Fuzzy-Driven Weighting Policy for Adaptive Federated Learning on non-IID Data

Antoni Jaszcz, Katarzyna Prokop, Dawid Połap, Marcin Woźniak, Rafał Brociek
Faculty of Applied Mathematics, Silesian University of Technology,
Kaszubska 23, 44-100 Gliwice, Poland
E-mails: aj303181@student.polsl.pl, katarzyna.prokop@polsl.pl, dawid.polap@polsl.pl,
marcin.wozniak@polsl.pl, rafal.brociek@polsl.pl

***Abstract*—Federated Learning (FL) describes a model training technique in which the process is distributed among separate clients, holding private training data. During an FL round, locally-trained models from clients are aggregated, forming a new global model, which is then redistributed amongst the clients. During this process, the global model indirectly utilizes information from clients' private data. However, accurate and adaptive measurement of the client's contribution to the global model is an important aspect of the FL system, especially in na on-IID scenario. In this paper, we propose a fuzzy logic-based aggregation policy that dynamically estimates the learning demand of the global model and weights the importance of clients accordingly. The server evaluates class-wise precision and recall and infers per-class learning demand that reflects whether the model needs more training exposure to the samples of each class. Client-side class distribution and dataset size are then used to determine client's contribution in the aggregation process. Unlike learned scheduling policies, the proposed approach provides transparent and easily interpretable decision rules using Sugeno-type inference. The proposed methodology was tested on MNIST, EMNIST and CIFAR-10 benchmark datasets, showing promising results in a non-IID setting.**

***Index Terms*—Federated Learning, Non-IID Data, Fuzzy Rules, Aggregation Policy**

I. Introduction

The standard solutions in Machine Learning (ML) are on the assumption, that training data can be directly acquired and used to train mathematical models. However, this is not always possible, as the learning process may be restricted by personal data protection regulations. This restriction is the main premise of Federated Learning (FL), where ML models are trained separately on clients possessing private training data and merged into a single global model during multiple federated learning rounds [1], [2]. Despite its enormous advantage in terms of privacy protection, FL systems are connected with significant challenges in practical solutions, where available training data is sparse and mostly heterogeneous [3], [4]. Such conditions result in biased local models, which directly correspond to the generalization ability of the global model. To address this issue, new aggregation techniques are constantly being developed, providing new strategies for balancing the learning process.

In recent years, various aspects of federated learning have seen significant advances. These include aggregation, consensus, security, and implemented classifier techniques. An example of such a model is the integration of learning models using different types and the fuzzy approach [5]. A fuzzy approach was used to combine predictions from three different learning models. Furthermore, the authors proposed an asynchronous solution for aggregating the model itself. Consequently, such integration allowed for a solution combining several analysis branches to increase performance efficiency. An interesting solution is presented in [6], where attention was paid to the dynamic adjustment of the learning rate and customer selection. Again, in [7] undertook research to address the problem of data heterogeneity. The idea was to introduce a cyclic clustering algorithm based on two-stage clustering. This solution also utilized a fuzzy approach to handle uncertain or incomplete data.

It is also important to pay attention to other elements in FL, as exemplified by the aforementioned aggregation. One such modification is the introduction of weighted aggregation to FL in conjunction with the Ditto algorithm [8]. The advantage of the solution is the ability to generalize thanks to data fusion.

Alternatively, weighted aggregation can be implemented depending on the size of the client's local data in FL [9]. Additionally, the authors used Lagrange interpolation to combine model parameters in a distributed system and ciphertexts. This solution allows clients to verify the correctness of the results sent by the server. In [10], a method was proposed that determines the weights in the aggregation process using client vectors in the learning process, which stores the direction of weight updates and local data changes. This method allows for weighted aggregation without violating client privacy or using additional datasets.

