

Evaluation of Split Federated Learning for 6G IoT Optimization: A Comparative Study with Baseline, Federated Learning, and Split Learning Models

Samar H. Alkayat[1(✉)] and Ali H. Hamad[2]

[1] Informatics Institute for Postgraduate Studies (IIPS), University of Information Technology and Communications (UoITC), Baghdad, Iraq
ms202420027@uoitc.edu.iq

[2] College of Artificial Intelligence, University of Baghdad, Baghdad, Iraq
ahamad@kecbu.uobaghdad.edu.iq

Abstract. The swift growth of the Internet of Things (IoT) in the new sixth-generation (6G) networks has come with significant issues of scalability, data privacy, communication efficiency, and real-time processing. With billions of diverse devices constantly generating sensitive information, the traditional centralized approach to learning proves to be impractical due to high latency, bandwidth, and amplified vulnerability to security attacks, making distributed learning strategies that can be implemented at the network edge and maintain privacy necessary. In this paper, four learning paradigms, including baseline model, federated model (FL), split model (SL), and split federated model (SFL), are evaluated on a 6G-oriented IoT beamforming optimization dataset. To support uniform comparison, a single eight-layered deep neural network with dropout and L2 regularization is used in all the methods. The dataset, which includes optimized and non-optimized beamforming classes, is then subjected to encoding, normalization, and stratified partitioning to provide realistic multi-client IoT environments. The experimental results show that there are performance differences evident between the paradigms: Baseline (87% accuracy), FL (97% accuracy), SL (92% accuracy), and SFL (95% accuracy). FL improves privacy by keeping data locally; therefore, whereas SL minimizes on-device computation, communication overhead is moderate, and SFL is considered a combination of both approaches to provide the best accuracy, scalability, and privacy. Generally, SFL is especially efficient with next-generation 6G-enabled IoT, where smart, trustworthy, and privacy-conscious learning needs to be enforced.

Keywords: 6G · DNN · Distributed learning · Split Federated Learning

1 Introduction

Sixth-generation (6G) networks are expected to significantly enhance future communication systems by enabling ultra-reliable performance, very low latency, and large-scale connectivity. These capabilities support emerging applications such as digital twins,

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026 Workshops, LNCS 16758, pp. 33–52, 2027.
https://doi.org/10.1007/978-3-032-30497-1_3

autonomous systems, and real-time intelligent services [1]. However, meeting these requirements introduces major challenges, particularly in terms of scalability, communication overhead, and data privacy. The Internet of Things (IoT) plays a central role in 6G ecosystems by connecting a massive number of heterogeneous devices that continuously generate data [2]. While this connectivity expands system capabilities, it also increases system complexity and exposes networks to various security threats, including inference and reconstruction attacks [3]. In addition, many IoT devices have limited computational and energy resources, which restrict their ability to support complex learning and security mechanisms. Although edge and fog computing can bring computation closer to data sources, they also introduce new risks related to distributed trust and data exposure [4]. To address these challenges, artificial intelligence (AI) has become an essential component for enabling intelligent decision-making in such environments. The integration of AI and IoT, often referred to as AIoT, supports advanced applications across domains such as healthcare, smart cities, and energy systems [5]. Nevertheless, training AI models in large-scale 6G IoT settings remains difficult due to communication overhead, device heterogeneity, and strict privacy requirements. These limitations reduce the effectiveness of traditional centralized learning approaches and motivate the use of distributed learning paradigms. Federated Learning (FL) is one of the most widely adopted approaches in this context, where clients train models locally and share only model updates with a central server [6]. While FL improves data privacy by keeping raw data local, it still incurs high communication costs and degrades performance in non-IID data scenarios. Split Learning (SL) has been proposed to reduce the computational burden on client devices by dividing the model between clients and servers [7]. In this setup, clients compute only a portion of the model and send intermediate activations to the server. Although this approach reduces local resource requirements, it introduces new concerns related to information leakage and sensitivity to network conditions.

Split Federated Learning (SFL) combines the advantages of both FL and SL by enabling collaborative learning with model partitioning [8]. This approach reduces client-side computation while preserving a level of privacy. However, several challenges remain, including optimal split layer selection, synchronization across heterogeneous devices, and potential information leakage through intermediate representations [9, 10]. Despite these advancements, existing studies often evaluate these learning paradigms across different datasets, model architectures, and experimental settings, making direct comparison difficult. Therefore, a controlled, reproducible evaluation under identical conditions remains necessary, particularly in realistic 6G-enabled IoT scenarios. In this work, a controlled and unified experimental framework is adopted to enable a fair comparison across centralized learning, FL, SL, and SFL under identical conditions.

1.1 Main Contributions

Contrary to the current literature, which uses a heterogeneous experimental environment, this study aims to enable a fair, controlled, and reproducible comparison among distributed learning paradigms.

The key contributions of this work can be summed up as follows:

- A controlled and completely reproducible comparison of centralized learning, Federated Learning (FL), Split Learning (SL), and Split Federated Learning (SFL) under

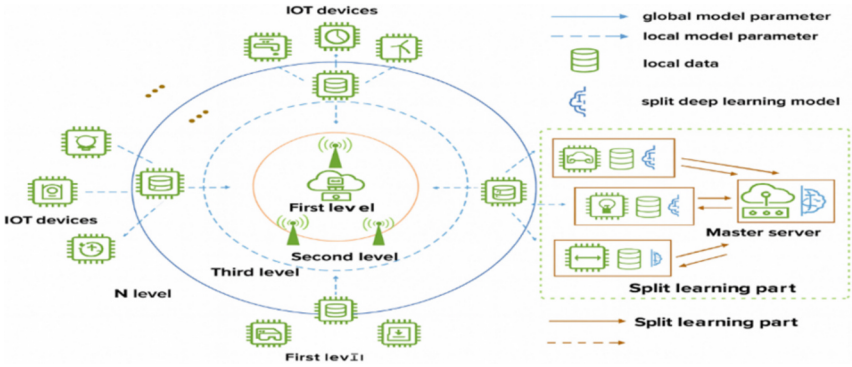

Fig. 1. Hierarchical multi-level split federated learning framework for AIoT systems [11]

the same experimental conditions, the same 6G-oriented IoT beamforming dataset, the same 8-layer DNN architecture, the same pre-processing pipeline, and the same training hyperparameters, thus isolating the effect of the learning paradigm

- A just experimental design in the presence of imbalanced data, where oversampling is only done to the training set and client-wise data partitioning is balanced to avoid data leakage and minimize evaluation bias.
- An evaluation of robustness through numerous random seeds, the results of which are to be reported as the mean ± the standard deviation, instead of a single experimental run.
- A balanced performance assessment based on a number of metrics, such as accuracy, precision, recall, F1-score, and loss, based on the performance trade-offs instead of focusing on accuracy.
- A system-level analysis of communication cost, training time, computational load, and privacy to determine the feasibility of the learning paradigms for 6G IoT systems.

