Advances in Science and TecŠology Research Journal

Advances in Science and Technology Research Journal, 2026, 20(1), 89–103
https://doi.org/10.12913/22998624/210058
ISSN 2299-8624, License CC-BY 4.0

Received: 2025.07.08
Accepted: 2025.09.28
Published: 2025.11.21

Smooth ordered weighted averaging in federated learning: A Newton-Cotes quadrature-inspired aggregation framework

Alicja Rachwał[1*], Albert Rachwał[1], Paweł Karczmarek[1]

[1] Department of Computational Intelligence, Lublin University of Technology, Nadbystrzycka 38D, 20-618 Lublin, Poland
* Corresponding author's e-mail: alicja.rachwal@pollub.pl

ABSTRACT

This paper presents a novel approach to federated learning based on the smooth ordered weighted averaging (OWA) operator which enables flexible and context-sensitive weighting of local models during the aggregation process. To enhance the quality of the aggregated weight computations, we incorporate numerical quadrature-inspired techniques, allowing for a more accurate representation of individual client contributions to the global model. Specifically, the approach utilizes classical OWA and several smoothed variants derived from Newton-Cotes quadratures, including the 3/8 rule, trapezoidal rule, and ONC4 (4-point open Newton-Cotes) formula. The study compares federated learning models using standard weight averaging against those incorporating both classical and smoothed OWA operators. This evaluation provides insight into how the smoothing mechanisms influence aggregation quality and final model accuracy. A neural network comprising several dense layers served as the classification model in the federated learning framework. Two experimental scenarios were considered: one where data was evenly distributed across local clients, and another with non-uniform data distribution to reflect real-world heterogeneity. Various strategies for extracting the OWA weights were explored, including performance-based weighting determined by the accuracy of local models during preliminary training rounds The proposed methodology has been tested on small-scale image datasets such as MNIST and it has demonstrated improved classification accuracy value compared to traditional federated learning approaches using simple averaging.

Keywords: federated learning, neural network, OWA operator, smooth OWA, weight aggregation.

INTRODUCTION

Federated learning is a more and more popular approach to distributed machine learning that enables multiple clients to collaboratively train a shared model without exchanging their raw data. This paradigm is particularly relevant in contexts where privacy, security or legal constraints limit centralized data collection. It also reduces the need for large-scale data transfers, which is beneficial in scenarios with limited bandwidth or distributed data sources, such as mobile devices or Internet of Things networks. Instead of aggregating data in one place, federated learning relies on the exchange of model updates, which are then combined to form a global model. A key step in this process is the aggregation mechanism, which determines how local updates are merged. As in many other areas of machine learning, recent years have seen numerous modifications to the Federated learning algorithm. The most classic and fundamental model is FedAvg [1] which averages updates from local models to create a global model. However, it assumes that data is independently and identically distributed (IID) among clients. This assumption is often not met in real-world applications which creates a strong need to improve FedAvg for heterogeneous data. FedProx [2] introduces a proximal term to address objective inconsistency between local and global optimization. FedNova [3] normalizes client updates to counteract unbalanced contributions in heterogeneous settings. Using a mechanism based on deep Q-learning, the Favor algorithm [4] selects client devices from past experience to participate in each round of federated learning, counterbalancing bias introduced by Non-IID data and thus accelerating convergence.

Due to the privacy-preserving nature of federated learning, despite using information from multiple client devices, the algorithm is particularly suitable for applications requiring a high level of data security, such as finance or healthcare. [5] provides an example of using federated learning in medicine for collaboration between multiple healthcare institutions without sharing patient data. Other works also highlight the need for algorithms that ensure patient data anonymity while still allowing for model training. [6] and [7] explore similar topics, including the detection of heart-related diseases and hospitalizations, while preserving patient privacy. Because of its decentralized structure, FL is also well-suited for applications in the internet of things and mobile devices. In [8], an interesting combination of medical application with mobile and IoT devices is presented. FL and transfer learning are applied to health data collected from wearable devices such as smartbands. Federated learning can also be successfully applied to modeling from mobile devices such as in mobile keyboard prediction [9].

Often combined with federated learning is image analysis, such as classification, using various neural network structures. Image analysis remains one of the most intensively developed areas within the broader field of artificial intelligence, with applications ranging from medical diagnostics and remote sensing to biometric systems and industrial automation. The paper [10] explores the application of deep convolutional neural networks to the classification of media images from individual sports, addressing the challenge of visual similarity across disciplines. Their study demonstrates how tailored deep learning architectures can effectively differentiate between subtle patterns in sports imagery, contributing to the refinement of automated recognition systems in this domain. A convolutional neural network-based approach for automated detection of bone fractures in X-ray images is presented in [11], addressing a critical challenge in medical image analysis. This research highlights the capacity of deep learning models to capture subtle diagnostic features, supporting the development of more efficient and accurate tools for clinical decision-making. The study [12] investigates the effectiveness of local image descriptors in face recognition systems under age-related variations. By evaluating several commonly used descriptors on the FG-NET aging database, this research analyzes their robustness across age groups and explores how their performance changes when paired with different similarity measures and Gabor wavelet representations. The paper [13] investigates the effectiveness of ensemble learning techniques in enhancing the classification of brain tumors from MRI images using convolutional neural networks. By integrating multiple pretrained models within a transfer learning framework, their approach demonstrates improved accuracy and robustness compared to single-model baselines, offering promising results for clinical image analysis applications. In [14], a hybrid approach is proposed, combining convolutional neural networks with traditional machine learning classifiers for the detection of clustered fruits, using grapes as a case study. By leveraging deep feature extraction from multiple CNN architectures and integrating it with support vector machines, the presented method demonstrates high precision, supporting advanced applications in agricultural automation and yield estimation. The paper [15] presents a content-based image retrieval method that identifies visual objects by following their edges—a technique referred to as edge crawling. The detected object shapes are then described using histograms of local features and angular measurements, allowing for fast retrieval of visually similar images. The approach, evaluated on several benchmark datasets, demonstrates competitive performance without relying on deep learning architectures. Another example of novel neural architecture design is the Weighted Probabilistic Neural Network presented in [16], introducing sensitivity-derived weights into the traditional Probabilistic Neural Network model. Although not focused on visual data, its strong performance across multiple benchmark classification tasks offers a compelling case for its future adaptation in federated or image-based settings where interpretability and adaptability are critical.

A frequently explored topic in research is the weighting of local models. One approach is to assess the influence of local models on the global model. Simple yet effective methods were proposed in [17] where contributions were calculated using the deletion method and the Shapley Index. The work in [18] identifies one of the issues with simple averaging in FedAvg: Conflicting gradients with large differences in magnitude.

A fair averaging framework, FedFV, is proposed, with an algorithm designed to eliminate potential conflicts in gradients between clients. There are also works proposing the use of aggregation operators other than arithmetic or weighted averages. In [19], a Choquet-based aggregation is employed, which in the future could also be improved by Choquet integral modifications [20]. Even with a simple weighted mean [21] there are many ways to define the weights, aiming to produce the most effective global model. Another suggestion is adaptive weighting based on criteria such as Inverse Distance [22]. Many concepts used in federated learning are inspired by methods used in ensemble learning. It is worth noting that although federated learning and ensemble learning both involve combining multiple models, they are conceptually and architecturally distinct. Ensemble learning typically trains multiple models independently, often on the same dataset, and combines their predictions to improve generalization. In contrast, federated learning trains local models on disjoint, decentralized data and aggregates their parameters or updates to form a single global model. The aggregation in federated learning serves to synchronize model knowledge across clients, rather than to directly combine predictions from multiple models.