Based on literature analysis, it was found that introducing expert knowledge in the form of fuzzy inference system could improve the explainability within federated learning. The existing methods such as accuracy-based FedAvg, although intuitive and efficient, often lack the adaptiveness of the learning policy, hindering the the system learning capabilities in non-IID environment. In this paper, we propose a federated learning weighting policy based on per-class learning demands calculated using fuzzy inference system. The resulting learning policy is used to dynamically assess clients based on their potential to satisfy these learning demands based on their data resources. The main contributions of our paper are:

- a new model aggregation strategy in FL using fuzzy logic to adaptively weight client contributions to the global model update process,

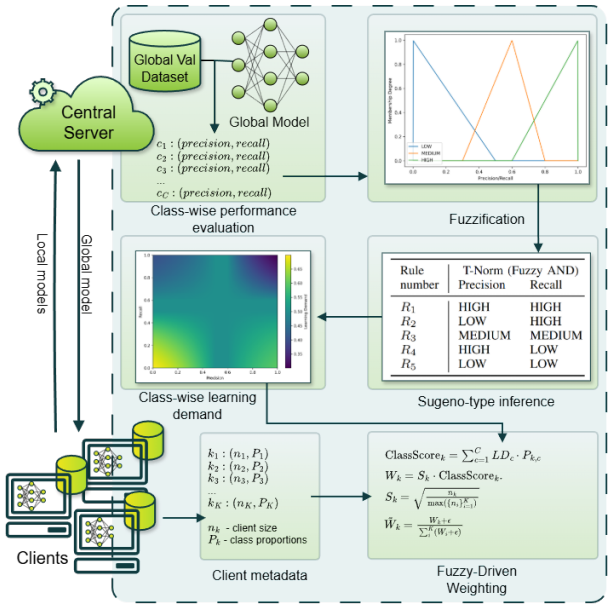

Fig. 1: Visualization of the proposed federated learning with fuzzy-driven weighting policy

- dynamic estimation of the demand for teaching individual classes via the server,
- the use of Sugeno-type reasoning as a decision-making mechanism, which allows for the transparency of the aggregation process and adaptability to various FL scenarios.

II. Methodology

A. Federated Learning

Federated Learning (FL) is a machine learning paradigm where, instead of developing a model using a single agent, the training process is built upon a system with K clients working collaboratively. The main premise of FL is that each client has a set of private data $\mathcal{D}_k$ that cannot be shared. Given that constraint, model training consists of multiple rounds, during which each client trains their private model M_k using data available to it and shares globally only the model information (f.g., in the form of model weights). After the local training concludes, the central server or chosen client aggregates local models to form a new global model M_G, which is then redistributed back among the clients.

In FL, there exist many aggregation strategies, which measure client importance, client unanimity or are simply based on learned scheduling policies. Each strategy aims to address certain issues and optimize problem-specific requirements. One of the most common and universal aggregation techniques is Federated Averaging (FedAvg) [11], in which the global model is formed as the weighted average of the client models' parameters:

$$\theta_G = \sum_{k=1}^{K} w_k \cdot \theta_k, \tag{1}$$

where the weight is typically based on the client's local dataset size or a measured performance metric. In practice, since FL often involves a large number of clients, only some of them participate in the aggregation process due to communication limitations, device availability, and scalability issues. Therefore, many FL systems introduce prior performance-based client selection for the aggregation process. In a server-based FL framework, the server evaluates clients and prioritizes the most impactful ones. This can also be implemented by using competition-driven selection, where each round, clients compete against each other in order to be selected for the aggregation and obtain a reward. While these strategies can improve convergence under non-IID data distributions, they introduce additional design challenges related to stability, fairness, and interpretability, especially when clients have uneven data distribution.

B. Server-Side Fuzzy Inference for Class Learning Demand Estimation

Considering a server-driven FL system, where the server holds a representative validation set, the aggregation process can be additionally guided by measuring the performance of the clients chosen for the aggregation. The server (also acting as an aggregator) can then base aggregation models' weights, based on measured validation metrics. In this paper, we propose introducing a weighting policy based on class-wise precision and recall scores and a Sugeno-style fuzzy inference system. The purpose of this approach is to detect underperforming classes and to prioritize clients, who can satisfy the learning demand for such class. The overview of the system is presented in: FIG REF. The proposed method makes two assumptions. Firstly, the validation set on the server has an even number of samples for each class and the class distribution matches the desired distribution for the given classification problem. Only then the class-wise precision and recall can be viewed as impactful parameters for determining the classification performance of the model. Secondly, the clients are willing to share metadata relevant to their class distribution and dataset size. Without that, clients can not be properly assessed in terms of their relevance to the learning objectives.

