Feature Skew Control for In-Network Federated Learning

Yasmine Chaouche, Patient Ntumba, Stefano Secci, Pedro B. Velloso
Cnam, Paris, France. firstname.lastname@cnam.fr

***Abstract*—Applying Federated Learning (FL) to real-world network environments, such as telecommunication networks, presents significant challenges. In this paper, we investigate the intrinsic problem of non-IID data training across distributed in-network FL clients. More specifically, we focus on the feature distribution discrepancy in the use of in-network FL for infrastructure monitoring. While previous works have made notable progress in the design of aggregation functions that compensate strong polarization in data distributions, limited attention was directed toward data load-balancing from sources to processing nodes, which leverages the role and characteristics of the data itself. Our goal is to address feature heterogeneity in in-network federated learning by piloting how data is forwarded in the network on the way to its consumers. To this end, we design and evaluate several scenarios for dynamic data load balancing between FL clients within the network topology, subject to latency constraints.**

***Index Terms*—In-Network Federated Learning, Feature skew, Anomaly Detection.**

I. Introduction

Nowadays, a large volume of data is continuously generated at the network edge due to the widespread usage of mobile devices and development of pervasive sensing applications [1]. Moreover, the penetration of computing and virtualization within network edges requires in-depth and large-scale monitoring of infrastructure components. Typically, this massive amount of data is aggregated and stored ‘in the cloud’ alongside the installment of native machine learning runtimes at the edge cloud operating systems. In this context, Artificial Intelligence Functions (AIFs) are adopted at the edge cloud and provider premises as AI-powered components of end-to-end infrastructure and vertical applications. They are designed for deployment across edge-enabled 5G and upcoming Beyond 5G [2] systems to offload, or even replace, legacy remote cloud deployments. This trend goes beyond basic centralized learning, to enable the deployment of in-network AI services at scale [3]; indeed, transmitting a huge volume of sensitive data to remote clouds over the network raises significant privacy and network efficiency concerns: privacy ones gave rise to Federated Learning (FL) [4], while efficiency concerns are leading to the integration of FL as an in-network AI service, for the analysis of infrastructure data instead of end-device data.

Unlike centralized learning, FL enables the collaborative training of a global model across distributed nodes using local data. Specifically, only the parameters of a locally trained model are sent to a central server for aggregation. The global model is then updated and redistributed to the local nodes for the next iteration. Thus, the global model is trained without transmitting local data and compromising user privacy. Conventional FL applications, such as those in computer vision and natural language processing, often assume that training data is Independent and Identically Distributed (IID) across local nodes or clients. However, in real-world mobile access network scenarios, local training data are often non-IID across distributed clients. This means that the data distribution on each node is unique and does not reflect the overall global distribution. Such discrepancy is typically caused by statistical heterogeneity, where the local data distributions differ among participants [5], or by system heterogeneity, which refers to variations in system parameters (e.g., the operating frequency of end devices). In such cases, a naive implementation of FL algorithms (e.g. FedAvg) leads to a degradation of model performance [6], [7]. This is particularly relevant in anomaly detection, as each node may get different types of normal data hence of anomalies.

Recently, in-network federated learning has emerged to analyze computing and communication infrastructure. The main idea is to support collaborative model training directly within the infrastructure [2]. In this perspective, Data Pipeline Systems (DPS) are proposed as an evolution of conventional datawarehouse systems to bring data from sources to AIFs [8]. Hence, AIFs use real-time infrastructure data to enable advanced applications such as anomaly detection, mobility prediction, and event forecasting, thereby supporting infrastructure orchestration as well as the identification of vulnerabilities and software failures. These capabilities are particularly important for network operators to meet Service Level Agreements (SLAs) requirements. However, a key challenge is the inherent heterogeneity of data across distributed sources, which can degrade the performance of FL models. Additionally, in-network FL with DPS support has to master the communication cost, both for model updates between FL clients and the server, and for the transfer of data from potentially large number of sources to FL clients via the DPS. This is especially critical in telecommunications, where bandwidth is limited, links can be unstable, latency requirements are strict, and SLAs must be ensured.

In this work, we propose DPS-based data load balancing policies to mitigate feature skew for in-network FL. The DPS aggregates data from distributed network nodes and routes it to AIF clients for local training. Our objective is to investigate their impact on model performance while also accounting for data communication costs. Preliminary evaluation results show

978-3-903176-75-1 ©2025 IFIP

how data load balancing scenarios affect data distributions among participants.

The remainder of the paper is organized as follows: Section II reviews related work. Section III defines the problem statement and Section IV presents our core contribution. Section V reports the evaluation methodology and the initial experimental results. Finally, we conclude this paper and direct our future work in Section VI.