Federated learning remains a relatively new and actively evolving algorithm. The uneven distribution of data among clients continues to pose a challenge, and the choice of aggregation weights—frequently discussed in various studies—is a crucial component of the framework. Furthermore, despite some attempts to use alternative aggregation strategies, most federated learning modifications still rely on simple or weighted averaging. It is possible that employing different aggregation operators could lead to a more effective modeling of client influence on the global model.

In this study, we aim to propose a novel modification to the classical FedAvg algorithm by introducing a recently developed aggregation operator: Smooth OWA [26], which is a smoothed version of the well-known OWA operator. These operators provide a flexible and interpretable mechanism for model combination, enabling the aggregation process to dynamically adapt to the relative performance of client models. As a result, the aggregation should allow for a more nuanced consideration of local model contributions to the global model. Through the use of smoothing, we can subtly account for interactions between models, since each coefficient is smoothed using neighboring coefficients from other local models. Such OWA-based aggregation is a promising alternative to the traditional aggregation methods. A measurable outcome of this modification should be an improvement in the accuracy of the resulting global model.

The paper is structured as follows: the Theoretical Background section recalls the fundamental concepts of federated learning as well as the definitions of the OWA and Smooth OWA operators. The proposed methodology section introduces our novel modification to the FedAvg algorithm. The numerical experiments section presents the results of our computations using the proposed method, along with a discussion. Finally, in the Conclusions and Future Work section, a summary of our findings and outlines directions for future work are given.

BACKGROUND

This section will recall the basic concepts of federated learning, followed by a description of the OWA operator and its recent modification called Smooth OWA.

Federated learning

Federated learning is a machine learning technique used in an environment where multiple entities train a model together, while keeping the data decentralized rather than stored centrally. Instead of transferring data to a central common space where calculations take place, local model coefficients are transferred to the global model. In order to get a single global model, one needs to combine all the model updates we have received from client nodes. This process is called aggregation, and there are many different ways to do it. The most basic is federated averaging [1], often abbreviated as FedAvg. In practice, it often uses a weighted average where the weights are the data sizes of each client.The federated learning framework can be described as follows [23]. Let us assume that there are n clients (local models) $\{M_1, M_2,..., M_n\}$. Each is assigned its own portion of the dataset $\{D_1, D_2,...,D_n\}$. Client C_j has access only to the assigned part of the set D_j. Global model M_G does not have access to any training data. A communication round (a training epoch) proceeds in the following way:

1. Global model M_G transmits its coefficients V^G to local models, i.e. the local models are initialized anew, with the current global coefficients.
2. For each $j=1, 2,..., n$ the model M_j is trained on the D_j part of the dataset. The coefficients V^{Mj} of the local model are updated.
3. The aggregation of local coeficients V^{Mj} occurs in the global model, producing new values of V^G coefficients.

This process is presented in Figure 1. The training is repeated a predetermined number of epochs or until a specified stop condition is met.

The problem of data distribution among clients is closely related to the issue of federated learning. For the testing process, it is often assumed that the data is independently and identically distributed among clients (Independent and Identically Distributed – IID). In practice, this is usually not the case, rather, clients have different amounts of data, and often qualitatively different, using the example of classification: If we have data of 10 classes, it may be that, for example, the first client has to deal with only two of them, another only with three, etc. We refer to such a situation as Non-IID. Partitioning data as Non-IID poses a challenge for models. Researchers often tackle this problem [24].

OWA and smooth OWA operators

Among the many existing aggregation methods, a family of ordered weighted averaging (OWA) operators stands out as flexible and very adaptable. In contrast to weighted average, the OWA operator applies weights not to specific components, but to the ordered position of them. The OWA operator [25] is defined as function OWA_w: $\mathbb{R}^n \to \mathbb{R}$ associated with a set of weights $w = [w_1, w_2,...,w_n]$ such that $w_1 \geq w_2 \geq ... \geq w_n$ and $\sum_{i=1}^{n} w_i = 1$, determined by the formula

$$OWA_w(x_1, x_2, \dots, x_n) = \sum_{i=1}^{n} w_i \cdot x_{(i)}, \qquad (1)$$

where: $x_{(I)}$ is the i-th largest value in the vector $(x_1, x_2,..., x_n) \in \mathbb{R}^n$.

Smooth OWA operator [26] is a modification of the OWA operator, associated additionally with smoothing method denoted as Q:

$$Smooth\ OWA_{w,Q}(x_1, x_2, \dots, x_n) = \sum^{n} w_i \cdot Q(x_{(i)}), \qquad (2)$$

where: $Q(x_{(i)})$ means an application of a chosen Newton-Cotes formula to the element $x_{(i)}$.

Let us recall few Newton-Cotes quadratures that can be used for such smoothing:

$$Q_{\frac{3}{8}}(x_{(i)}) = \frac{1}{8} \times_{(i-1)} + \frac{3}{8} \times_{(i)} + \frac{3}{8} \times_{(i+1)} + \frac{1}{8} \times_{(i+2)} \ ({}^{3}/_{8}\ quadrature) \qquad (3)$$

$$Q_T(x_{(i)}) = \frac{1}{2} \times_{(i)} + \frac{1}{2} \times_{(i+1)} \ (trapezoidal\ quadrature) \qquad (4)$$

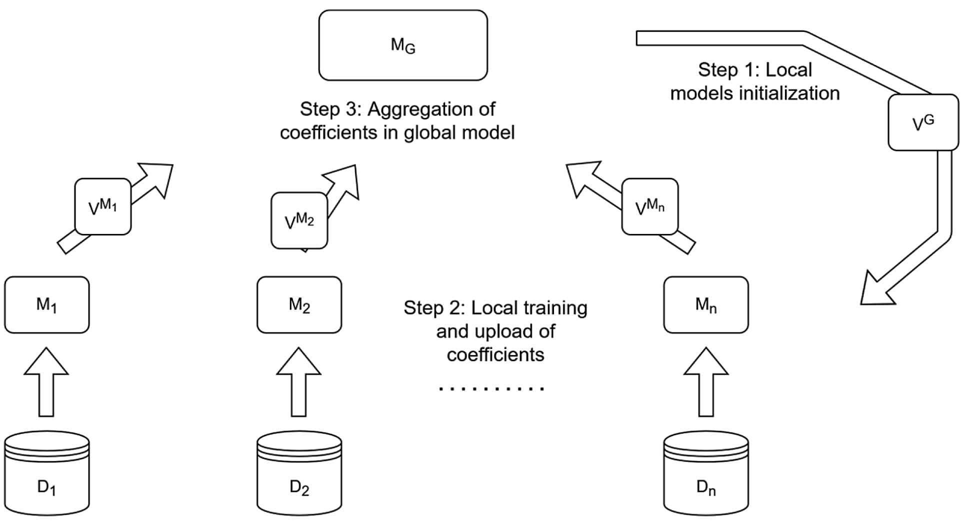

Figure 1. Federated learning general scheme

$$Q_{ONC4}(x_{(i)}) = \frac{11}{24} \times_{(i-1)} + \frac{1}{24} \times_{(i)} + \frac{1}{24} \times_{(i+1)} + \frac{11}{24} \times_{(i+2)} \quad (4-point\ Open\ NC\ quadrature) \tag{5}$$

With such operation of applying the Newton-Cotes quadratures, each element $x_{(I)}$ is smoothed by its neighboring elements in the vector of sorted input. Note that if the index of an element is less than 1, we take the value of $x_{(I)}$ instead of that element, and if the index is greater than n, we take $x_{(n)}$.