The proposed approach is fully illustrated in Fig. 1. On the server, a fuzzy inference system is designed to map precision and recall values of the class c to its learning demand $LD_c \in (0, 1)$. In the system, the input precision and recall values are first transformed into linguistic variables using fuzzy sets, as 3 intensity degrees: {LOW, MEDIUM, HIGH}. Membership degree functions are implemented as triangular functions, which are presented in Fig. 2.

The learning demand is defined over fuzzy sets and can take one of the following linguistic values: {VERY LOW, LOW, MEDIUM, HIGH, VERY HIGH}. The reasoning rules defined in the inference engine are a set of T-norms with fuzzified precision and recall values, presented in Tab. I. These rules aim to address common miss-classification errors, which can occur during training. R_1 and R_5 correspond to over- and under-saturation respectively. R_2 aims to address the issue of under-coverage, when the model recognises only a small amount of samples (even though correctly) as class c, with

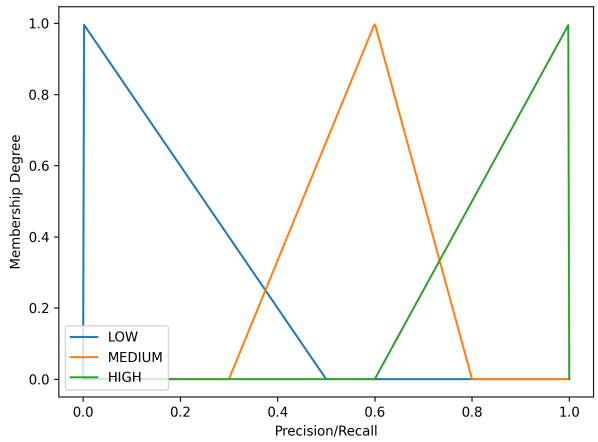

Fig. 2: Fuzzy membership functions for class-wise precision and recall

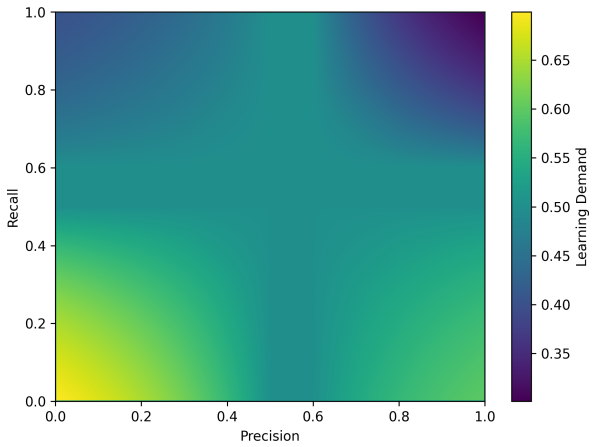

Fig. 3: Defuzzified Learning Demand Plane

TABLE I: Sugeno-type fuzzy rules

Rule number	T-Norm (Fuzzy AND) Precision	Recall	Learning Demand *linguistic*	*numerical*
R_1	HIGH	HIGH	VERY LOW	0.1
R_2	LOW	HIGH	LOW	0.3
R_3	MEDIUM	MEDIUM	MEDIUM	0.5
R_4	HIGH	LOW	HIGH	0.7
R_5	LOW	LOW	VERY HIGH	0.9

a great amount of false negative predictions. To prevent that, the model should 'see' more samples of class c, learning to correctly classify more difficult samples. On the other hand, high recall and low precision (R_4) can indicate the model's over-confidence. This can be corrected by subjecting the model to a larger number of samples from other classes, which will ultimately increase the level of certainty towards them. Hence, the learning demand for class c should be reduced, achieving exactly that. The R_3 smoothens the transitions between other learning demand states.

For each class, the learning demand is given as the weighted average of the consequent of each rule as:

$$\mathrm{LD}_c = \frac{\sum_{R_i} h_i \cdot z_i}{\sum_{R_i} h_i}, \tag{2}$$

where z_i is the consequent of rule R_i, in respect to membership value h_i of the antecedent. Given the constant output of the rules in the system and two input variables (precision and recall), the fuzzy inference system can be presented as a 2-argument function, yielding a continuous control signal representing the learning demand. The approximate plane of this function is presented in Fig. 3.