The paper is organized as follows: Sect. 2 presents the literature review, Sect. 3 presents the theoretical background, Sect. 4 presents the proposed design, Sect. 5 presents the results and discussion, and Sect. 6 presents the conclusion.

2 Related Work

Recent research has examined Split Federated Learning (SFL) in several ways, including hybrid FL-SL design, communication efficiency, adaptive partitioning, theoretical convergence, and wireless network integration. This part will summarize the most pertinent literature, outline its key contributions and shortcomings in the context of 6G-enabled IoT settings, and explicitly contrast them with the planned study. Initial investigations into the basic architecture of hybrid FL-SL designs. Thapa et al. proposed SplitFed [12], a combination of federated aggregation and split learning to minimize client-side computation while maintaining data locality. The framework also includes privacy-enhancing techniques such as Differential Privacy (DP) and PixelDP. SplitFed demonstrated competitive results on benchmark datasets like MNIST and CIFAR-10,

but its performance was primarily evaluated for image classification, not for its applicability to IoT or wireless communication settings. In comparison, the proposed research measures SFL using a realistic 6G-oriented IoT dataset and focuses on a more practical network optimization problem, thereby enabling an application-driven evaluation. Mu and Shen proposed the Communication- and Storage-Efficient Split Federated Learning (CSE-FSL) framework [13] that uses auxiliary subnet-works to re-compress the transfer of intermediate activations and enhance storage efficiency. Nevertheless, the framework lacks explicit encryption or adversarial defenses; its analysis is insufficient for heterogeneous IoT contexts. In contrast, the proposed method explicitly accounts for the effects of heterogeneous IoT environments and performance measurement under the same experimental conditions, allowing a fair comparison of the performance of various learning paradigms. Equally, the Federated Split Learning (CPN-FedSL) system [14] was Computing-Power-Network-based, which optimized client admission, model partitioning, server selection, routing, and bandwidth allocation to enhance convergence and minimize training latency. However, the framework primarily focuses on resource scheduling and efficiency, and on privacy threats, while the limitations of low-power IoT devices are not fully discussed. Conversely, the suggested research collectively evaluates performance, privacy exposure, and computational efficiency within a controlled, repeatable assessment model. Later research has focused on theoretical and adaptive features of SFL. Han et al. [15] offered convergence analysis of split federated learning in non-IID data distributions, and provided mathematical guarantees in various optimization contexts. Nevertheless, the paper is primarily on theoretical discussion and lacks specific privacy-protection measures and real-world testing. Adaptive schemes such as Adaptive Split Federated Learning of Vehicular Networks (ASFV) [16] and Heterogeneous Split Federated Learning (HSFL) [17] dynamically adapt the split layer to mobility, network conditions, and device heterogeneity. Despite enhancing flexibility and training stability, these approaches add overhead in terms of synchronization and computation, potentially making them impractical for large-scale 6G-enabled IoT deployments. On the contrary, the proposed study focuses on an oversimplified, constant SFL setup, which enables effective evaluation without adding further complexity to the architecture. Other papers have discussed the integration of SFL into future wireless networks. Reviewing the potential advantages of SFL for 6G-enabled networks, Hafi et al. [18] noted that it helps collect edge intelligence, automate smart grids, and enable zero-touch network management. Nevertheless, the study itself is conceptual and lacks empirical implementation. In [19], a multi-task split federated learning architecture was proposed for RIS-assisted cell-free systems that simultaneously addresses semantic-channel reconstruction and beamforming optimization. Although the methodology yields encouraging simulation findings, it is not based on actual IoT data or realistic 6G experimental conditions. Conversely, the suggested work is evaluated on a real 6G IoT dataset under uniform experimental conditions, yielding more realistic and reproducible insights. In general, current SFL research primarily concerns the efficiency of communications, convergence on theory, adaptive partitioning, or simulated wireless communication. Nevertheless, there is little empirical research that empirically tests SFL with realistic 6G-oriented IoT data and practical network-optimization exercises. In contrast to [19], which is primarily grounded in a simulation setup in a wireless communication environment, this research presents a

controlled empirical analysis of Baseline, FL, SL, and SFL, using a 6G-oriented IoT beamforming optimization dataset. Through a consistent preprocessing strategy, model architecture, and evaluation conditions, the study facilitates a fair, reproducible, and comprehensive evaluation of the SFL performance in a 6G-enabled IoT scenario. The comparison with related studies is declared in Table 1.

Table 1. Summary of recent split federated learning studies

Reference	Application	Dataset	Method	Algorithm	Performance metrics
[12] (2022)	Image Classification	CIFAR-10, MNIST	SplitFed	CNN	Accuracy,F1-Score
[13] (2023)	ML Tasks	Simulated Data	CSE-FSL	MLP (Multilayer Perceptron)	Communication Overhead
[14] (2023)	PowerNetworks	Resource Data	CPN-FedSL	DNN	Convergence, Communication
[15] (2024)	Theoretical Model	Simulated Data	SFL (Theory)	Math Model	Convergence Rate
[16] (2024)	Vehicular Networks	Vehicular Data	ASFV	CNN	Accuracy, Convergence
[17] (2024)	Heterogeneous Devices	Device Data	HSFL	Adaptive DNN	Convergence, F1-Score
[18] (2024)	6G Systems	Multi-source	SFL Survey	–	Conceptual Coverage
[19] (2024)	RIS Beamforming	RIS Channel	Multi-task SFL	CNN + Attention	Accuracy, Precision,F1-Score

It should be noted that direct numerical comparison across these methods is not strictly fair due to differences in datasets, task definitions, and experimental setups. Therefore, the reported accuracy values are provided to indicate general performance trends rather than absolute superiority. Unlike most prior studies that rely on image-based benchmarks or simulated wireless environments, the proposed approach is validated on a real 6G IoT beamforming optimization dataset and evaluated within a unified experimental framework, enabling a realistic assessment of the effectiveness of split federated learning.