PROPOSED METHODOLOGY

The process of federated learning with smooth OWA aggregation is presented in Figure 2. Local model 1,... Local model n represent n clients in the federated learning scheme. For each client, a local model is trained and its coefficients are calculated (more accurately, these are the weights in the neural network, but the name 'weights' can be misleading because they are also used in this work in the context of aggregation operators). Each local model sends its parameters to the global model, and gets back the aggregated coefficients, that is, the coefficients updated in the global model based on information from all n clients. The global model is the central model that collects all local coefficients from clients and aggregates them. In the basic method, this is done using an average. In another existing modification, it is the classic OWA operator. In our work we propose to use the smooth OWA operator with a set of weights w and a selected smoothing method Q.

Smooth OWA is an aggregation function that sorts the input data (here: the weights of neurons from different clients), smooths it by a combination of neighboring values (e.g., by Newton-Cotes rules), and applies the defined weights w to the aggregation. Q is a type of smoothing rule – e.g., Trapezoidal, Three-eighths – that affects how strong the smoothing of values is during aggregation.

A more detailed diagram of the process is shown in Figure 3. At the beginning of each round of communication (training epoch), each of the n local models receives a copy of the weights from the global model – this is the classic starting point of a federated learning round. Each client performs one local learning epoch on its set and updates all the coefficients of its model (weights in the neural network), denoted as

$$\{V_1, V_2, \ldots, V_k\} \tag{6}$$

Therefore every coefficient V_j, $j = 1,2,..., k$ has n versions – one for each client. For each position j all values of coefficient V_j are collected from the client models:

$$\{V_{j,local\ 1}, V_{j,local\ 2}, \ldots, V_{j,local\ n}\} \tag{7}$$

Then these values for each k separately are sorted in descending order:

$$\{V_{j,(1)}, V_{j,(2)}, \ldots, V_{j,(n)}\} \tag{8}$$

The sorted values are processed by the selected quadrature Q, resulting in the following set of smoothed coefficients.

$$\{Q(V_{j,(1)}), Q(V_{j,(2)}), \ldots, Q(V_{j,(n)})\} \tag{9}$$

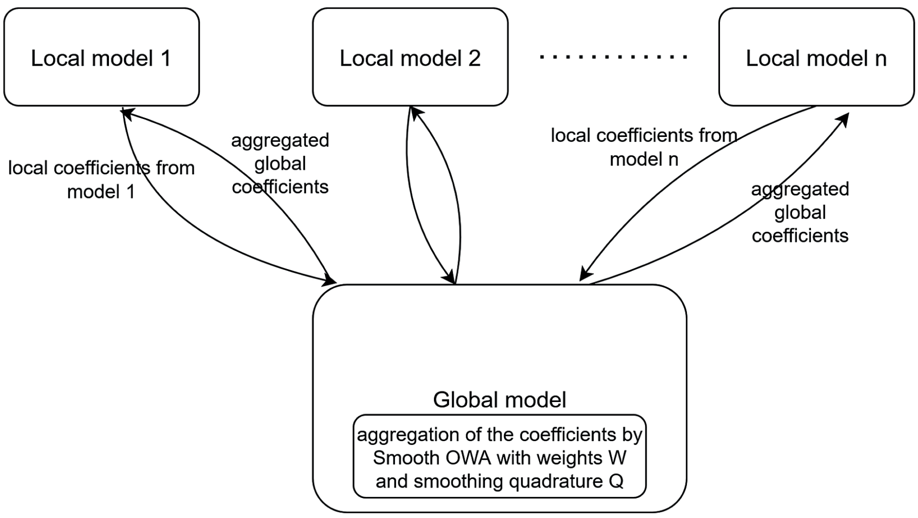

Figure 2. The process of federated learning with smooth OWA aggregation

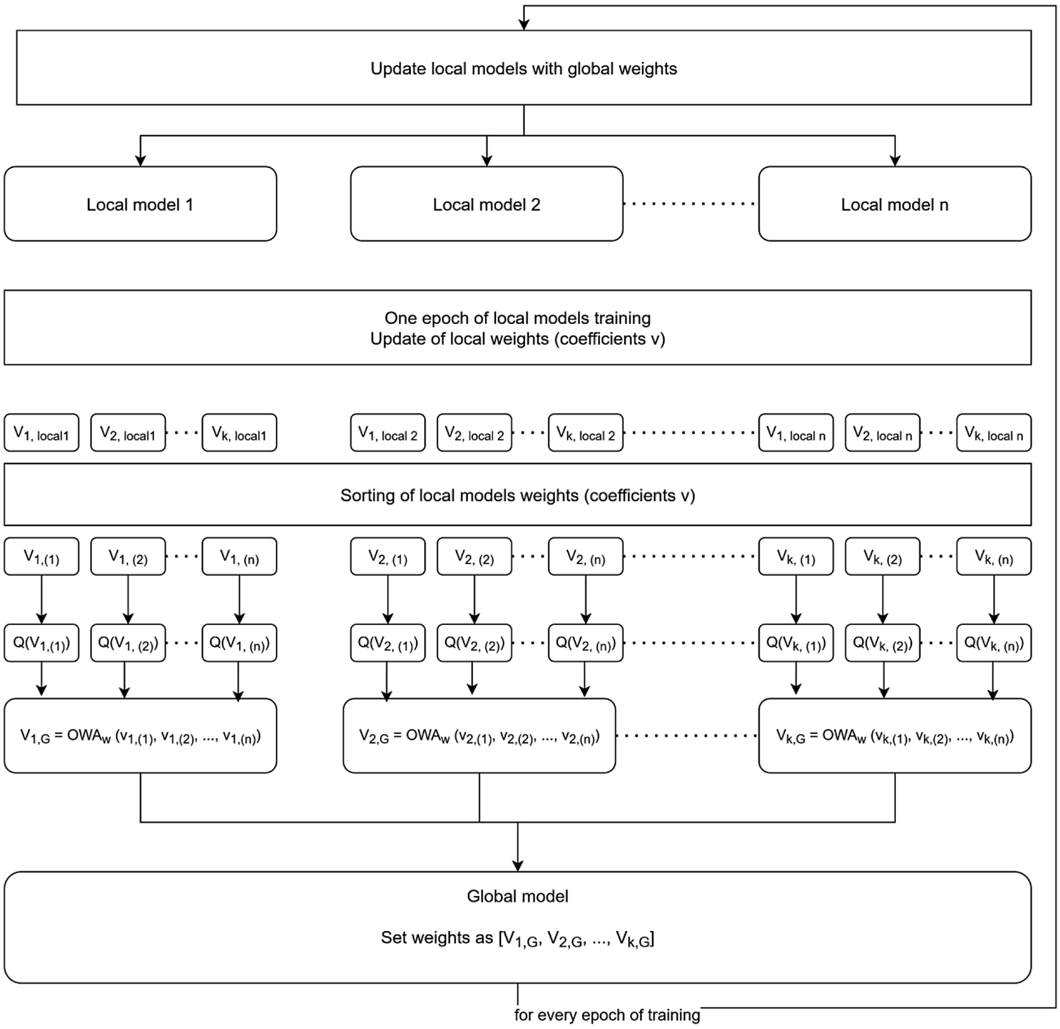

Figure 3. Detailed workflow of federated learning with smooth OWA aggregation

After the smoothing process, the resulting coefficients are aggregated as in the classical OWA operator with weights *w*. Thus the final coefficients of the global model have the form

$$V_j^G = OWA_w(Q(V_{j,(1)}), Q(V_{j,(2)}), \dots, Q(V_{j,(n)})) \quad (10)$$

At the end of the learning epoch, the global model is characterized by a set of coefficients

$$[V_1^G, V_2^G, \dots, V_k^G] \quad (11)$$

and such weights are propagated to local models at the beginning of the next communication round.