C. Client Relevance Modelling

With the estimated learning demands, each client chosen for the aggregation can then be assessed for their ability to fulfil them. Within the presented FL system, we especially consider non-IID cases, where clients have heterogenous class distributions and training set sizes. Therefore, to properly measure client k contribution to the system, the proportion of class c samples and sample quantity are used to calculated client's fitness. Each client k holds their class proportions:

$$P_{k,c} = \frac{n_{k,c}}{n_k}, \tag{3}$$

where $n_{k,c}$ and n_k denote the number of samples of class c and the total number of training samples on client k respectively. The relevance of client k to the current learning objectives is then defined as:

$$\mathrm{ClassScore}_k = \sum_{c=1}^{C} LD_c \cdot P_{k,c}. \tag{4}$$

The obtained class score measures how well the client's data aligns with the current learning policy, determined by the current learning demand. However, to account for varying dataset sizes, a size factor is introduced:

$$S_k = \sqrt{\frac{n_k}{\max(\{n_i\}_{i=1}^{K})}}, \tag{5}$$

where the denominator is the maximum dataset size among the clients selected for the aggregation. The final aggregation weight for the client k is computed as:

$$W_k = S_k \cdot \mathrm{ClassScore}_k. \tag{6}$$

The obtained client weights are then normalized across participating clients to maintain aggregation stability:

$$\tilde{W}_k = \frac{W_k + \epsilon}{\sum_i^K (W_i + \epsilon)}, \tag{7}$$

where $\epsilon \sim 0^+$ ensures numerical stability. Given client model parameters $\{\theta_k\}_{k=1}^K$ and normalized aggregation weights $\{\tilde{W}_k\}_{k=1}^K$, the global model is updated as:

$$\theta_G = \sum_{k=1}^{K} \tilde{W}_k \theta_k. \tag{8}$$

This procedure is repeated in each FL round, allowing the fuzzy weighting policy to dynamically adapt to the estimated learning demands of the system.

TABLE II: Data description

Dataset	Number of classes	training	testing
MNIST	10	60,000	10,000
EMNIST	47	112,800	18,800
CIFAR-10	10	50,000	10,000

III. Experiments

A. Data

The experiment was run on three benchmark image datasets, the details of which are presented in Tab. II. The MNIST [12] dataset contains $28 \times 28 \times 1$ images of hand-written digits, while the extended EMNIST [13] contains additional hand-written letters. Balanced version of the EMNIST was used, which contains even number of samples of all classes. Additionally, letters which are similarly written in lower- and upper-case were assigned the same class. Additionally, tests were also conducted on CIFAR [14] dataset, containing $32 \times 32 \times 3$ images depicting different objects (vehicles, animals, etc.).

As the datasets used have originally even class distributions, we introduced quantity and label distribution skew to simulate non-IID environment. To achieve this, we introduce data sampling based on Dirichlet distribution [15]. First, we sample the initial client sizes. Using the concentration parameter $\alpha_q = 0.5$, we define the initial dataset size for the client k as:

$$q \sim \text{Dir}(\alpha_q), \tag{9}$$

$$n_k = \lfloor q_k \cdot N \rfloor, \tag{10}$$

where N is the total number of training samples in the dataset. Next, using the dataset grouped by classes as:

$$\mathcal{D}_c = \{(x_i, y_i) \in \mathcal{D} : y_i = c\}, \quad N_c = |\mathcal{D}_c| \tag{11}$$

for each class $c \in \mathcal{C}$, client-specific proportions of are sampled using $\alpha_p = 0.5$ as:

$$p_c \sim \text{Dir}(\alpha_p), \tag{12}$$

and each client receives:

$$n_{k,c} = \lfloor p_{k,c} \cdot N_c \rfloor. \tag{13}$$

To prevent insufficient data splits, we enforced quantity constrains to maintain a minimum number of samples per client. In the experiments, the minim number of training samples per client was set to 100. After data splitting, each client randomly divided its portion of the dataset into their private training and validation datasets in an 80:20 ratio. In the experiments, we conducted tests on both the homogenous and heterogenous data distributions. The example splits on the MNIST data and 10 clients (with $\alpha_q, \alpha_p = 0.5$) are presented in Fig. 4-5. 20% of the test set was taken to create the server's validation set, with homogenous class distribution preserved.