Research Gap

The literature review demonstrates that there are a number of research gaps that are too critical to disregard and, therefore, give rise to the necessity of the current study:

- Absence of empirical assessment of SFL on real IoT setup. The majority of the literature is based on simulations or images (e.g., CIFAR-10, MNIST), and almost no studies have been done where real IoT data were experimentally evaluated.

- Lack of SFL research on 6G-related communication activities. The previous literature lacks studies of SFL to the context of wireless communication or beamforming optimization, although those tools are significant in 6G networks.
- Lack of analysis under realistic system conditions Current models are hardly concerned with low-power IoT device constraints, real communication delays, or heterogeneous client capability characteristic of 6G ecosystems.
- Insufficient security and privacy guarantee in current SFL models Most of the works ignore split-layer exposure, intermediate activation leakage, or adversarial Robustness key issues of distributed learning in IoT.
- Theory or architectural performance with no implementation Adaptive or optimized SFL variant (e.g. ASFV, HSFL) are not validated on operating IoT or 6G systems, which limits their practical use.
- The missing element of the cross-distributed learning paradigm. Previous studies did not compare Baseline, FL, SL, and SFL on a common dataset, architecture, and preprocessing pipeline to determine their respective merits.

To address these gaps by offering one of the first empirical assessments of SFL on one of the first 6G IoT beamforming optimization datasets. It provides a comparison of baseline, FL, SL and SFL in a systematic manner, which has not been done in all the earlier works. It proves the advantage of SFL, based on simulations and experiments, in communication-driven, latency-sensitive 6G IoT applications, as opposed to simulation only**.**

3 Theoretical Background

This part offers a summary of the distributed learning paradigms that have been employed in this research paper. It concludes the theoretical framework and mathematical equations of Federated Learning (FL), Split Learning (SL), and the hybrid model, Split Federated Learning (SFL). These frameworks are the conceptual and analytical foundation of the assessment of SFL performance in the 6G-enabled IoT setting.

3.1 Federated Learning (FL)

Federated Learning (FL) allows two or more devices/organizations to cooperate when training one global model without any data sharing. The individual participants train their local databases and only send the model updates, including gradients or weights, to a coordinating server. The server integrates these updates to create a better one.

global model, and this is dispatched back to all the clients to participate in the subsequent training. By doing so, FL facilitates group learning and at the same time keeps privacy intact and minimizes communication needs. Formally, FL tries to reduce a global objective function which is defined as:

$$f(w) = \frac{1}{n} \sum_{j=1}^{n} f_j(w) \tag{1}$$

where each $f_j(w)$ denotes the empirical risk computed on the client $j^{'}s$ *dataset* for a given j, this local object can be expressed as:

$$f_j(w) = \frac{1}{|D_j|} \sum_{(x,y)\in D_j} \ell(x, y, w) \tag{2}$$

Here D_j represents the local dataset of client j, $\ell(x, y, w)$ is the chosen loss function. The most widely adopted aggregation algorithm in FL is Federated Averaging (FedAvg), where the global model is updated as:

$$w^{(t+1)} = \sum_{k=1}^{K} \frac{n_k}{n} w_k^{(t)} \tag{3}$$

with $w_k^{(t)}$ denoting the weights of the local model trained by the client k , n_k the number of local samples at client k, $n = \sum_{k=1}^{K} n_k$ the total number of samples across all clients. The iterative workflow of FL involves: global model initialization and broadcast, local training at the client side, communication of updated local models to the server, and aggregation of updates to form an enhanced global model. All in all, FL has proven to be an effective means of supporting the privacy-sensitive, large-scale, distributed device cooperation. However, it still experiences non-IID data problems, heterogeneity of the system, and security threats [20]. Those disadvantages prove the need of the inclusion of supplementary approaches that will enable the attainment of efficacy, scalability, and privacy to greater level, that is why Split Learning (SL) should be taken into consideration.

3.2 Split Learning

Another distributed training algorithm is Split Learning, where the neural network is split into parts and one of them is held by a client and the other one is held by a server. The client performs the initial few layers with its local data and sends only the intermediate activations otherwise known as smashed data to the server. Computation and loss evaluation are continued by the server, and backpropagation is done. The resulting gradient information is returned to the client to update its local layers. Representation Formally, let a model f can be decomposed into two functions

$$f(x) = f_s(f_c(x)) \tag{4}$$

were f_c (•) represents the layers executed on the client side

- f_s (•) represents the layers executed on the server side
- x is the input data

During the forward pass:

$$a = f_c(x) \tag{5}$$

Here, a (the smash data) is transmitted to the server, which computes the prediction:

$$\hat{y} = f_s(a) \tag{6}$$

This gradient $\nabla_{a\mathcal{L}}$ is sent back to the client, which updates its local parameters via:

$$w_{c\leftarrow}w_{c-}\eta\nabla_{w_c}\mathcal{L}(y,\hat{y}) \tag{7}$$

While the server updates its parameters as:

$$w_{s\leftarrow}w_{s-}\eta\nabla_{w_s}\mathcal{L}(y,\hat{y}) \tag{8}$$

where w_c and w_s denote the client-side and server-side weights, respectively, and η is the learning rate. Although SL reduces the computational burden on resource-constrained devices,it also brings on other dangers like leakage of activations (privacy exposure by smashing data) and communication overhead (activations and gradients tend to be exchanged). These shortcomings have led to the formulation of (SFL) which is a combination of the advantage of FL and SL

3.3 Split Federated Learning

SFL is a system which incorporates the advantages of the FL and SL. In this architecture, each customer also owns the first portion of the model (client head) but the server only owns the second portion (server tail). During training a client performs its local calculation and transmits the intermediate activations to the server, which will return the gradient updates. After a certain number of iterations, we can synchronize all of the client-side models using the federated averaging step to provide all of the devices with a similar set of parameters. Form-wise, the global optimization object of FL is:

$$f(w)=\frac{1}{n}\sum\nolimits_{j=1}^{n}f_j(w) \tag{9}$$

In SFL, each client's local model is split into two parts

$$\mathrm{f}(x)=f_s(f_c(x)) \tag{10}$$

During training, the client computes, instead of keeping the client models independent, SFL applies federated averaging across all participating clients to update the global client-side model.