II. Related Work

According to [9], there are three main types of 'non-IIDness': (i) Label skew, where labels differ among clients; (ii) Feature skew, where attribute values differ among clients; and (iii) Quantity skew (i.e. imbalanced data), where the amount of data varies among clients. Current practices [4], [6], [7], [10]–[12] offer limited strategies for distributing data between FL clients, where data heterogeneity is modeled as non-IID label skew. They are primarily designed for classification tasks, and their effectiveness in addressing feature distribution skew has not been explored. For instance, for autoencoder-based anomaly detection systems considered in this study, where models are trained only on normal data, feature distribution issues become more critical.

In this paper, we consider the in-network FL context, where a dedicated Data Pipeline System (DPS) transfers data from distributed sources to FL client AIFs [8], and where these AIFs are executed within the network infrastructure itself. In this context, the issue of data heterogeneity is exacerbated due to intrinsic geographical and topological characteristics of the network between data-sources and FL clients [8]. As a consequence, if left unmanaged, the resulting data-to-client distribution policy can significantly reduce the degree of IIDness, thereby degrading overall FL performance. The authors in [13] introduce a FL framework for Anomaly Detection (AD), named FLAD, which uses multiple edge AIFs to train autoencoders and share model parameters with a central AIF server. However, the data is split using a sequential partitioning strategy, where each AIF receives a block of consecutive samples. This results in a data distribution that fails to meet the IID assumption across FL clients, which is essential for maximizing FL efficiency.

III. In-network FL deployment with Feature Skew Control

Figure 1(a) depicts the reference operator network architecture used for the in-network federated learning use-case. It comprises a set of base stations (BS), core network nodes (CN), and border nodes (BN). A BN is typically deployed in close proximity to a CN node to enable low-latency content delivery [8]. In this architecture, each CN is connected to a subset of BSs and is associated to a corresponding BN. We consider two types of network delays in this setup: (i) *front-hauling delay* fd_{ij}, which represents the propagation delay between a BS_i and a CN_j; and (ii) *back-hauling delay* bd_{jk}, which denotes the propagation delay between a CN_j and its associated BN_k. In in-network FL [2], [8], the AIF server and a set of AIF clients $C = \{AIF_1, \ldots, AIF_c\}$ are deployed at the BN. A set of data sources $S = \{s_1, \ldots, s_{|S|}\}$ is located at the BS level. The data generated by these sources is forwarded to the AIF clients through the DPS, represented by a set of brokers B which are executed within the CN nodes. Figure 1(b) illustrates this setup.

A. Towards Feature Skew Control

We consider an in-network FL setup in which each AIF client c performs local training on received data and sends model updates to a centralized AIF server. The server synchronously aggregates the local model parameters and redistributes the resulting global model to all participating AIF clients.

In conventional FL [4], local models are trained independently on individual client devices to preserve data locality, which is the case in practical real-world systems, due to constraints arising from data sensitivity or limited network connectivity that prevents data transfer. We adopt a similar decentralized learning paradigm within the operator network, where each AIF client $c \in C$ receives data from a unique subset of data sources $S_b \subset S$ through a broker $b \in B$, forming a many-to-one and one-to-one mapping: $S_b \to b \to c$. Since each AIF client c operates at the border of the CN, we assume negligible network delay between data broker and their corresponding AIF clients.

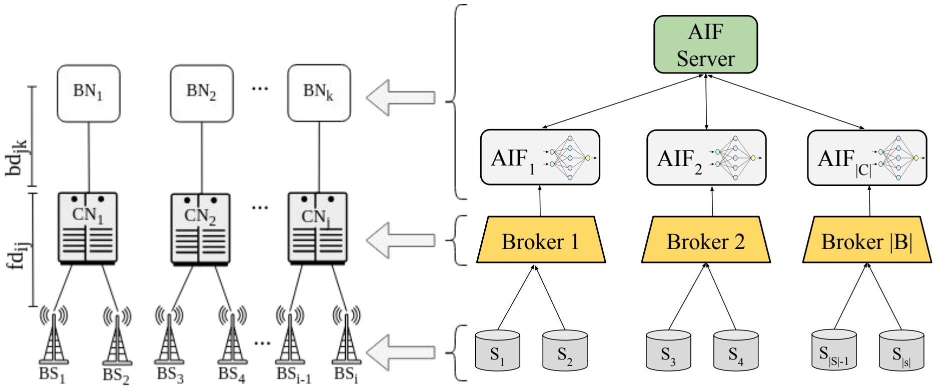
While the conventional FL scenario outlined above preserves privacy and enables decentralized model training, it encounters fundamental limitations when dealing with non-IID data across AIF clients. In practice, data sources are often highly heterogeneous, due to different user behaviors or variable hardware characteristics. As a result, each AIF receives a uniquely skewed subset of the global data distribution, which can lead to local model updates that may diverge significantly.