NUMERICAL EXPERIMENTS

This section includes a description of the datasets, the neural network model used for classification, methods used to extract weights for the OWA operator, and a presentation and discussion of the results obtained from the numerical experiments.

Datasets descprition

Three widely known datasets containing small scale images were used for the experiments. MNIST is a dataset of 60 000 28 × 28 grayscale images of the 10 digits, along with a test set of 10 000 images. Fashion-MNIST is a similar dataset containing 60,000 28 × 28 grayscale images of 10 fashion categories, along with a test set of 10 000 images. CIFAR10 is a dataset of 50 000 32 × 32 color training images and 10 000 test images, labeled over 10 categories. The collections were loaded from the Tensorflow library resources in Python. For each set, a merge of the training and test sets was performed, due to the fact that in each iteration of the experiment a different division between the learning and test parts was planned, in order to make the numerical results more reliable. The size of the sets after combining the training and testing parts is shown in Table 1, along with the number of classes and whether the images are grayscale or color.

In each iteration of the experiment, the entire dataset is split into a learning and testing part, and then the training set is divided among 10 clients, each client (local classification model) having access only to its part of the set. For the IID split, the data is first shuffled and thus each client gets random observations from different classes. For Non-IID splitting, the data is sorted by class label before splitting, and so clients usually get observations only from 2–3 classes. This is a more difficult task for the classification model, but more similar to the real-world case of federated learning. The global model does not have access to the data, but only gets information (coefficients) from local models. At the end of the learning epochs, the global model is validated on the test part of the dataset.

Classification model

A simple neural network with dense layers was used for classification in the federated learning model. The overall processing scheme of the network is presented in Figure 4. The first two layers contain 200 neurons each and a relu activation function. The output layer consists of 10 neurons (since there were 10 classes in each dataset) and a softmax activation function. The input image was flattened to a one-dimensional vector of appropriate size before being passed to the neural network. A categorical crossentropy loss function and accuracy metric were used in the training process.

Experimental results

The proposed method was compared with federated learning with the arithmetic mean and federated learning with the classical OWA operator. 10 iterations of experiments were performed on each of the three data sets. Each iteration included IID and Non-IID scenarios of data division. Each iteration of the experiment consisted of 30 communication epochs where local models were trained and the weights of the global model were updated. The classification accuracy from the global model was tested on the test set.

Several methods of selecting weights have been proposed:

- Without considering any information from the dataset – as a decreasing sequence of inverses of consecutive natural numbers: [1/2, 1/3, 1/4,..., 1/11]
- Based on the accuracy of local classifiers in five rounds of pre-training – a vector of accuracies is assembled, next it is sorted in decreasing order and normalized to sum equal to 1.
- On the basis of the size of the client sets – [0.23, 0.18, 0.15, 0.12, 0.09, 0.08, 0.06, 0.045, 0.035, 0.03].

Table 1. Dataset information

Dataset	Number of observations	Number of classes	Number of color channels
MNIST	70 000	10	1 (grayscale)
Fashion-MNIST	70 000	10	1 (grayscale)
CIFAR10	60 000	10	3 (color)

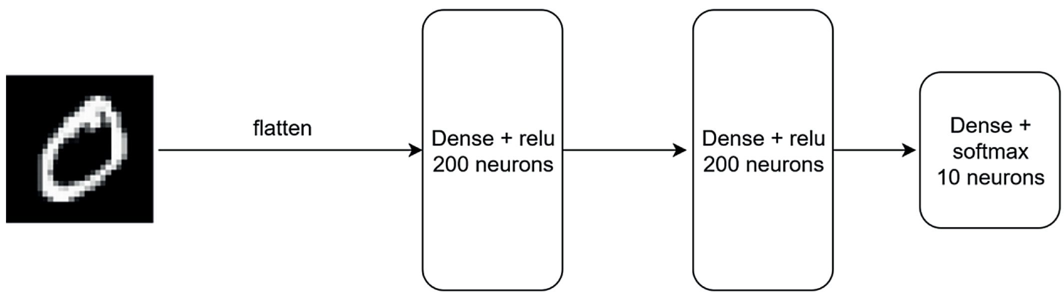

Figure 4. Scheme of simple neural network used for federated learning classification

Prior to aggregation, the weights were normalized so that their sum was equal 1. The results obtained with those sets of weights that best suited the particular set under study will be presented.

For the MNIST dataset, the best results were obtained while using the set of weights derived from accuracy of local classifiers in pre-training rounds. These results are presented in Table 2 for IID scenario and in Table 3 for Non-IID.

The average accuracy on the MNIST set for the IID case in the base method was 93.45%. Aggregation with the classical OWA operator yielded a similar result. With the help of smoothing of the OWA operator, this result was slightly increased to around 93.51–93.53%, so less than 0.1 percentage points. This result was not statistically significant in the Wilcoxon test.

When partitioning the MNIST dataset Non-IID, the differences between the methods used were much greater. For the baseline method, the average accuracy was 77.32%, and the range of it was from about 76 to about 80%. The classical OWA operator achieved a slightly worse mean result of 77.25%. Smooth operators were able to achieve better results. Smoothing with the trapezoidal quadrature yielded a statistically insignificant 0.17 percentage point improvement in the score. For the 3/8 quadrature, the improvement was significant and equal about 1 percentage point. Higher scores can also be observed based on the minimum and maximum for this method: from 76.7 to 81.3%. The biggest difference in accuracy in favor of the smooth OWA operator appeared with the ONC4 quadrature. The average accuracy reached 80.11%, an improvement of about 2.8 percentage point over the baseline method. The accuracy range for the ONC4 method was 78.7 to 82.5%. The standard deviation for the ONC4 method was lower than for the other methods which is also an advantage because it means that the method behaved more stably regardless of the exact split into the learning and test sets.