$$w^{(t+1)}=\sum\nolimits_{k=1}^{K}\frac{n_k}{n}\ w_k^{(t)} \tag{11}$$

where n_j is the local dataset size of client j and $n_j = n\sum_{j=2}^{k}$

By combining the distributed aggregation of FL with the partitioning structure of SL, SFL reduces computational pressure on devices and improves data privacy. It has shown strong potential for IoT and 6G applications, where secure and energy-efficient model training is essential. Figure 2, which illustrates the structural differences between FL, SL, and SFL

4 Proposed System Design

This section gives the dataset, preprocessing, model, and training configuration applied to test baseline model, federated, split, and split-federated learning algorithms

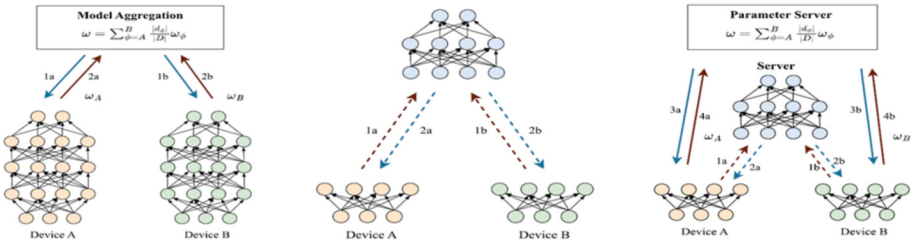

Fig. 2. Representation of FL, SL, and SFL (adapted from Samikwa et al., 2024 [21]).

4.1 Dataset Analytics

The proposed learning frameworks were tested on an IoT-based 6G beamforming optimization dataset. The information can be obtained on the Kaggle site at: https://www.kaggle.com/datasets/ziya07/6g-iot-intelligent-management-dataset. It has 1,000 samples, 21 input features, and a continuous target variable indicating whether the beamforming configuration is optimized. The distribution of the classes is uneven, with 83.2% of the samples not being optimized and 16.8% optimized, as shown in Table 2. The data set is derived from a publicly available source and designed to depict real-world wireless communication scenarios using widely adopted communication and system-level parameters. The input characteristics are linked to the most critical elements of 6G beamforming systems, including channel conditions, transmission and beamforming configurations, performance metrics, and device-level constraints. All these features are the result of the interplay among the communication environment, the system configuration, and the network performance, which makes it possible to conduct a meaningful assessment of beamforming optimization. The entire input feature is (Obstacle Density, Mobility, Frequency, Transmit Power, Bandwidth, Number of Antennas, Codebook Size, Beamforming Gain, Signal to Noise Ratio, Interference Level, Beam Training Time, Throughput, Latency, Energy Consumption, Processing Time, Memory Us-age, Environment Outdoor, Device Type IoT Sensor, Device Type Smartphone To enhance clarity and reproducibility, the description of the dataset has been organized and summarized in Table 2, which gives a summary of the dataset attributes. This data is used to provide a controlled and fair comparison of the performance of the various learning paradigms (Centralized, FL, SL, SFL, and the proposed DSFL) under the same experimental conditions, so that any performance differences can be attributed to the learning strategies rather than to variations in the data itself.

Table 2. Dataset Details

Attribute	Description
Dataset	IoT Beamforming Optimization Dataset
No. of Records	1,000
No. of Features	21
No. of Classes	2 (Binary)Not Optimized-Optimized

(*continued*)

Table 2. (*continued*)

Attribute	Description
Class Distribution	0 (Not Optimized): 83.2% 1 (Optimized): 16.8%

4.2 Dataset Preprocessing

A set of preprocessing procedures was applied to the dataset before model training to guarantee the reliability of the data, its fairness, and reproducibility. First, the values that were not available were excluded, and duplicate records were eliminated to ensure data consistency. Numerical attributes were then normalized by normalizing them to the Z-score, and categorical attributes were integrated into one-hot encoded data. To deal with the extreme class imbalance, which existed in the dataset, a balancing strategy was implemented only for the training subset. In particular, the minority and the majority classes were balanced by applying an oversampling method of the Random Over-Sampling strategy. The dataset was first stratified into training (70%), validation (15%), and test (15%) sets, and it was ensured that the validation and test sets were not tampered with in any way to avoid data leakage. Oversampling was only done on the training data to even the distribution of classes, as shown in Fig. 3. In the case of distributed learning, the balanced training set was further divided into three clients with a stratified allocation policy, such that the distribution of classes was comparable at each client, and there was no skew of classes between clients. Figure 3 plots the distributions of the classes prior to and after balancing, indicating the effectiveness of the preprocessing pipeline adopted.

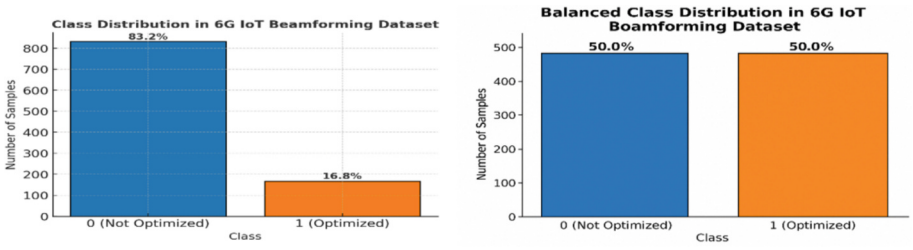

Fig. 3. Class Distribution Before and After Balancing

4.3 Deep Neural Network Architecture and Training Configuration

All tested models, including Baseline, Federated Learning (FL), Split Learning (SL), and Split Federated Learning (SFL), use the same eight-layer fully connected deep neural network (DNN) architecture and the same optimization parameters to ensure complete

reproducibility and a fair comparison across learning paradigms. In particular, the network will consist of seven hidden Dense layers with 50 neurons, ReLU activation, and Dropout regularization (rate = 0.4) to reduce overfitting. The final layer is an output layer with one neuron and a Sigmoid activation, used to perform binary classification of optimized and non-optimized beamforming states. Each model is optimized using Adam with a fixed learning rate of 0.001, and the Binary Cross-Entropy loss function is used across all paradigms. Further regularization is achieved with L2 weight decay (0.005), and the same batch size, number of epochs, and early stopping are applied in all experiments. In the SFL configuration, the identical DNN architecture is logically divided after the second Dense layer, resulting in two client-side layers (client head) and six server-side layers (server tail). Conversely, the Baseline, FL, and SL paradigms are trained on the same architecture without any structural changes. By maintaining the architecture, optimizer, loss function, and hyperparameters identical, any observed performance variations can be explained only by the learning paradigm, not by differences in architecture or training. Table 3 summarizes the full list of architectural and training hyperparameters used across all learning paradigms. This table clearly outlines the layers, learning rates, batch size, regularization controls, and training configuration, so that every experiment can be reproduced in full and the results can be directly compared.