To address this challenge, we propose a methodological shift from simply assuming strict data locality to a more clever controlled data sharing via strategic data routing, as permitted by DPS brokers load-balancing. Unlike in conventional FL, where each broker $b \in B$ serves a single AIF client $c \in C$, we propose a broker-assisted data load balancing scheme in which each broker can forward data to multiple AIF clients forming a subset $C_b \subseteq C$, which is feasible thanks to publish-subscribe model of the DPS. This flexible mapping $S_b \to b \to C_b$ allows brokers to implement dynamic routing policies, selectively dispatching data to AIF clients to mitigate data imbalance and reshape local feature distributions.

To enhance model generalization and stability, our objective is to reduce inter-client heterogeneity by approximating an IID distribution across AIF clients. Achieving this through broker-assisted data distribution introduces a fundamental trade-off between statistical benefits and network communication costs.

IV. Load-Balancing for Feature Skew Control

To mitigate non-IID limitations in federated learning, we introduce broker-assisted data load balancing policies, as presented in Figure 2. The core idea is to use brokers to uniformly



(a) Operator Network architecture (b) Conventional Federated Learning

Fig. 1: In-network Federated Learning

redistribute data across AIF clients, which mitigates data skew and enables balanced training. We propose two variants: the *Fully Connected Scenario* for maximum uniformity, and the *Least Broker-AIF Delay Scenario* to reduce communication costs by selectively limiting broker-to-AIF connectivity.

In the *Fully Connected scenario*, each broker $b \in B$ maintains connections to all AIF clients. As such, data from the subset $S_b \in S$ assigned to the broker $b \in B$ are uniformly distributed over all AIF clients. This results in a data allocation that approximates an IID setting. For the *Least Broker-AIF Delay Scenario*, the communication cost of the Fully Connected setup is reduced while still approximating IID distributions. Each broker $b \in B$ deactivates a proportion (ratio) $r \in]0, 1]$ of its highest-latency links to AIF clients, as illustrated in Figure 2. Data from each source assigned to broker b is then uniformly distributed over its reduced client set. Although this may slightly deviate from a perfect IID setup, it significantly reduces communication cost and better suits large-scale or resource-constrained deployments.

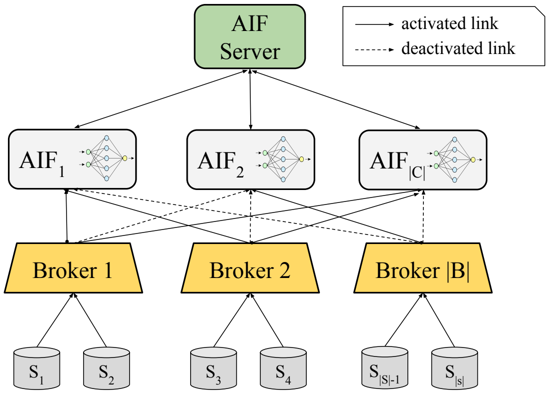


Fig. 2: Least Broker-AIF Delay Scenario

A. Communication Cost Model

Based on empirical delay traces from a major European provider [8], the data transmitted from the base stations to the CN nodes, where the brokers are located, experience a negligible and quasi-constant *front-hauling delay* compared to the *back-hauling delay*. Therefore, we exclude the *front-hauling delay* from our communication cost analysis. Instead, we focus on the network link between the core nodes and the border nodes, where the brokers and AIF clients are located, respectively.

We evaluate the communication cost conventionally as a bandwidth-delay-product, to represent the amount of data in-flight on the way to AIFs. The required bitrate β (in Mbps) for transmitting a single data sample is calculated as the product of the sample size and the sampling frequency of the data source. In practice, data samples are typically represented using a 32-bit floating-point format [14]. The cost is computed as follows:

$$\text{cost} = \beta \cdot \left(\sum_{b,c} a_{bc} \cdot bd_{jk} \right) \tag{1}$$

where a_{bc} is the number of data samples transmitted from broker $b \in B$ to AIF client $c \in C$, and bd_{jk} represents the *back-hauling delay* between core network node CN_j and border node BN_k, where broker b and AIF client c are respectively deployed.