The graphs in Figure 5 and Figure 6 depict the average accuracy values at successive epochs of training on the MNIST set for the Non-IID data distribution. In the case of the ONC4 quadrature (Figure 5), it can be seen that in the first few epochs the difference in favor of this method is relatively small, and by the 10th epoch it becomes more pronounced and remains so until the end. In the case of the 3/8 quadrature (Figure 6), the course of accuracy values is almost perfectly similar to the baseline method up to about epoch 15, and then the 3/8 method begins to show an improvement over the baseline method – small at first, and by the end of learning somewhat more pronounced although much smaller than in the case of ONC4 quadrature. The fashion-MNIST image collection is a very similar set to MNIST. The images are also grayscale and the same size, but they differ in the subject presented. It turns out that for the

Table 2. Accuracy on MNIST dataset for IID scenario, with weight set based on local models' accuracy in pre-training

Method	Min	Mean	Median	Std	Max	Wilcox. Is better than avg	Wilcox. Is better than OWA
Average	93.1357	93.4521	93.4250	0.3006	94.0357	-	-
Base OWA	93.1286	93.4486	93.4107	0.3025	94.0286	No	-
OWA 3/8	93.1857	93.5264	93.5107	0.2969	94.1357	No	No
OWA trap.	93.1857	93.5236	93.5107	0.2959	94.1286	No	No
OWA ONC4	93.1857	93.5143	93.4929	0.2927	94.1143	No	No

Table 3. Accuracy on MNIST dataset for Non-IID scenario, with weight set based on local models' accuracy in pre-training

Method	Min	Mean	Median	Std	Max	Wilcox. Is better than avg	Wilcox. Is better than OWA
Average	75.9571	77.3200	77.2536	1.1482	80.0143	-	-
Base OWA	0.758786	0.772471	0.771821	1.1561	79.9500	no	-
OWA 3/8	0.767286	0.783514	0.782107	1.3154	81.3214	yes	yes
OWA trap.	0.760214	0.774907	0.773071	1.4045	80.7000	no	no
OWA ONC4	0.786929	0.801071	0.799821	1.0856	82.5071	yes	yes

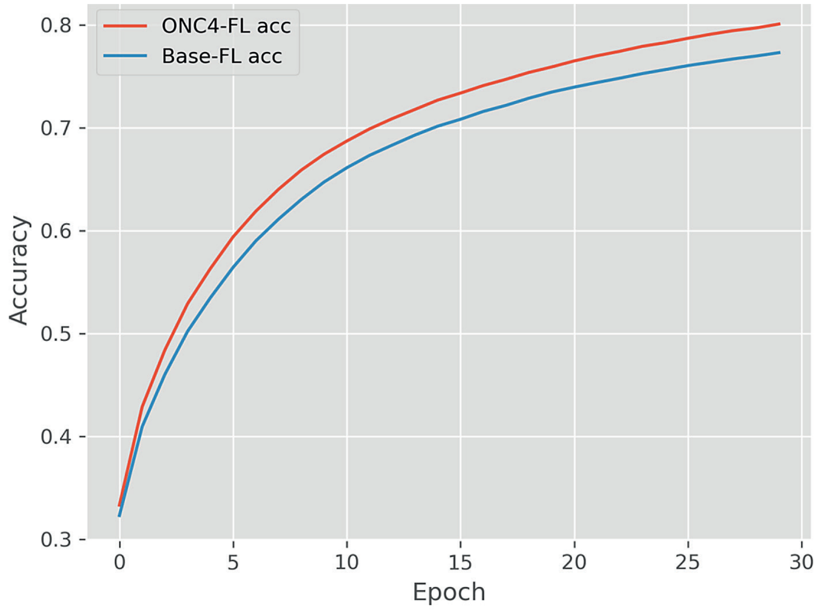

Figure 5. Average accuracy in subsequent epochs on test sets from MNIST dataset for non-IID scenario, with weight set based on local models' accuracy in pre-training, smooth OWA with ONC4 quadrature compared to base federated learning with averaging

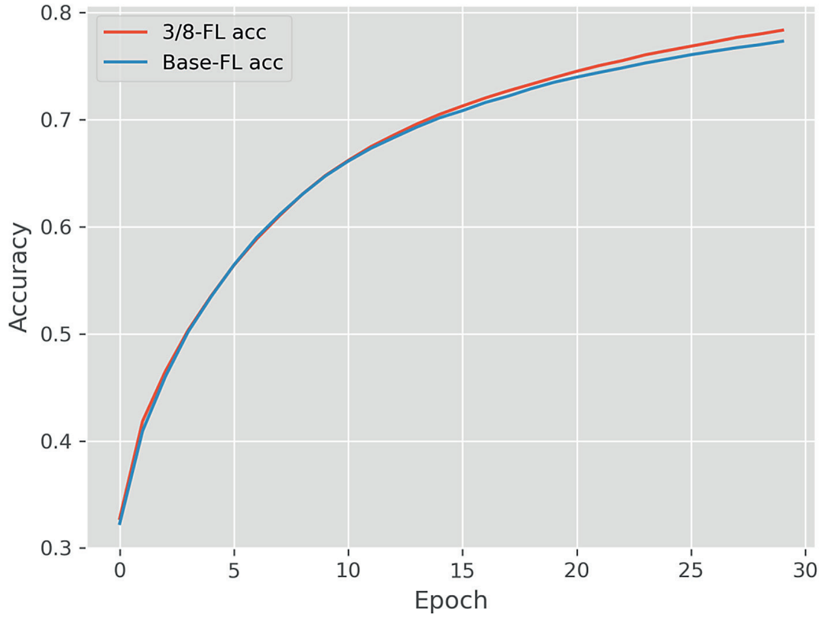

Figure 6. Average accuracy in subsequent epochs on test sets from MNIST dataset for non-IID scenario, with weight set based on local models' accuracy in pre-training, smooth OWA with 3/8 quadrature compared to base federated learning with averaging

fashion-MNIST set, the best performance was with the set of weights obtained by an analogous method as for MNIST, that is, from the accuracy of local classifiers in pre-training. The results for fashion-MNIST are shown in Table 4 for IID division and in Table 5 for Non-IID.

For the fashion-MNIST dataset and the IID data split, no significant improvement in the accuracy metric can be observed using either the classical or smooth OWA operator. Regardless of the method used, the results are very similar, accuracy is around 85%, and the standard deviation is very small at around 0.3%.

When splitting the data with Non-IID variant in the fashion-MNIST set, the use of smooth OWA operators yields a significant improvement

Table 4. Accuracy on fashion-MNIST dataset for IID scenario, with weight set based on local models' accuracy in pre-training

Method	Min	Mean	Median	Std	Max	Wilcox. Is better than avg	Wilcox. Is better than OWA
Average	84.5929	85.0121	84.9893	0.3192	85.5286	-	-
Base OWA	84.6143	85.0143	84.9821	0.3207	85.5714	no	-
OWA 3/8	84.6357	85.0229	84.9821	0.3192	85.5571	no	no
OWA trap.	84.6643	85.0343	84.9893	0.3164	85.5643	no	no
OWA ONC4	84.6500	85.0364	84.9964	0.3122	85.5571	no	no

Table 5. Accuracy on fashion-MNIST dataset for Non-IID scenario, with weight set based on local models' accuracy in pre-training

Method	Min	Mean	Median	Std	Max	Wilcox. Is better than avg	Wilcox. Is better than OWA
Average	69.2714	71.2264	71.0964	1.4824	73.6500	-	-
Base OWA	69.2143	71.2021	71.0571	1.4807	73.6357	no	-
OWA 3/8	70.1429	72.3850	72.7500	1.5405	74.0143	yes	yes
OWA trap.	69.5714	71.8829	72.4429	1.5724	73.5429	no	no
OWA ONC4	71.0714	73.1136	73.4250	1.4757	74.8286	yes	yes

in the accuracy metric. The baseline method achieved an average accuracy of 71.23%, while aggregation with the classical OWA operator produced a similar result of 71.2%. Smoothing with the trapezoidal quadrature was able to raise this score to 71.88%, but this improvement was not found to be statistically significant in the Wilcoxon test. Instead, a significant improvement was achieved with the 3/8 and ONC4 methods, where for 3/8 there is an average accuracy of 72.39% (about 1.2 percentage points higher than for the baseline method), and for ONC4 accuracy averaged 73.11% which is about 1.9 percentage points better than with the baseline method.