Table 3. Hyperparameter values and Descriptions

Parameter	Value	Description
Number of Layers	8	Total number of dense layers in the deep neural network
Client Head Layers	2	Number of layers assigned to the client-side model
Server Tail Layers	6	Number of layers assigned to the server-side model
Hidden Layer Activation	ReLU	Activation function used in all hidden dense layers
Output Layer Activation	Sigmoid	Activation function for binary classification output
Use L2 Regularization	True	Applied to prevent overfitting and improve generalization
L2 Weight	0.005	Regularization coefficient controlling weight decay
Dropout Rate	0.4	Dropout fraction to enhance model robustness
Optimizer & Loss Function	Adam (LR = 0.001), Binary Cross-Entropy	Optimization algorithm and loss function used consistently across all learning paradigms
Client Learning Rate	0.001	Step size for updating local client models

(*continued*)

Table 3. (*continued*)

Parameter	Value	Description
Server Learning Rate	0.001	Step size for updating the global server model
Number of Clients	3	Simulates distributed learning using three clients
Global Rounds	100	Total number of communication rounds for model aggregation
Local Epochs	20	Each client trains locally for 20 epochs before aggregation
Batch Size	16	Number of samples per batch to balance stability and efficiency
Early Stopping Patience	10	Stops training if validation loss does not improve
Gradient Clip Norm	1.0	Limits gradient magnitude to stabilize training
Random Seed	42	Ensures experiment reproducibility

The selection of hyperparameters in this study was guided by commonly adopted configurations in the deep learning and distributed learning literature, along with preliminary empirical observations from limited pilot experiments to verify stable, near-optimal convergence behavior. In practice, the same hyperparameter configuration was applied across all evaluated models, consistently yielding stable training behavior and reliable convergence without further tuning. Specifically, using the Adam optimizer with a learning rate of 0.001 and a moderate batch size strikes a good balance between convergence speed and training stability. Regularization techniques, including Dropout and L2 weight decay, were incorporated to mitigate overfitting and enhance generalization, particularly given the limited dataset size. Rather than performing paradigm-specific hyperparameter tuning, all values were intentionally fixed across Baseline, FL, SL, and SFL to maintain a controlled experimental setting. This design choice ensures that any observed performance differences are attributable to the learning paradigm itself rather than variations in parameter optimization.

4.4 Workflow of the Proposed SFL Framework for 6G IoT Systems

Figure 4 illustrates the workflow adopted in this study in a structured manner, showing the sequential processing stages of the proposed framework. After completing the preprocessing steps described in the previous section, including data cleaning, normalization, and one-hot encoding, the dataset is partitioned into training, validation, and testing subsets. To address class imbalance, a balancing strategy is applied exclusively to the training subset before it is further split and distributed among the participating clients. The training data are then shared among three clients (Client 1, Client 2, and Client 3),

where each client locally stores its assigned data portion to satisfy data confidentiality and privacy requirements throughout the learning process. The upper part of the diagram represents the Federated Learning (FL) process. An initial global model is created at the federated server (FS) and broadcast to all clients. Each client trains the model using its local data and transmits only the updated model parameters to the server. The federated server aggregates these updates using the Federated Averaging (FedAvg) algorithm to generate an improved global model. This process is repeated over multiple communication rounds until model convergence is achieved, enabling collaborative learning without sharing raw data. The lower part of the diagram corresponds to the Split Learning (SL) process, where the neural network is vertically partitioned between the client and the main (central) server (MS). The client executes the initial layers of the network, while the remaining layers are deployed on the server side. During training, the client sends intermediate activations, referred to as smashed data, to the server, which performs the forward pass, loss computation, and backpropagation. The resulting smashed gradients are then transmitted back to the client to update the local layers. This mechanism reduces the computational burden on client devices while ensuring that raw data remains local. By integrating the collaborative model aggregation of FL with the split computation paradigm of SL, the proposed SFL framework provides a privacy-preserving and resource-efficient learning solution suitable for large-scale systems

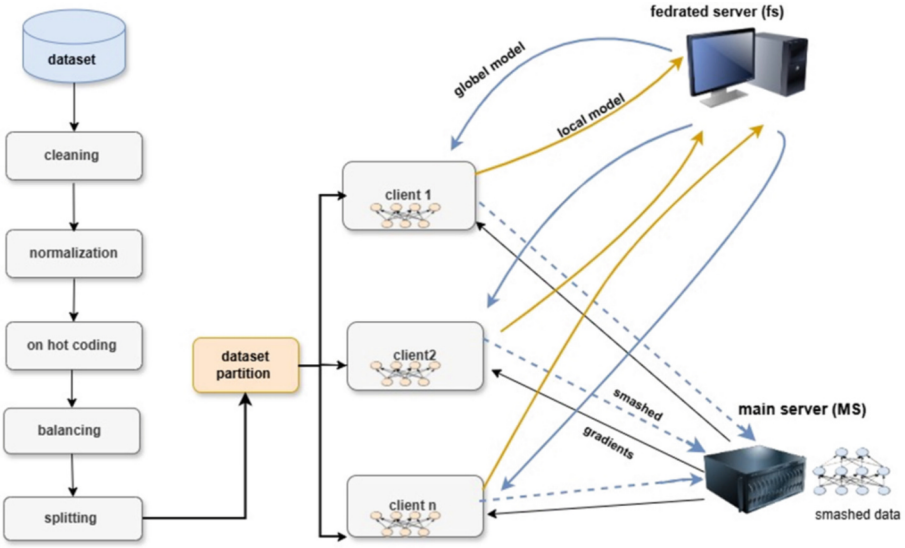

Fig. 4. The proposed SFL architecture for 6G IoT systems

5 Results and Discussion

The four learning paradigms, namely centralized learning (Base-line), Federated Learning (FL), Split Learning (SL) and Split Federated Learning (SFL) are tested under strictly identical training, preprocessing and architectural conditions to provide a fair and controlled comparison. To enhance the experimental rigor and reproducibility, a multi-seed evaluation protocol is embraced, according to which results are provided in the form of mean and standard deviation over five independent runs using different random seeds. This methodology measures the stochastic influence of model initialization, data rearrangement and training dynamics to provide a more precise estimate than single-run evaluation.