B. Quantifying Non-IID Data Distributions

Quantifying the degree of non-IIDness is essential to address the issue of feature distribution heterogeneity, as it provides insights into the diversity of data distributions across clients. In the literature, several approaches have been proposed to align probability distributions, such as using statistical distance measures like the Wasserstein distance and the Kullback-Leibler Divergence (KLD) [15]. In this work, we adopt the Jensen-Shannon Divergence (JSD), which is a symmetrized and smoothed version of KLD. The JSD between two probability distributions P and Q is defined as [16]:

$$\text{JSD}(P, Q) = \frac{1}{2}\text{KLD}(P \parallel M) + \frac{1}{2}\text{KLD}(Q \parallel M) \tag{2}$$

where $M = \frac{1}{2}(P + Q)$ is the mixture distribution of P and Q. The Kullback-Leibler Divergence [17], denoted

$\mathrm{KLD}(P \parallel Q)$, between two discrete probability distributions over a shared space Z, is defined as: $\mathrm{KLD}(P \parallel Q) = \sum_{z\in Z} P(z)\log\left(P(z)/Q(z)\right)$.

A lower JSD value indicates a higher similarity between two probability distributions. In this work, for each feature f and AIF client $c \in C$, we estimate the empirical probability distributions of the feature values denoted P_{cf} for AIF client c and P_f for the global dataset using normalized histograms. We then compute the JSD between these distributions as $\mathrm{JSD}(P_{cj} \parallel P_f)$. To obtain a feature-wise measure of how much the distribution of each feature differs between clients and the global distribution, we compute the average JSD across all $|C|$ clients: $\bar{D}_f = \frac{1}{|C|}\sum_{c=1}^{|C|} \mathrm{JSD}(P_{cf} \parallel P_f)$.

V. Evaluation

To evaluate our proposal, we use the in-network FL environment from [8]. We consider 5 AIF clients and 5 broker nodes, whose placement in the operator network follows the deployment presented in Figure 1. For the evaluation, we define five values for the ratio of link deactivation per broker: 0%, 20%, 40%, 60%, and 80%, in which the first value corresponds to the Fully Connected scenario. Finally, we present preliminary experimental results and analyze the impact of our proposed scenario on feature heterogeneity.

A. Dataset

To evaluate our proposed solution, we use the publicly available CIC-IDS-2017 dataset [18]. It was collected in a network environment with simulated traffic to evaluate realistic intrusion detection scenarios. The dataset was generated using a B-profile system designed to replicate background traffic and capture the abstract behavior of 25 users across multiple protocols, including HTTP, HTTPS, FTP, SSH, and email. To simulate malicious activity, six attack profiles were incorporated (e.g., DDoS, SQL Injection, Brute Force, Botnet). Data collection was conducted over five consecutive days, with Monday dedicated exclusively to benign traffic and the remaining days containing a mix of normal and attack traffic.

B. Data Preprocessing

Let $\mathcal{D} \in \mathbb{R}^{n\times d}$ denote the dataset, where n is the number of samples and $|F|$ is the number of features. In the DPS data preprocessing function, each feature $f \in F$ is scaled to the range $[0,1]$ using min-max normalization, resulting in the normalized dataset $\tilde{\mathcal{D}} \in \mathbb{R}^{n\times |F|}$.

To reduce dimensionality and improve the quality of feature distribution analysis, we apply Principal Component Analysis (PCA) to both training and test sets prior to feature splitting among participants. This transformation projects the data into a lower-dimensional subspace defined by the most informative directions of variance: $\mathbf{Z} = \tilde{\mathcal{D}} \cdot \mathbf{W}$, where $\mathbf{W} \in \mathbb{R}^{|F|\times k}$ contains the top k principal components, and $\mathbf{Z} \in \mathbb{R}^{n\times k}$ is the resulting reduced representation. This step ensures that only the most informative components of the original feature space are retained, which is particularly important given the scarcity of naturally non-IID tabular datasets for anomaly detection. In our experiments, PCA retained 97.40% of the original variance for the CIC-IDS-2017 dataset.

