://doi.org/10.1007/978-3-031-91798-1_5

[49] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y. Arcas, Communication-efficient learning of deep networks from decentralized data, in: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, PMLR, 2017, pp. 1273–1282. ISSN: 2640-3498, https://proceedings.mlr.press/v54/mcmahan17a.html.

[50] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks, in: Proceedings of Machine Learning and Systems, 2, 2020, pp. 429–450. https://proceedings.mlsys.org/paper_files/paper/2020/hash/1f5fe83998a09396ebe6477d9475ba0c-Abstract.html.

[51] S. J. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, H. B. McMahan, Adaptive federated optimization, Vienna, Austria, 2021. https://openreview.net/forum?id=LkFG3lB13U5.

[52] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, A. T. Suresh, SCAFFOLD: Stochastic controlled averaging for federated learning, in: Proceedings of the 37th International Conference on Machine Learning, PMLR, 2020, pp. 5132–5143. ISSN: 2640–3498, https://proceedings.mlr.press/v119/karimireddy20a.html.

[53] M. M. Rahman, S. A. Shakil, M. R. Mustakim, A survey on intrusion detection system in IoT networks, Cyber Secur. Appl. 3 (2025) 100082. https://www.sciencedirect.com/science/article/pii/S2772918424000481. https://doi.org/10.1016/j.csa.2024.100082

[54] A. Momand, S. U. Jan, N. Ramzan, ABCNN-IDS: Attention-based convolutional neural network for intrusion detection in IoT networks, Wireless Personal Commun. 136 (4) (2024) 1981–2003. https://doi.org/10.1007/s11277-024-11260-7

[55] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É. Duchesnay, Scikit-learn: machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830. https://dl.acm.org/doi/10.5555/1953048.2078195.

[56] G. D. Pecherle, R. Ş. Győrödi, C. A. Győrödi, Federated learning-based intrusion detection in industrial IoT networks, Future Internet 18 (1) (2026) 2. https://www.mdpi.com/1999-5903/18/1/2. https://doi.org/10.3390/fi18010002

[57] A. Imteaj, U. Thakker, S. Wang, J. Li, M. H. Amini, A survey on federated learning for resource-constrained IoT devices, IEEE Internet Things J. 9 (1) (2022) 1–24. https://ieeexplore.ieee.org/abstract/document/9475501/. https://doi.org/10.1109/JIOT.2021.3095077

[58] S. Chennoufi, Y. Han, G. Blanc, E. De Cristofaro, C. Kiennert, PROTEAN: federated intrusion detection in non-IID environments through prototype-based knowledge sharing, in: V. Nicomette, A. Benzekri, N. Boulahia-Cuppens, J. Vaidya (Eds.), Computer Security - ESORICS 2025, Springer Nature Switzerland, Cham, 2026, pp. 103–125. https://doi.org/10.1007/978-3-032-07884-1_6

[59] M. Fang, X. Cao, J. Jia, N. Z. Gong, Local model poisoning attacks to byzantine-robust federated learning, in: Proceedings of the 29th USENIX Conference on Security Symposium, SEC'20, USENIX Association, USA, 2020, pp. 1623–1640. https://dl.acm.org/doi/10.5555/3489212.3489304.

[60] T. D. Nguyen, P. Rieger, H. Chen, H. Yalame, H. Möllering, H. Fereidooni, S. Marchal, M. Miettinen, A. Mirhoseini, S. Zeitouni, F. Koushanfar, A.-R. Sadeghi, T. Schneider, FLAME: Taming backdoors in federated learning, in: Proceedings of the 31th USENIX Conference on Security Symposium (USENIX Security 22), SEC'22, 2022, pp. 1415–1432. https://www.usenix.org/conference/usenixsecurity22/presentation/nguyen?utm_source=chatgpt.com.

[61] A. N. Bhagoji, S. Chakraborty, P. Mittal, S. Calo, Analyzing federated learning through an adversarial lens, in: Proceedings of the 36th International Conference on Machine Learning, PMLR, 2019, pp. 634–643. https://proceedings.mlr.press/v97/bhagoji19a.html.

[62] C. Fung, C. J. M. Yoon, I. Beschastnikh, The limitations of federated learning in sybil settings, 2020, pp. 301–316. https://www.usenix.org/conference/raid2020/presentation/fung.

[63] G. K. Jagarlamudi, A. Yazdinejad, R. M. Parizi, S. Pouriyeh, Exploring privacy measurement in federated learning, J. Supercomput. 80 (8) (2024) 10511–10551. https://doi.org/10.1007/s11227-023-05846-4

[64] R. Aziz, S. Banerjee, S. Bouzefrane, T. Le Vinh, Exploring homomorphic encryption and differential privacy techniques towards secure federated learning paradigm, Future Internet 15 (9) (2023) 310. Publisher: Multidisciplinary Digital Publishing Institute, https://doi.org/10.3390/fi15090310

[65] L. Yuan, Z. Wang, L. Sun, P. S. Yu, C. G. Brinton, Decentralized federated learning: a survey and perspective, IEEE Internet Things J. 11 (21) (2024) 34617–34638. https://ieeexplore.ieee.org/document/10542323. https://doi.org/10.1109/JIOT.2024.3407584

[66] X. Liu, Y. Deng, A. Nallanathan, M. Bennis, Federated learning and meta learning: approaches, applications, and directions, IEEE Commun. Surv. Tutor. 26 (1) (2024) 571–618. https://ieeexplore.ieee.org/document/10310213. https://doi.org/10.1109/COMST.2023.3330910

[67] W. Huang, D. Wang, X. Ouyang, J. Wan, J. Liu, T. Li, Multimodal federated learning: concept, methods, applications and future directions, Inf. Fusion 112 (2024) 102576. https://www.sciencedirect.com/science/article/pii/S1566253524003543. https://doi.org/10.1016/j.inffus.2024.102576