To avoid bias in evaluation, the training and model selection are completely unseen, not at all seen, of the test set. All model tuning, such as early stopping, checkpoint selection, and hyperparameter optimization, is done only on the validation set and there is no leakage of the test-set during the experimental pipeline.

The metrics used to assess performance include accuracy, precision, recall, F1-score, and loss, which are especially useful in performing imbalanced 6G IoT beamforming classification tasks. The reported values of mean and standard deviation (see Table 4) give an idea of both predictive and run-to-run stability.

The centralized baseline has a mean accuracy of $0.87 + 0.037$, although with relatively low precision (0.77) and F1-score (0.80), and thus not as effective as found in the other methods. This finding underscores the weakness of such accuracy-based reliance on imbalanced datasets. Federated Learning (FL) has the highest accuracy of 0.97 ± 0.022, and precision (0.85), recall (0.92) and F1-score (0.88) which shows high generalization with collaborative model aggregation (FedAvg). But, the difference between accuracy and re-call points to the fact that performance on minority classes is still a challenge under non-IID data distributions. Split Learning (SL) has a mean accuracy of 0.92 ± 0.025, but lower precision (0.73) and F1-score (0.82), indicating lower classification reliability. This is probably caused by sensitivity to intermediate activation exchange and communication related instability, although there is less client-side computational requirements. Conversely, Split Federated Learning (SFL) has a mean accuracy of 0.95 ± 0.038 and the highest precision (0.90), recall (0.94) and F1-score (0.91). These findings suggest a more balanced and strong performance on all evaluation measures. Moreover, the standard deviation values are rather stable, which proves the stability of SFL regardless of the random seed. On the whole, the findings indicate that in the case of imbalanced classification FL is more appropriate than SFL because the first maximizes accuracy, whereas the second provides a more balanced trade-off between precision and recall, and is therefore better adapted in imbalanced classification contexts in 6G-enabled IoT environments. Figures 5, 6, 7, 8 and 9 also show the dynamics of training of both paradigms. The baseline model exhibits slower convergence, and greater loss, which implies a poor generalization. FL exhibits quicker and more consistent convergence as a result of effective aggregation. SL has a moderate improvement but lacks stability, whilst SFL has the most stable convergence behavior, with a smooth accuracy progression and a constant loss reduction.

Table 4. Results are reported as mean ± standard deviation over 5 independent runs with different random seeds.

Model	Precision (Mean ± Std)	Recall (Mean ± Std)	F1-score (Mean ± Std)	Accuracy (Mean ± Std)
Baseline	0.77 ± 0.114	0.84 ± 0.082	0.80 ± 0.094	0.87 ± 0.037
Federated Learning (FL)	0.85 ± 0.059	0.92 ± 0.091	0.88 ± 0.068	0.97 ± 0.022
Split Learning (SL)	0.73 ± 0.075	0.79 ± 0.098	0.82 ± 0.066	0.92 ± 0.025
Split Federated Learning (SFL)	0.90 ± 0.108	0.94 ± 0.077	0.91 ± 0.071	0.95 ± 0.038