C. Building Non-IID Data Setting

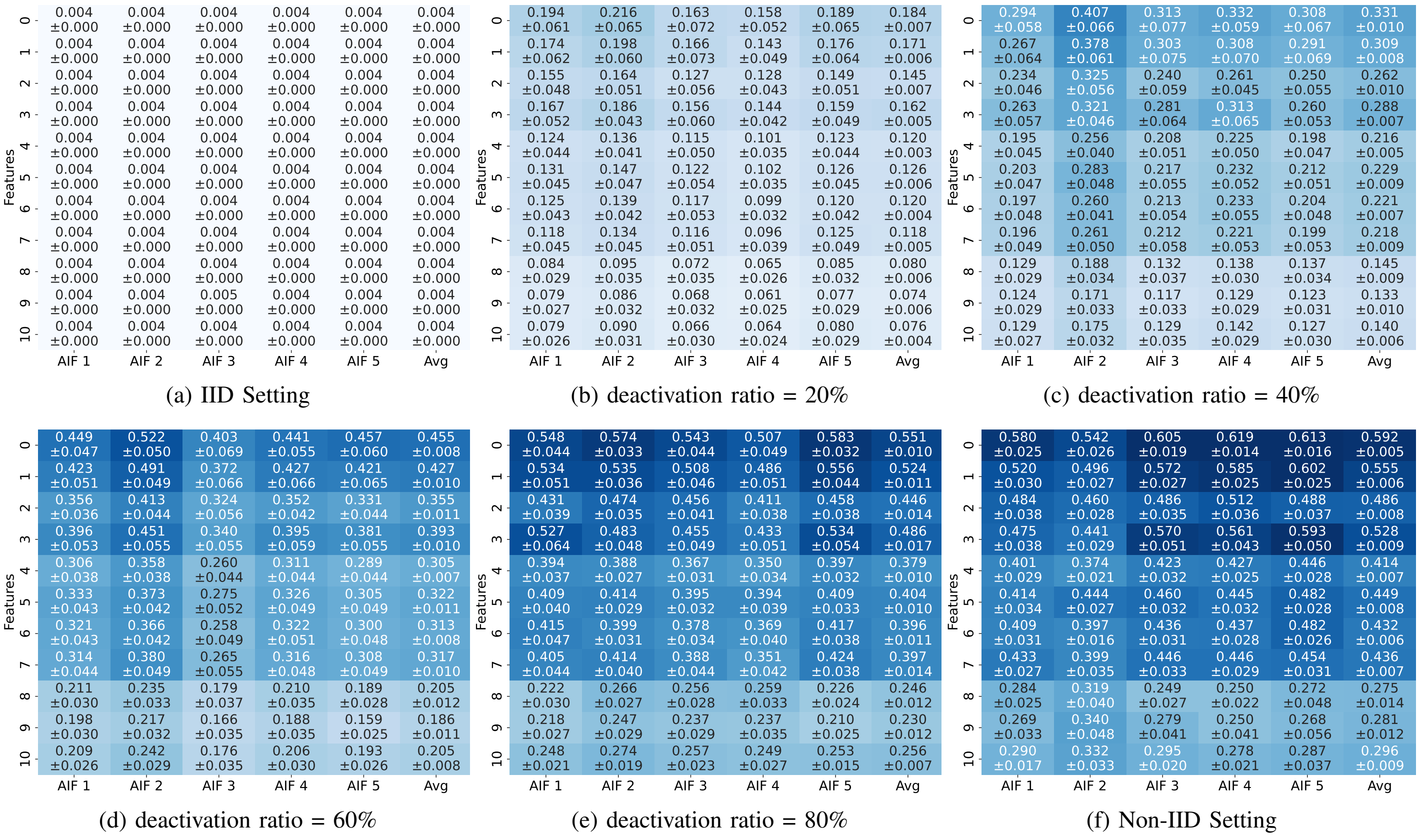
To stress a non-IID feature skew, we adopt a synthetic partitioning strategy based on feature separation at the data source level. This design choice allows us to highlight the impact of non-IIDness on the model performance. To this end, we apply unsupervised clustering on the global training data using the K-Means algorithm, which groups samples based on their similarity in the feature space. Each subset of data sources, illustrated in Figure 1, is then assigned data samples exclusively from a single cluster, thereby creating feature-based heterogeneity across base stations. This methodology enables a controlled yet realistic evaluation of the robustness and adaptability of FL paradigms under heterogeneous conditions, which allows us to evaluate the scenarios we are proposing in this study.

D. Preliminary Results

Figure 3 shows the JSD values computed for each feature between the distribution of individual clients and the global distribution, as defined in Section IV-B, for the CIC-IDS-2017 dataset. The last column in each subfigure represents the average JSD per feature across all clients, which serves as an aggregate indicator of the overall distributional heterogeneity. The results show a clear pattern: increasing the deactivation ratio of broker-to-AIF links leads to higher levels of statistical heterogeneity. In the Fully Connected configuration, JSD values are close to zero, indicating minimal deviation from the global distribution and thus well-balanced feature allocation. As the deactivation ratio rises (e.g., 40%, 60%, and 80%), feature divergence becomes more pronounced. The highest heterogeneity is observed in the conventional FL setup, where no data balancing is applied. These results confirm the effectiveness of network data load balancing policies to reduce feature heterogeneity.