The graphs in Figure 7 and Figure 8 show the average accuracy values at successive epochs on the fashion-MNIST set with the Non-IID data split. Regardless of the method used, it can be seen that accuracy increases somewhat more rapidly in the first epochs than for the MNIST set. Differences in favor of smooth OWA operators are also noticeable quickly, already in the initial epochs, and remain at a similar level for most of the training. For the 3/8 method, the difference is more pronounced than on the MNIST set, and the ONC4 method behaves similarly on both datasets.

In case of the CIFAR10 dataset, no significant improvement in accuracy metrics could be obtained using OWA operators with weights determined by the accuracy of local models in pre-training (improvement if appeared was small and not statistically significant). Instead, good results were obtained using two other sets of weights. They allowed a significant improvement in the classification metric for the IID data distribution. However, this did not transfer to good results with the Non-IID distribution. In the latter case, a significant decrease in accuracy was even apparent using OWA and smooth OWA compared to the baseline method. Only the results for IID division are shown. Table 1 Dataset information Table 6 presents results for weight set of inverses of consecutive natural numbers, while Table 7 shows scores for weight set corresponding to client sets sizes.

From Table 6, it can be observed that the use of smoothing quadratures resulted in a significant improvement in accuracy by almost 1 percentage point (the largest growth for trapezoid and ONC4 quadratures). The use of the classical OWA operator improved the result of the baseline method by almost 0.8 percentage points. Both the classic OWA and its smoothed version yielded accuracy improvements that were significant in the Wilcoxon test, but the smooth OWA with either quadrature was not significantly different from the baseline OWA. An additional advantage of the methods with smoothing is their lower standard deviation than for federated learning with averaging or with classic OWA.

As can be seen from Table 7, the classic OWA method did not yield a significant improvement over averaging in federated learning. In contrast, the smooth OWA operators significantly improved both the result of the baseline

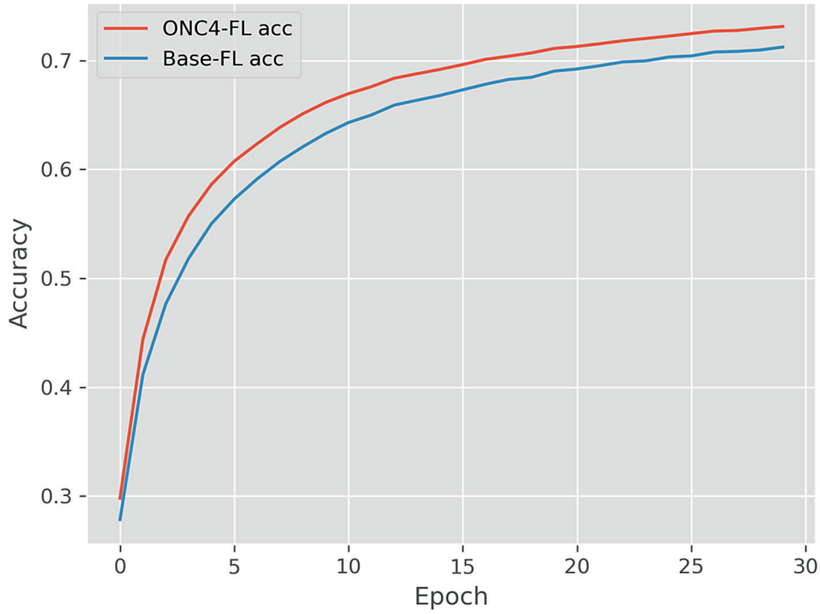

Figure 7. Accuracy in subsequent epochs on fashion-MNIST dataset for Non-IID scenario, with weight set based on local models' accuracy in pre-training, smooth OWA with ONC4 quadrature compared to base federated learning with averaging

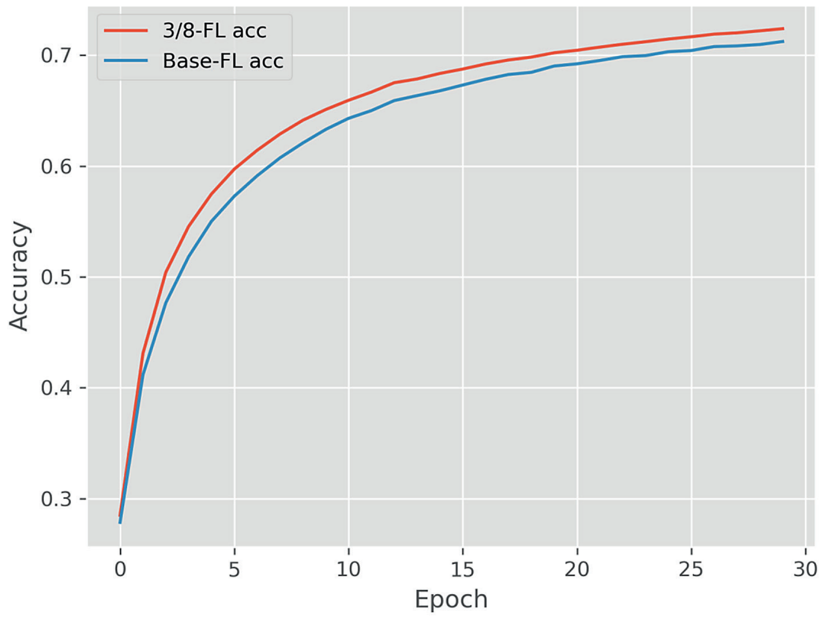

Figure 8. Average accuracy in subsequent epochs on test sets from fashion-MNIST dataset for Non-IID scenario, with weight set based on local models' accuracy in pre-training, smooth OWA with 3/8 quadrature compared to base federated learning with averaging

method and the classic OWA. The recorded improvement in accuracy is about 0.5 percentage points – the least for trapezoidal quadrature and the most for ONC4 quadrature. There is a noticeably smaller standard deviation for OWA and Smooth OWA than for the baseline method.