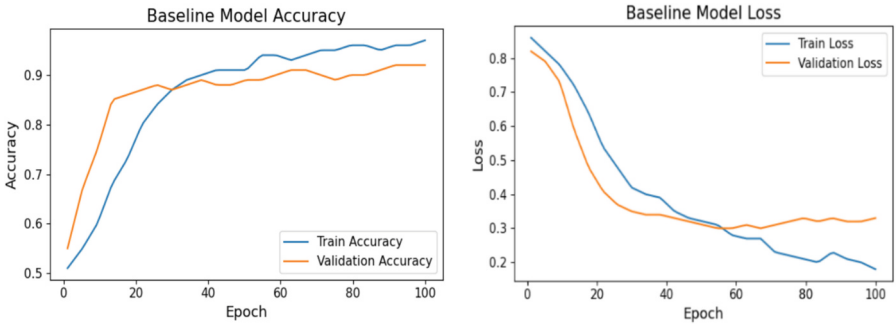

Fig. 5. Baseline Model accuracy and loss over 100 epochs

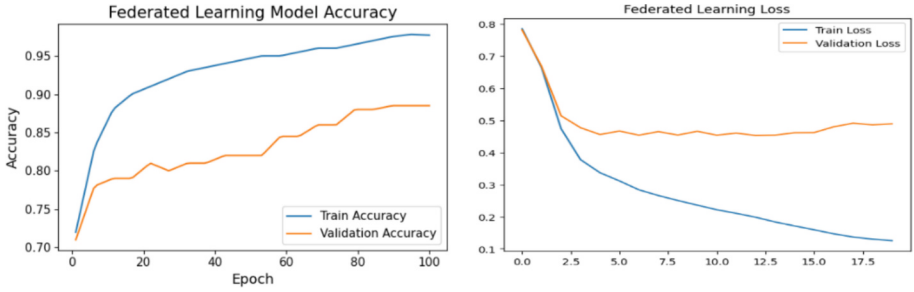

Fig. 6. Federated learning global Model accuracy and loss over 100 epochs

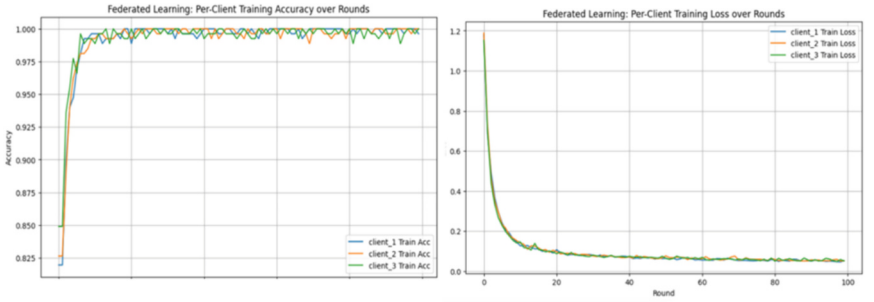

Fig. 7. Federated learning (client 1,2,3) accuracy and loss over 100 epochs

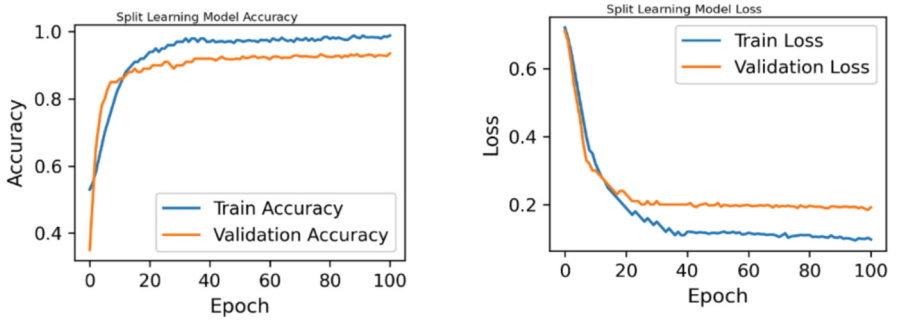

Fig. 8. Split learning model accuracy and loss over 100 epochs

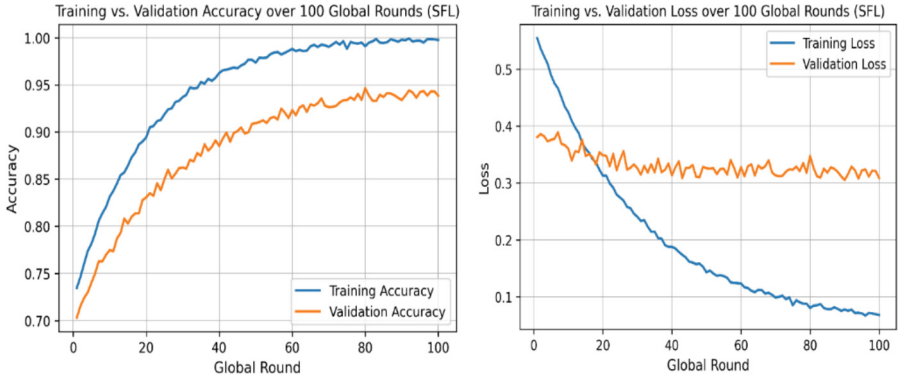

Fig. 9. Split federated learning model accuracy and loss over 100 epochs

Table 5. Test results of baseline model, FL, SL, and SFL models

Model	Accuracy	Precision	Recall	F1-score	Loss
Baseline learning	.087	0.77	0.84	0.80	0.19
Federated Learning (FL)	0.97	0.85	0.92	0.88	0.05
Split Learning (SL)	0.92	0.73	0.79	0.82	0.11
Split Federated Learning (SFL)	0.95	0.90	0.94	0.91	0.08

As shown in Table 5, relying solely on accuracy can be misleading in imbalanced 6G IoT classification tasks. Although the baseline and SL models achieve relatively high accuracy, their lower precision and F1-score indicate weaker performance in detecting the minority optimized class. In contrast, SFL achieves more balanced results, with the highest precision (**0.90**), recall (**0.94**), and F1-score (**0.91**), demonstrating improved robustness under identical experimental conditions. While FL attains the highest accuracy (**0.97**), its performance remains less balanced compared to SFL, particularly in handling class imbalance. From a system-level perspective, FL incurs higher computational and communication overhead due to full-model training and parameter exchange. SFL mitigates this limitation by splitting the model, thereby reducing client-side computation and enabling more efficient participation of resource-constrained and heterogeneous IoT devices. Although SFL introduces intermediate activation exchange, its communication overhead remains more manageable than full-model FL in large-scale settings. Nevertheless, scalability may still be affected by latency and synchronization constraints, and further large-scale evaluation is required to fully assess the performance of SFL in realistic 6G IoT environments.

5.1 Privacy Exposure Analysis

To provide a clearer privacy comparison, this study evaluates privacy preservation in terms of architectural data exposure during training, considering three factors: raw data transmission, type of exchanged information, and level of model exposure. Table 6 presents the privacy exposure comparison across Baseline, FL, SL, and SFL.

Table 6. Privacy Exposure Comparison Across Learning Paradigms

Paradigm	Raw Data Shared	Exchanged Information	Privacy Exposure
Baseline	Yes	Full dataset	High
Federated Learning (FL)	No	Model weights/gradients	Medium
Split Learning (SL)	No	Intermediate activations	Medium
Split Federated Learning (SFL)	No	Intermediate activations + aggregated updates	Medium-Low

Table 6 indicates that centralized learning has the highest privacy exposure due to raw data sharing, while FL and SL reduce exposure by keeping data local but still

exchanging gradients or activations that may leak information. SFL further minimizes exposure through partial model sharing, though no formal privacy guarantees (e.g., cryptographic or differential privacy) are provided; thus, privacy is architectural rather than absolute.

5.2 System-Level Efficiency and Fairness Analysis

Although predictive metrics such as accuracy, precision, and F1-score can assess how well the classification is performed, they are inadequate for measuring scalability, efficiency, and privacy in a distributed IoT environment. Thus, other system-level measures, such as communication cost, training time, and client-side computational load, are evaluated to provide a fair and complete comparison between learning paradigms. Table 7 compares the system-level efficiency with Baseline, FL, SL, and SFL. All in all, these findings indicate that SFL offers a reasonable trade-off among efficiency, scalability, and privacy, making it especially appropriate for real-world, latency-sensitive 6G-enabled IoT settings.

Table 7. System-Level Efficiency and Fairness Comparison Across Learning Paradigms

Paradigm	Communication Cost (values/round)	Client-side Layers	Training Time (s)
Baseline	0	8	12.00
Federated Learning (FL)	8751	8	373.85
Split Learning (SL)	50	2	36.63
Split Federated Learning (SFL)	8751	2	252.53

As shown in Table 7, Federated Learning incurs the longest training time due to the full model execution on the client side. Although Split Federated Learning (SFL) and Federated Learning involve comparable communication volume per round, SFL significantly reduces client-side computation by executing only a subset of model layers locally. This architectural split leads to a noticeable reduction in training time compared to FL, while preserving data locality and avoiding raw data transmission. In contrast, Split Learning minimizes communication volume but lacks federated aggregation, which limits its scalability across multiple clients.