VI. Conclusion

In this work, we propose a data pipelining load-balancing technique with feature-skew control for in-network federated learning: when data sources and processing nodes are not co-located but linked by a network. We propose to load balance data source features to processing nodes so as to control data heterogeneity and thereby improve AIF accuracy. We explore two variants: (1) Fully Connected, which ensures statistical uniformity but increases communication overhead, and (2) Least Broker-AIF Delay, which reduces network load by selectively limiting broker-client links. Initial results demonstrate the effectiveness of our strategy in reducing feature heterogeneity. Future work will investigate the impact of our solution on model performance and communication costs, with a particular focus on characterizing the trade-off between them.



(a) IID Setting

Features	AIF 1	AIF 2	AIF 3	AIF 4	AIF 5	Avg
0	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000
1	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000
2	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000
3	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000
4	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000
5	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000
6	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000
7	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000
8	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000
9	0.004 ±0.000	0.004 ±0.000	0.005 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000
10	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000	0.004 ±0.000

(b) deactivation ratio = 20%

Features	AIF 1	AIF 2	AIF 3	AIF 4	AIF 5	Avg
0	0.194 ±0.061	0.216 ±0.065	0.163 ±0.072	0.158 ±0.052	0.189 ±0.065	0.184 ±0.007
1	0.174 ±0.062	0.198 ±0.060	0.166 ±0.073	0.143 ±0.049	0.176 ±0.064	0.171 ±0.006
2	0.155 ±0.048	0.164 ±0.051	0.127 ±0.056	0.128 ±0.043	0.149 ±0.051	0.145 ±0.007
3	0.167 ±0.052	0.186 ±0.043	0.156 ±0.060	0.144 ±0.042	0.159 ±0.049	0.162 ±0.005
4	0.124 ±0.044	0.136 ±0.041	0.115 ±0.050	0.101 ±0.035	0.123 ±0.044	0.120 ±0.003
5	0.131 ±0.045	0.147 ±0.047	0.122 ±0.054	0.102 ±0.035	0.126 ±0.045	0.126 ±0.006
6	0.125 ±0.043	0.139 ±0.042	0.117 ±0.053	0.099 ±0.032	0.120 ±0.042	0.120 ±0.004
7	0.118 ±0.045	0.134 ±0.045	0.116 ±0.051	0.096 ±0.039	0.125 ±0.049	0.118 ±0.005
8	0.084 ±0.029	0.095 ±0.035	0.072 ±0.035	0.065 ±0.026	0.085 ±0.032	0.080 ±0.006
9	0.079 ±0.027	0.086 ±0.032	0.068 ±0.032	0.061 ±0.025	0.077 ±0.029	0.074 ±0.006
10	0.079 ±0.026	0.090 ±0.031	0.066 ±0.030	0.064 ±0.024	0.080 ±0.029	0.076 ±0.004

(c) deactivation ratio = 40%

Features	AIF 1	AIF 2	AIF 3	AIF 4	AIF 5	Avg
0	0.294 ±0.058	0.407 ±0.066	0.313 ±0.077	0.332 ±0.059	0.308 ±0.067	0.331 ±0.010
1	0.267 ±0.064	0.378 ±0.061	0.303 ±0.075	0.308 ±0.070	0.291 ±0.069	0.309 ±0.008
2	0.234 ±0.046	0.325 ±0.056	0.240 ±0.059	0.261 ±0.045	0.250 ±0.055	0.262 ±0.010
3	0.263 ±0.057	0.321 ±0.046	0.281 ±0.064	0.313 ±0.065	0.260 ±0.053	0.288 ±0.007
4	0.195 ±0.045	0.256 ±0.040	0.208 ±0.051	0.225 ±0.050	0.198 ±0.047	0.216 ±0.005
5	0.203 ±0.047	0.283 ±0.048	0.217 ±0.055	0.232 ±0.052	0.212 ±0.051	0.229 ±0.009
6	0.197 ±0.048	0.260 ±0.041	0.213 ±0.054	0.233 ±0.055	0.204 ±0.048	0.221 ±0.007
7	0.196 ±0.049	0.261 ±0.050	0.212 ±0.058	0.221 ±0.053	0.199 ±0.053	0.218 ±0.009
8	0.129 ±0.029	0.188 ±0.034	0.132 ±0.037	0.138 ±0.030	0.137 ±0.034	0.145 ±0.009
9	0.124 ±0.029	0.171 ±0.033	0.117 ±0.033	0.129 ±0.029	0.123 ±0.031	0.133 ±0.010
10	0.129 ±0.027	0.175 ±0.032	0.129 ±0.035	0.142 ±0.029	0.127 ±0.030	0.140 ±0.006