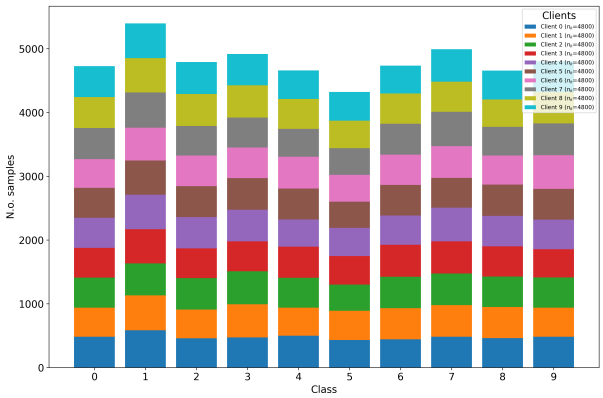

Fig. 4: Example IID training data distribution of MNIST dataset across 10 clients

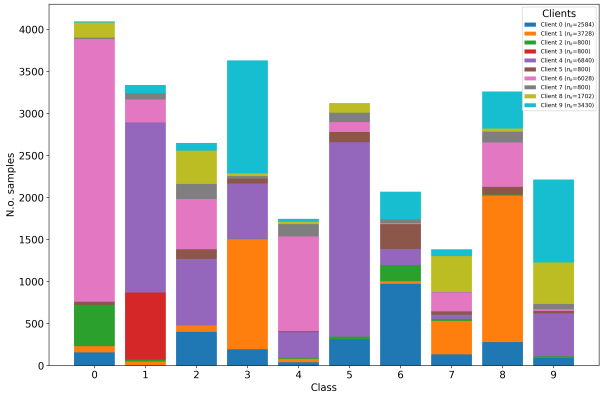

Fig. 5: Example non-IID training data distribution of MNIST dataset across 10 clients

B. Experimental settings

The experiments were conducted using the proposed approach against standard accuracy-weighted FedAvg. Two FL system sizes were considered: one involving 100 clients and other with 1000 clients total. In both cases, the aggregation process involved only a portion of the clients, chosen by the reported highest validation accuracy scores on their private sets. The proportion of clients chosen for the aggregation was respectively 3%, 10% and 30%.

The chosen classifier was a shallow CNN, with 2 consecutive convolutional layers with 32 and 64 neurons respectively, with (3×3) kernels and ReLU activation function. Following, was the max pooling layer with 2×2 window and a dense layer with 128 neurons and ReLU. The number of neurons in the output dense layer corresponded to the number of classes in the considered dataset.

In each case, the training lasted for a total of 1000 FL rounds, with 1 epoch per client with 64 samples per batch. During whole learning process, each client used priorly initialized ADAM optimizer and cross-entropy loss.

C. Results

In Tab. III, the accuracy scores from the non-IID settings are presented. The proposed method is compared against the base