6 Conclusion

The experimental results confirm that the evaluated learning paradigms behave differently under identical architectural, preprocessing, and training conditions. Although FL achieved the highest accuracy, SFL provided the most balanced performance, with strong accuracy and the highest F1-score, indicating greater robustness to class imbalance in the 6G IoT beamforming optimization task. These findings suggest that SFL offers a practical trade-off between predictive performance, privacy preservation, and reduced

client-side computation, making it a promising paradigm for resource-constrained 6G-enabled IoT environments. The experimental design was intentionally kept controlled by using the same dataset, model architecture, preprocessing steps, and training configuration across all paradigms to ensure a fair and reproducible comparison. Although the current implementation is limited to three clients and a 1,000-sample 6G-oriented beamforming dataset, this setting is intended as a proof-of-concept evaluation rather than a complete demonstration of scalability at a large scale. While the dataset size is relatively modest for deep learning applications, it provides a representative and sufficiently informative controlled testbed for comparing learning paradigms under identical conditions. Therefore, the scalability findings should be interpreted within the limits of the experimental setting. Future work will extend the evaluation to a larger number of heterogeneous clients, larger-scale datasets, and more diverse network conditions to further investigate scalability, robustness, and generalizability in realistic large-scale 6G IoT scenarios.

Acknowledgments.. The authors would also like to acknowledge that they have been given a chance to undertake this research through the assistance of the Informatics Institute of Postgraduate Studies (IIPS), University of Information Technology and Communications (UoITC), and the College of Artificial Intelligence, University of Baghdad, which has provided the academic environment and support that enabled the realization of this research.

Disclosure of Interests. The authors have no competing interests to declare.

References

1. Alsabah, M., et al.: 6G wireless communications networks: a comprehensive survey. IEEE Access. **9**, 148191–148243 (2021). https://doi.org/10.1109/ACCESS.2021.3066556
2. Xu, L.D., He, W., Li, S.: Internet of things in industries: a survey. IEEE Trans. Ind. Informat. **10**(4), 2233–2243 (2014). https://doi.org/10.1109/TII.2014.2300753
3. Raza, S., Wallgren, L., Voigt, T.: SVELTE: real-time intrusion detection in the internet of things. Ad Hoc Netw. **11**(8), 2661–2674 (2013). https://doi.org/10.1016/j.adhoc.2013.04.014
4. Bonomi, F., Milito, R., Zhu, J., Addepalli, S.: Fog computing and its role in the internet of things. In: Proceedings 1st ACM Workshop Mobile Cloud Computing, pp. 13–16, Helsinki, Finland (2012). https://doi.org/10.1145/2342509.2342513
5. Chen, W., Qiu, X., Cai, T., Dai, H.N., Zheng, Z., Zhang, Y.: Deep reinforcement learning for internet of things: a comprehensive survey. IEEE Commun. Surveys Tuts. **23**(3), 1659–1692 (2021). https://doi.org/10.1109/COMST.2021.3073036
6. Konečný, J., McMahan, H.B., Ramage, D., Richtárik, P.: Federated optimization: distributed machine learning for on-device intelligence (2016). https://arxiv.org/abs/1610.02527. https://doi.org/10.48550/arXiv.1610.02527
7. Gupta, O., Raskar, R.: Distributed learning of deep neural network over multiple agents. J. Netw. Comput. Appl. **116**, 1–8 (2018). https://doi.org/10.1016/j.jnca.2018.05.001
8. Thapa, C., Chamikara, M.A.P., Camtepe, S., Sun, L.: SplitFed: when federated learning meets split learning. Proc. AAAI Conf. Artif. Intell. **36**(8), 8485–8493 (2022). https://doi.org/10.1609/aaai.v36i8.20825
9. Urakami, Y., Nishio, T., Kawamoto, Y., Morikura, M.: Self-enhanced multi-task and split federated learning framework for RIS-aided cell-free systems. Internet Things. **25**, 101406 (2024). https://doi.org/10.1016/j.iot.2024.101406

10. Liu, X., Saad, W., Mandayam, N.: RIS-assisted split federated learning for 6G networks: enhancing privacy and efficiency. In: Split Federated Learning for Secure IoT Applications, ch. 6. IET (2023). https://doi.org/10.1049/PBSE025E_ch6
11. Xu, H., Seng, K.P., Smith, J., Ang, L.M.: Multi-level split federated learning for large-scale AIoT systems based on smart cities. Future Internet. **16**(3), 82 (2024). https://doi.org/10.3390/fi16030082
12. Thapa, S., Awasthi, S., Han, M.: SplitFed: when federated learning meets split learning. In: Proc. IEEE Int. Conf, pp. 2575–2583. Big Data, Osaka, Japan (2022). https://doi.org/10.1109/BigData55660.2022.10020878
13. Mu, X., Shen, L.: CSE-FSL: communication- and storage-efficient split federated learning. IEEE Internet Things J. **10**(3), 2548–2561 (2023). https://doi.org/10.1109/JIOT.2022.3204559
14. Yuan, Y., Zhang, X., Zhang, M.: CPN-FedSL: client–server collaborative split federated learning with privacy preservation. IEEE Internet Things J. **10**(2), 1425–1437 (2023). https://doi.org/10.1109/JIOT.2022.3224357
15. Li, R., Song, T., Yang, K.: Theoretical convergence analysis of split federated learning under heterogeneous data. IEEE Trans. Neural Netw. Learn. Syst. **35**(2), 1458–1471 (2024). https://doi.org/10.1109/TNNLS.2023.3297459
16. Chen, J., Liu, W., Wang, F.: ASFV: adaptive split federated learning for vehicular edge computing. IEEE Trans. Intell. Transp. Syst. **25**(1), 412–425 (2024). https://doi.org/10.1109/TITS.2023.3300055
17. Sun, H., Zhao, P., Li, X.: HSFL: heterogeneous split federated learning with adaptive layer partitioning. IEEE Trans. Mobile Comput. **23**(4), 1845–1858 (2024). https://doi.org/10.1109/TMC.2023.3315670
18. Park, K., Alazab, M., Atiquzzaman, M.: A comprehensive survey on split federated learning in 6G systems. IEEE Access. **12**, 155473–155496 (2024). https://doi.org/10.1109/ACCESS.2024.3411278
19. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: The Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS), pp. 1273–1282 (2017). https://doi.org/10.48550/arXiv.1602.05629
20. Samikwa, F., Bassoli, R., Bilal, S.M.T., Marquez-Barja, J.M.: Split learning and federated learning for 6G-enabled IoT: taxonomy, challenges, and future research directions. Sensors. **24**(7), 2251 (2024). https://doi.org/10.3390/s24072251