(d) deactivation ratio = 60%

Features	AIF 1	AIF 2	AIF 3	AIF 4	AIF 5	Avg
0	0.449 ±0.047	0.522 ±0.050	0.403 ±0.069	0.441 ±0.055	0.457 ±0.060	0.455 ±0.008
1	0.423 ±0.051	0.491 ±0.049	0.372 ±0.066	0.427 ±0.066	0.421 ±0.065	0.427 ±0.010
2	0.356 ±0.036	0.413 ±0.044	0.324 ±0.056	0.352 ±0.042	0.331 ±0.044	0.355 ±0.011
3	0.396 ±0.053	0.451 ±0.055	0.340 ±0.055	0.395 ±0.059	0.381 ±0.055	0.393 ±0.010
4	0.306 ±0.038	0.358 ±0.038	0.260 ±0.044	0.311 ±0.044	0.289 ±0.044	0.305 ±0.007
5	0.333 ±0.043	0.373 ±0.042	0.275 ±0.052	0.326 ±0.049	0.305 ±0.049	0.322 ±0.011
6	0.321 ±0.043	0.366 ±0.042	0.258 ±0.049	0.322 ±0.051	0.300 ±0.048	0.313 ±0.008
7	0.314 ±0.044	0.380 ±0.049	0.265 ±0.055	0.316 ±0.048	0.308 ±0.049	0.317 ±0.010
8	0.211 ±0.030	0.235 ±0.033	0.179 ±0.037	0.210 ±0.035	0.189 ±0.028	0.205 ±0.012
9	0.198 ±0.030	0.217 ±0.032	0.166 ±0.035	0.188 ±0.035	0.159 ±0.025	0.186 ±0.011
10	0.209 ±0.026	0.242 ±0.029	0.176 ±0.035	0.206 ±0.030	0.193 ±0.026	0.205 ±0.008

(e) deactivation ratio = 80%

Features	AIF 1	AIF 2	AIF 3	AIF 4	AIF 5	Avg
0	0.548 ±0.044	0.574 ±0.033	0.543 ±0.044	0.507 ±0.049	0.583 ±0.032	0.551 ±0.010
1	0.534 ±0.051	0.535 ±0.036	0.508 ±0.046	0.486 ±0.051	0.556 ±0.044	0.524 ±0.011
2	0.431 ±0.039	0.474 ±0.035	0.456 ±0.041	0.411 ±0.038	0.458 ±0.038	0.446 ±0.014
3	0.527 ±0.064	0.483 ±0.048	0.455 ±0.049	0.433 ±0.051	0.534 ±0.054	0.486 ±0.017
4	0.394 ±0.037	0.388 ±0.027	0.367 ±0.031	0.350 ±0.034	0.397 ±0.032	0.379 ±0.010
5	0.409 ±0.040	0.414 ±0.029	0.395 ±0.032	0.394 ±0.039	0.409 ±0.033	0.404 ±0.010
6	0.415 ±0.047	0.399 ±0.031	0.378 ±0.034	0.369 ±0.040	0.417 ±0.038	0.396 ±0.011
7	0.405 ±0.044	0.414 ±0.040	0.388 ±0.044	0.351 ±0.042	0.424 ±0.038	0.397 ±0.014
8	0.222 ±0.030	0.266 ±0.027	0.256 ±0.028	0.259 ±0.033	0.226 ±0.024	0.246 ±0.012
9	0.218 ±0.027	0.247 ±0.029	0.237 ±0.029	0.237 ±0.035	0.210 ±0.025	0.230 ±0.012
10	0.248 ±0.021	0.274 ±0.019	0.257 ±0.023	0.249 ±0.027	0.253 ±0.015	0.256 ±0.007