Figure 9 shows the average accuracy values at successive epochs on the CIFAR10 set for the IID data distribution. It can be seen that in the first epochs the differences between the baseline federated learning and the one with ONC4 smoothing method are small, until somewhere between epochs 5 and 10 they become pronounced. The accuracy values were lower after 30 epochs than on the other sets, which is due to the fact that the images in CIFAR10 were larger and contained

Table 6. Accuracy on CIFAR10 dataset for IID scenario, with weight set of inverses of consecutive natural numbers

Method	Min	Mean	Median	Std	Max	Wilcox. Is better than avg	Wilcox. Is better than OWA
Average	44.2417	45.0025	44.9375	0.5332	45.8500	-	-
Base OWA	44.9417	45.8058	45.6875	0.5396	46.4917	yes	-
OWA 3/8	45.3250	45.8517	45.8083	0.4369	46.3833	yes	no
OWA trap.	45.3583	45.9775	45.9458	0.4483	46.6250	yes	no
OWA ONC4	45.3083	45.9667	45.9250	0.4723	46.6333	yes	no

Table 7. Accuracy on CIFAR10 dataset for IID scenario, with weight set based on client sets sizes

Method	Min	Mean	Median	Std	Max	Wilcox. Is better than avg	Wilcox. Is better than OWA
Average	44.5250	45.2425	45.2042	0.5851	46.1917	-	-
Base OWA	44.6417	45.3442	45.3833	0.3926	45.9500	no	-
OWA 3/8	44.8417	45.8033	45.8417	0.4228	46.2167	yes	yes
OWA trap.	45.0000	45.7758	45.8833	0.3835	46.2917	yes	yes
OWA ONC4	45.0417	45.8358	45.8375	0.4263	46.4667	yes	yes

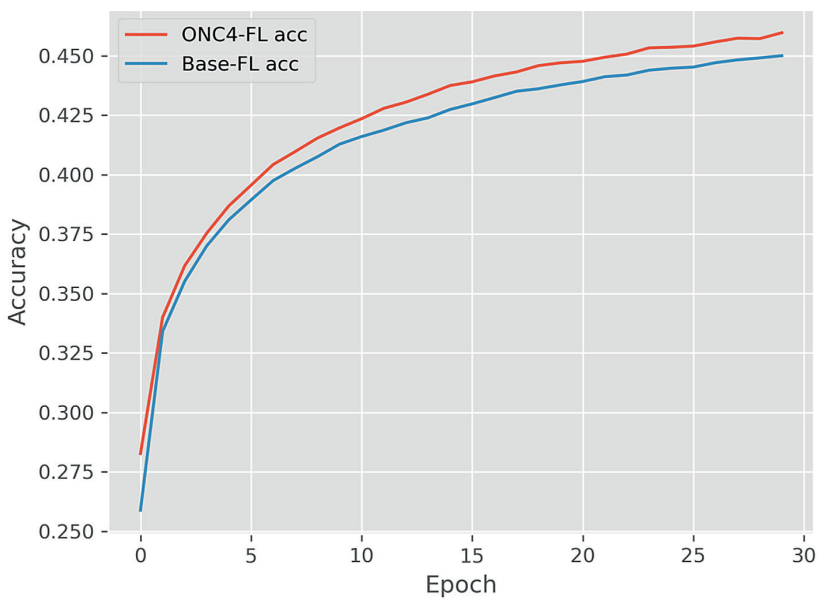

Figure 9. Average accuracy in subsequent epochs on test sets from CIFAR10 dataset for IID scenario, with weight set of inverses of consecutive natural numbers, smooth OWA with ONC4 quadrature compared to base federated learning with averaging

3 color channels, in contrast to the less complex images in the rest of the sets. To achieve higher accuracy, more epochs of training or a more complex network structure such as a convolutional network, would be required.

DISCUSSION

The federated learning algorithm, in its basic form known as FedAvg [1], can be enhanced by replacing the arithmetic mean with a weighted average, where different strategies for weight assignment can be considered [21], including the use of dynamically adjusted weights [27]. A further modification of FedAvg involves replacing the averaging mechanism entirely with more advanced and flexible aggregation operators.The introduction of new aggregation methods in federated learning is currently a significant and promising research direction, as demonstrated by the works such as [19], which presents the integration of the Sugeno integral as an aggregation mechanism in the FL framework. Our proposal to use the smooth OWA operator aligns with this recent trend.

The application of OWA and smooth OWA operators also supports FL's overarching goal of maximizing data security and privacy [28]. Even in the baseline FedAvg, no raw data is transmitted; only the local model parameters are shared, which already ensures a considerable level of data privacy, though some vulnerability to information leakage still remains [29]. Operators from the OWA family, however, assign weights not to individual clients but to ranks in the sorted set of values, making it impossible to infer which client received which weight, in contrast to weighted averaging.

In our work, we also introduced several promising strategies for selecting weights for OWA-family operators. In some cases, these strategies effectively handled Non-IID data, which is a critical issue in FL [4, 24]. An interesting direction for future research could involve combining OWA operators with dynamically assigned weights based on the characteristics of local models [30] or with other forms of dynamic adjustments [31].

CONCLUSIONS

In this work, a recent modification of the OWA operator was successfully applied in the aggregation of local model coefficients in the federated learning framework. The proposed method was tested in two scenarios of data partitioning: when the data is independently and identically distributed (IID) among clients, and in the opposite situation (Non-IID), more difficult for the model but closer to reality. Three datasets containing small-scale images were used in the experiments. Three methods of selecting weights for the OWA and smooth OWA operator as well as three quadratures for smoothing were tested.

The results of numerical experiments indicate that the use of smooth OWA in federated learning can contribute to a significant increase in model accuracy, both relative to the base method (federated learning + average) and to aggregation with the classical OWA operator. Often, aggregation using smooth OWA also exhibited a lower standard deviation of accuracy across multiple iterations of the experiment than occurred for the baseline method. This indicates that the proposed method is more stable with different divisions of the set into training and testing parts. In the cases studied, usually the greatest improvement in accuracy was provided by using ONC4 quadrature, followed by the 3/8 method, and usually a slightly smaller increase was for trapezoidal quadrature. The improvement also tended to occur with Non-IID splitting, which makes it possible to apply the new method also to less obvious splitting of a dataset between clients. Smooth OWA operators, when applied to federated learning, proved to be very sensitive to the set of weights used: often different weights had to be selected for different datasets and different split methods (IID/Non-IID).

On the MNIST and fashion-MNIST sets, smooth OWA operators performed best with weights based on the accuracy of local models in pre-training rounds. A significant increase in accuracy was observed on the Non-IID data split. For the MNIST set, significant improvements were made by the 3/8 (about 1 percentage point) and ONC4 (almost 2.8 percentage points) quadratures. For the fashion-MNIST set, these two quadratures also produced the best results, improving the baseline method by 1.2 and 1.9 percentage points, respectively. On the CIFAR10 set, equally good results were not achieved with this set of weights and the Non-IID distribution, but significant improvements in accuracy of 0.5 to sometimes almost 1 percentage point were achieved with the IID distribution and two other sets of weights. Lower standard deviations were also recorded for methods using smooth OWA than for the baseline method.

The experimental results confirm that the proposed smooth OWA-based aggregation methods offer a significant improvement over traditional schemes, often also in Non-IID and unbalanced federated learning scenarios. The introduction of smooth operator variants allows for more nuanced control over the aggregation process. This study demonstrates that incorporating operator-theoretic concepts such as OWA into federated model aggregation not only improves accuracy but also opens a new direction for interpretable and customizable learning frameworks.

In the future, attempts to select the best possible weights for OWA operators can be continued, such as weights based on the diversity of local sets, determining the quality of these sets, for example, based on the number of classes present in them (with Non-IID partitioning). The structure of the neural network serving as a classifier can be expanded to see if the proposed aggregation method would work equally well using a convolutional network. It is also worthwhile to introduce other aggregation methods instead of OWA, such as Choquet integral modifications, which perhaps better take into account the importance of individual local models or the interactions between them.