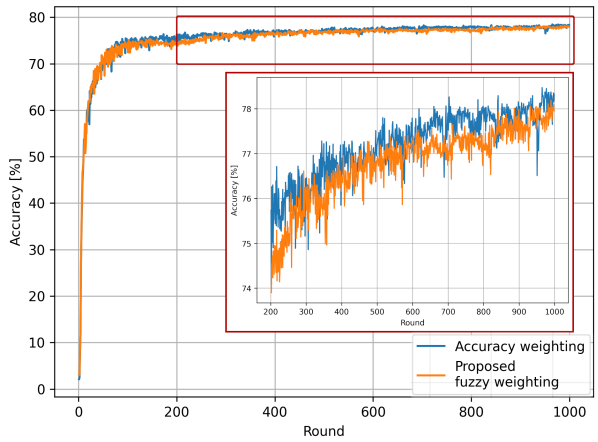

Fig. 6: Test accuracy during FL rounds on non-IID EMNIST with 100 clients

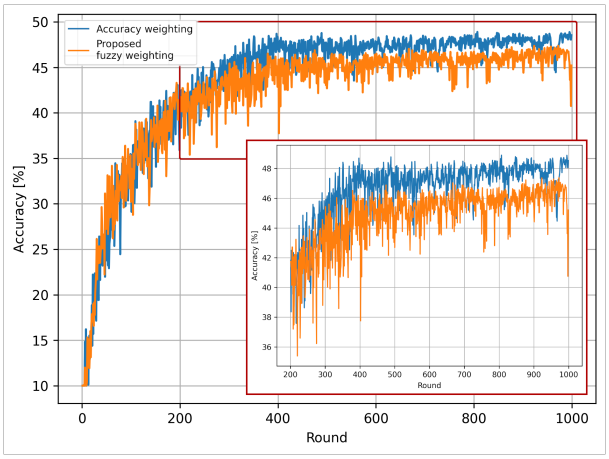

Fig. 7: Test accuracy during FL rounds on non-IID CIFAR-10 with 100 clients

method, which was FedAvg weighted by validation accuracy. The results indicate, that the fuzzy-weighted method improved end accuracy when smaller number of aggregators where considered, and the difference faded as the aggregator percentage increased. The exception was the MNIST dataset, where the accuracy scores did not differ more than 0.3 percentage points in any case.

However, the clear advantage of the proposed technique can be observed in the training curves. In Fig. 6-7, test accuracy over the course of FL rounds. In both cases, the proposed fuzzified weighting results in a higher average accuracy score compared to weighting driven by the reported validation accuracy. This demonstrates the adaptability of the proposed method, compared to the baseline approach.

Experiments on IID setting were also conducted. It is worth pointing out, that in this scenario, as each client had equal dataset sizes and class distribution, the proposed algorithm was practically no different than the unweighted FedAvg. This effect is desired, as in this scenario each client has equal potential for training the global model. Furthermore, in the

TABLE III: Experimental results on non-IID setting with 100 clients

Aggregators percentage	Dataset MNIST	EMNIST	CIFAR10
	Accuracy-weighted FedAvg		
3%	94.26	74.75	40.87
10%	95.37	78.43	45.79
30%	96.68	80.45	49.88
	Proposed fuzzy-weighted FedAvg		
3%	93.94	75.41	42.69
10%	95.55	78.47	48.40
30%	96.81	80.29	48.11

TABLE IV: Experimental results on the IID setting with 100 clients

Aggregators percentage	Dataset MNIST	EMNIST	CIFAR10
	Accuracy-weighted FedAvg		
3%	98.28	80.21	52.61
10%	97.16	81.32	54.18
30%	97.36	83.64	59.65
	Proposed fuzzy-weighted FedAvg		
3%	98.01	79.69	52.45
10%	97.08	82.13	54.77
30%	97.42	83.72	59.39

IID environment, the client subjective validation score is much more consistent with global data. The test results are presented in Tab. IV. The results indicate, that there was practically no difference in accuracy scores. This result was expected, as the training environment (with homogenous data) was stable, and all clients were contributing to the system practically uniformly.

D. Discussion

The experiments were primarily conducted in a heavily non-IID environment, where class imbalance was the main factor for consideration. On top of that, the total of 100 clients participating in FL resulted in small local training sets, which were already imbalanced. This setup imitates environment, where individual clients have very limited generalization capabilities. Because of that, only a percentage of clients were chosen for the aggregation process in each round. The selection was based on the clients' reported accuracy scores on their private validation sets, which were also biased. This created a challenging environment for developing a dependable global model. Each experiment was run for 1000 FL rounds, providing enough time for the system to achieve stable results. It was observed that in the non-IID scenario, on average, the proposed fuzzy-driven weighting approach results in higher accuracy than the accuracy-driven approach. However, the final test results, with the models' best performance, were not that definitive. First thing observed was, that with the increase of the number of aggregators, the accuracy increased. Furthermore, the proposed method showed the largest performance increase, when fewer aggregators were involved, and the difference faded as their number was increased. The only exception was the 3%-aggregator setting on MNIST, where the proposed method

scored worse. However, this can be explained by the simplicity of the MNIST dataset, compared to the other two (EMNIST, considering the larger number of classes and CIFAR10, whose classification problem is more complex). Nevertheless, the obtained results show potential, especially in scenarios, where the number is limited and the client's importance should be carefully considered.