(f) Non-IID Setting

Features	AIF 1	AIF 2	AIF 3	AIF 4	AIF 5	Avg
0	0.580 ±0.025	0.542 ±0.026	0.605 ±0.019	0.619 ±0.014	0.613 ±0.016	0.592 ±0.005
1	0.520 ±0.030	0.496 ±0.027	0.572 ±0.027	0.585 ±0.025	0.602 ±0.025	0.555 ±0.006
2	0.484 ±0.038	0.460 ±0.028	0.486 ±0.035	0.512 ±0.036	0.488 ±0.037	0.486 ±0.008
3	0.475 ±0.038	0.441 ±0.029	0.570 ±0.051	0.561 ±0.043	0.593 ±0.050	0.528 ±0.009
4	0.401 ±0.029	0.374 ±0.021	0.423 ±0.032	0.427 ±0.025	0.446 ±0.028	0.414 ±0.007
5	0.414 ±0.034	0.444 ±0.027	0.460 ±0.032	0.445 ±0.032	0.482 ±0.028	0.449 ±0.008
6	0.409 ±0.031	0.397 ±0.016	0.436 ±0.031	0.437 ±0.028	0.482 ±0.026	0.432 ±0.006
7	0.433 ±0.027	0.399 ±0.035	0.446 ±0.033	0.446 ±0.029	0.454 ±0.031	0.436 ±0.007
8	0.284 ±0.025	0.319 ±0.040	0.249 ±0.027	0.250 ±0.022	0.272 ±0.048	0.275 ±0.014
9	0.269 ±0.033	0.340 ±0.048	0.279 ±0.041	0.250 ±0.041	0.268 ±0.056	0.281 ±0.012
10	0.290 ±0.017	0.332 ±0.033	0.295 ±0.020	0.278 ±0.021	0.287 ±0.037	0.296 ±0.009

Fig. 3: Jensen-Shannon Divergence (mean ± 95% CI) across varying levels of client heterogeneity.

Acknowledgment

This work was funded by the CoCo5G (ANR-22-CE25-0016; coco5g.roc.cnam.fr), TREES (ANR-24-TSIA-0004; trees.roc.cnam.fr) and IPCEI ME/CT (DOS0239248/00; ipcei-me-ct.eu) projects.

References

[1] J. Ren, D. Zhang, S. He, Y. Zhang, and T. Li, “A survey on end-edge-cloud orchestrated network computing paradigms: Transparent computing, mobile edge computing, fog computing, and cloudlet,” *ACM Computing Surveys (CSUR)*, vol. 52, no. 6, pp. 1–36, 2019.

[2] N.-E.-H. Yellas, B. Addis, R. Riggio, and S. Secci, “Function Placement and Acceleration for In-Network Federated Learning Services,” in *CNSM’22*, 2022.

[3] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, “Edge intelligence: Paving the last mile of artificial intelligence with edge computing,” *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Artificial intelligence and statistics*, pp. 1273–1282, 2017.

[5] Z. Lu, H. Pan, Y. Dai, X. Si, and Y. Zhang, “Federated learning with non-iid data: A survey,” *IEEE Internet of Things Journal*, 2024.

[6] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, “Federated optimization in heterogeneous networks,” *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[7] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, “SCAFFOLD: Stochastic Controlled Averaging for Federated Learning,” 2021.

[8] P. Ntumba, N.-E.-H. Yellas, S. Bin-Ruba, F. Ben Abdesslem, and S. Secci, “Data pipeline system designs for in-network learning,” in *CNSM’24*, 2024.

[9] Q. Li, Y. Diao, Q. Chen, and B. He, “Federated Learning on Non-IID Data Silos: An Experimental Study,” in *IEEE International Conference on Data Engineering (ICDE)*, 2022.

[10] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, “Federated learning with non-IID data,” *arXiv:1806.00582*, 2018.

[11] M. Luo, F. Chen, D. Hu, Y. Zhang, J. Liang, and J. Feng, “No fear of heterogeneity: Classifier calibration for federated learning with non-iid data,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 5972–5984, 2021.

[12] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, “Tackling the objective inconsistency problem in heterogeneous federated optimization,” *Advances in neural information processing systems*, vol. 33, pp. 7611–7623, 2020.

[13] S. B. Ruba, N. E.-H. Yellas, and S. Secci, “Anomaly detection for 5G softwarized infrastructures with federated learning,” in *International Conference on 6G Networking (6GNet)*, 2022.

[14] Z. Zhao, J. Xia, L. Fan, X. Lei, G. K. Karagiannidis, and A. Nallanathan, “System Optimization of Federated Learning Networks With a Constrained Latency,” *IEEE Transactions on Vehicular Technology*, vol. 71, no. 1, pp. 1095–1100, 2022.

[15] A. Rakotomamonjy, M. Vono, H. J. M. Ruiz, and L. Ralaivola, “Personalised Federated Learning On Heterogeneous Feature Spaces,” 2023.

[16] A. Caticha, “Relative entropy and inductive inference,” in *American Institute of Physics Conference*, vol. 707, 2004.

[17] T. Van Erven and P. Harremos, “Rényi divergence and kullback-leibler divergence,” *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 3797–3820, 2014.

[18] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani, *et al.*, “Toward generating a new intrusion detection dataset and intrusion traffic characterization.,” *ICISSp*, vol. 1, no. 2018, pp. 108–116, 2018.