REFERENCES

1. McMahan HB, Moore E, Ramage D, Hampson S, Arcas BA y. Communication-Efficient Learning of Deep Networks from Decentralized Data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54. C
2. Li T, Sahu AK, Zaheer M, Sanjabi M, Talwalkar A, Smith V. Federated Optimization in Heterogenous Networks. In: Proceedings of the 3 rd MLSys Conference, Austin, TX, USA, 2020.
3. Wang J, Liu Q, Liang H, Joshi G, Poor HV. Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization. In: Proceedings of the 34th International Conference on Neural Information Processing Systems 2020; 638: 7611–7623.
4. Wang H, Kaplan Z, Niu D, Li B. Optimizing Federated Learning on Non-IID Data with Reinforcement Learning. In: IEEE INFOCOM 2020 - IEEE Conference on Computer Communications. IEEE; 2020; 1698–707. https://doi.org/10.1109/INFOCOM41043.2020.9155494
5. Sheller MJ, Edwards B, Reina GA, Martin J, Pati S, Kotrotsou A, et al. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. Sci Rep. 2020 Jul 28;10(1):12598. https://doi.org/10.1038/s41598-020-69250-1
6. Bos JW, Lauter K, Naehrig M. Private predictive analysis on encrypted medical data. J Biomed Inform. 2014 Aug;50:234–43. https://doi.org/10.1016/j.jbi.2014.04.003
7. Brisimi TS, Chen R, Mela T, Olshevsky A, Paschalidis IC, Shi W. Federated learning of predictive models from federated Electronic Health Records. Int J Med Inform. 2018 Apr; 112: 59–67. https://doi.org/10.1016/j.ijmedinf.2018.01.007
8. Chen Y, Wang J, Yu C, Gao W, Qin X. FedHealth: A federated transfer learning framework for wearable healthcare. IEEE Int Syst 2019; 35(4).
9. Hard A, Rao K, Mathews R, Ramaswamy S, Beaufays F, Augenstein S, et al. Federated learning for mobile keyboard prediction. arXiv: 2018; 1811: 03604.
10. Mizher HS, Taher MA, Falih BS, Gierz ŁA, Warguła Ł, Wieczorek B. Detecting and classifying media images of athletes using convolutional neural networks – case study: Individual sports images. Adv Sci Technol Res J. 2025 Jun 1; 19(6): 152–66. https://doi.org/10.12913/22998624/199798
11. Saad A, Ullah Sheikh U, Moslim MS. Developing convolutional neural network for recognition of bone fractures in X-ray images. Adv Sci Technol Res J. 2024 Aug 1; 18(4): 228–37. https://doi.org/10.12913/22998624/188656
12. Bereta M, Karczmarek P, Pedrycz W, Reformat M. Local descriptors in application to the aging problem in face recognition. Pattern Recognit. 2013 Oct; 46(10): 2634–46. https://doi.org/10.1016/j.patcog.2013.03.010
13. Sterniczuk B, Charytanowicz M. An ensemble transfer learning model for brain tumors classification using convolutional neural networks. Adv Sci Technol Res J. 2024 Dec 1; 18(8): 204–16. https://doi.org/10.12913/22998624/193627
14. Falih BS, Gierz ŁA, Al-Zaidi GA. Detecting clustered fruits using a hybrid of convolutional neural networks and machine learning classifiers – Case study. Adv Sci Technol Res J. 2025 Apr 1; 19(4): 1–9. https://doi.org/10.12913/22998624/195588
15. Grycuk R, Wojciechowski A, Wei W, Siwocha A. Detecting visual objects by edge crawling. J Artif Intell Soft Comput Res. 2020 Jul 1; 10(3): 223–37. https://doi.org/10.2478/jaiscr-2020-0015
16. Kusy M, Kowalski PA. Weighted probabilistic neural network. Inf Sci (Ny). 2018 Mar; 430–431: 65–76. https://doi.org/10.1016/j.ins.2017.11.036
17. Wang G, Dang CX, Zhou Z. Measure Contribution of Participants in Federated Learning. In: 2019 IEEE International Conference on Big Data (Big Data). IEEE; 2019; 2597–604. https://doi.org/10.1109/BigData47090.2019.9006179
18. Wang Z, Fan X, Qi J, Wen C, Wang C, Yu R. Federated Learning with Fair Averaging. In: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence 2021.
19. Pękala B, Wilbik A, Szkoła J, Dyczkowski K, Żywica P. Federated Learning with the Choquet Integral as Aggregation Method. In: 2024 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). IEEE; 2024; 1–8. https://doi.org/10.1109/FUZZ-IEEE60900.2024.10611748
20. Karczmarek P, Gregosiewicz A, Łagodowski ZA, Dolecki M, Gałka Ł, Powroźnik P, et al. Smooth and enhanced smooth quadrature-inspired generalized Choquet integral. Fuzzy Sets and Systems 2024; 483: 108926.
21. Wilbik A, Pękala B, Dyczkowski K, Szkoła J. A Comparison of Client Weighting Schemes in Federated Learning. In: Proceedings of International Workshop on Intuitionistic Fuzzy Sets and Generalized Nets, 2023; 116–28. https://doi.org/10.1007/978-3-031-45069-3_11

22. Yeganeh Y, Farshad A, Navab N, Albarqouni S. Inverse Distance Aggregation for Federated Learning with Non-IID Data. In: Proceedings of MICCAI Workshop on Domain Adaptation and Representation Transfer 2020.
23. Zhang C, Xie Y, Bai H, Yu B, Li W, Gao Y. A survey on federated learning. Knowledge-Based Syst. 2021 Mar; 216: 106775. https://doi.org/10.1016/j.knosys.2021.106775
24. Zhao Y, Li M, Lai L, Suda N, Civin D, Chandra V. Federated Learning with Non-IID Data. 2018. https://doi.org/10.48550/arXiv.1806.00582
25. Yager RR. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. IEEE Trans Syst Man Cybern. 1988;18(1):183–90. https://doi.org/10.1109/21.87068
26. Rachwał A, Karczmarek P, Rachwał A. smooth ordered weighted averaging operators. Inf Sci (Ny). 2025 Jan; 686: 121343. https://doi.org/10.1016/j.ins.2024.121343
27. Liu J, Wu J, Chen J, Hu M, Zhou Y, Wu D. FedDWA: Personalized Federated Learning with Dynamic Weight Adjustment. In: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, 2023.
28. Mothukuri V, Parizi RM, Pouriyeh S, Huang Y, Dehghantanha A, Srivastava G. A survey on security and privacy of federated learning. Futur Gener Comput Syst. 2021 Feb; 115: 619–40. https://doi.org/10.1016/j.future.2020.10.007
29. Xu X, Wu J, Yang M, Luo T, Duan X, Li W, et al. Information Leakage by Model Weights on Federated Learning. In: Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice. New York, NY, USA: ACM; 2020; 31–6. https://doi.org/10.1145/3411501.3419423
30. Zeng Y, Mu Y, Yuan J, Teng S, Zhang J, Wan J, et al. Adaptive federated learning with Non-IID Data. Comput J. 2023 Nov 14 ;66(11): 2758–72. https://doi.org/10.1093/comjnl/bxac118
31. Zhang J, Guo S, Qu Z, Zeng D, Zhan Y, Liu Q, et al. Adaptive federated learning on Non-IID data with resource constraint. IEEE Trans Comput. 2022 Jul 1; 71(7): 1655–67. https://doi.org/10.1109/TC.2021.3099723