IV. Conclusion

In the presented work, fuzzy inference system was introduced to define class-wise learning demands of the FL system. In each round, new learning demands were calculated based on precision and recall metrics, and the aggregation clients were weighted based on these demands and their private resources. The proposed solution adaptively and intuitively adjusts learning policy, which is crucial when data in the FL system is strongly heterogenous. Unlike data-driven or opaque scheduling mechanisms, the use of Sugeno-type fuzzy inference provided a transparent and interpretable decision-making, focused on even multi-class classification performance. The experimental results on MNIST, EMNIST, and CIFAR-10 datasets demonstrate that the proposed method effectively improves global model performance under non-IID conditions, highlighting fuzzy logic as a viable and explainable tool for adaptive federated learning aggregation.

Acknowledgment

This work is supported by the Silesian University of Technology under Grants No. 09/020/SDU/10-21-01 and by the Polish Ministry of Science and Higher Education under project no. MNiSW/2025/DPI/650 "Intelligent mechanisms of selective information processing in deep neural networks" as part of "Supporting students to improve their competencies and skills" program.

References

[1] A. Aminifar, J. Dan, and D. Atienza, "Federated learning with patient-annotated data in epileptic seizure detection," in *2025 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2025, pp. 1–10.

[2] P. H. Barros, T. Polido, J. C. Guevara, L. Villas, D. Guidoni, N. L. Da Fonseca, and H. S. Ramos, "Personalized federated learning for sedentary behavior classification with heterogeneous feature distributions under adversarial threats," in *2025 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2025, pp. 1–8.

[3] C. Chen, T. Liao, X. Deng, Z. Wu, S. Huang, and Z. Zheng, "Advances in robust federated learning: A survey with heterogeneity considerations," *IEEE Transactions on Big Data*, 2025.

[4] X. Fang, M. Ye, and B. Du, "Robust asymmetric heterogeneous federated learning with corrupted clients," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[5] W. Jiang, Y. Zhang, H. Han, X. Liu, J. Gwak, W. Gu, A. Shankar, and C. Maple, "Fuzzy ensemble-based federated learning for eeg-based emotion recognition in internet of medical things," *Journal of Industrial Information Integration*, vol. 44, p. 100789, 2025.

[6] F. Sabah, Y. Chen, Z. Yang, A. Raheem, M. Azam, N. Ahmad, and R. Sarwar, "Fairdpfl-scs: Fair dynamic personalized federated learning with strategic client selection for improved accuracy and fairness," *Information Fusion*, vol. 115, p. 102756, 2025.

[7] C.-F. Lai, Y.-H. Lai, M.-C. Kao, and M.-Y. Chen, "Enhancing global model accuracy in federated learning with deep neuro-fuzzy clustering cyclic algorithm," *Alexandria Engineering Journal*, vol. 112, pp. 474–486, 2025.

[8] Y. Ding, W. Zhao, and K. Huang, "Fedmwad: Module-wise weighted aggregation federated learning combined with ditto for patient-independent seizure prediction," *Information Processing & Management*, vol. 62, no. 6, p. 104253, 2025.

[9] Y. He and J. Yu, "Towards secure weighted aggregation for privacy-preserving federated learning," *IEEE Transactions on Information Forensics and Security*, 2025.

[10] C. Shi, H. Zhao, B. Zhang, M. Zhou, D. Guo, and Y. Chang, "Fedawa: Adaptive optimization of aggregation weights in federated learning using client vectors," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 30 651–30 660.

[11] 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*. PMLR, 2017, pp. 1273–1282.

[12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[13] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, "Emnist: Extending mnist to handwritten letters," in *2017 international joint conference on neural networks (IJCNN)*. IEEE, 2017, pp. 2921–2926.

[14] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[15] D. M. J. Gutierrez, A. Anagnostopoulos, I. Chatzigiannakis, and A. Vitaletti, "Fedartml: A tool to facilitate the generation of non-iid datasets in a controlled way to support federated learning research," *IEEE Access*, vol. 12, pp. 81 004–81 016, 2024.