The Sugeno Integral Used for Federated Learning with Uncertainty for Unbalanced Data

The Sugeno integral used for federated learning with uncertainty for unbalanced data

Anna Wilbik
Maastricht University
Maastricht, The Netherlands
a.wilbik@maastrichtuniversity.nl

Barbara Pękala
University of Rzeszów,
University of Information Technology and Management
Rzeszów, Poland
bpekala@ur.edu.pl

Jarosław Szkoła
University of Rzeszów
Rzeszów, Poland
jszkola@ur.edu.pl

Krzysztof Dyczkowski
Adam Mickiewicz University
Poznań, Poland
chris@amu.edu.pl

***Abstract*—Data is crucial in the digital economy. Many businesses collect and use their data to enhance their performance. However, limited data or low data quality can hinder model development, particularly in dynamic environments. To overcome this, companies collecting similar data may opt to exchange knowledge without sharing their data, due to privacy or legal issues. This is where federated learning comes in. In horizontal federated learning, each client (organization) iteratively improves its model, so that it can be regularly aggregated and shared with all clients participating in the federation for further improvements. In federated averaging, the aggregation mechanism is based on the weighted average and the weights depend on the amount of data available to each client. In this paper, we propose to use a more advanced aggregation mechanism, namely the Sugeno integral. The initial results are promising.**

I. Introduction

With advancements in technology, more data is being generated through business processes and IoT devices. The amount of data generated has been growing at a rapid pace, with some sources estimating that the amount of data generated globally will continue to grow at an exponential rate. According to a report by IDC, the total amount of data generated globally is expected to reach 175 zettabytes by 2025, which is a significant increase from the 33 zettabytes generated in 2018. The report also states that approximately 50% of this data will be generated by IoT devices. Another study by Markets and Markets estimates that the IoT data analytics market will grow from US$12.0 billion in 2018 to US$40.6 billion by 2023, at a CAGR of 26.9% during the forecast period [1], [2], [3].

Organizations seek to utilize this data to inform their decision-making, with one common approach being the creation of predictive models using patterns from past data to predict future situations. However, building high-quality predictive models requires a substantial amount of historical data. In some cases, an organization may not have enough data to develop an effective model on its own. Sharing data with other organizations may not be an option due to competitive or privacy concerns, such as those arising from the EU's General Data Protection Regulation (GDPR) [4]. Federated learning offers a solution by enabling the creation of models through data collaboration across organizations, avoiding these privacy and security issues.

Federated learning allows multiple parties to train a machine learning model together without sharing their local data [5]. This makes it a privacy-preserving approach. Success in federated learning is measured by the improvement of at least one party's model performance compared with its local model [6]. Federated learning has been successfully applied in various domains, including cross-device federated learning on mobile devices, such as Google Keyboard [7], and cross-silo federated learning among organizations in industries like healthcare [8] and in finance for transaction fraud detection [9]. The use of federated learning for processing IoT data to support decision-making in business processes was explored by building a concept model [10] and a demonstrator [11].

The main idea behind federated learning is the continuous improvement of the model by means of regular aggregation of the models trained on the local data of each client in the federation. In the original proposal [12], a weighted average was used with weights depending on the amount of data available to each client. However, this is not the only possible choice for the aggregation function. Driven by the curiosity, we want to test whether using a more complex aggregation will improve the quality of the federated model. In this work, to achieve a better understanding of the impact of this function, we experiment with a Sugeno integral as an aggregation mechanism. Initial results show that the use of a more advanced aggregation method improves the quality of the federated model.

II. Background

A. Federated learning

Federated learning enables collaboration between multiple parties to jointly train a machine learning model without exchanging their local data [5]. The federated learning model was originally proposed by Google researchers [13], [14], [12]. Their main idea was to build machine learning models based on datasets that are distributed across multiple devices (cf. [15], [16] or [17]).

Generally, FL can be divided into different scenarios based on how the data is partitioned or distributed among the data owners, i.e., horizontally or vertically. Horizontal federated learning is used when different parties collect the same features but from different subjects. A common example of horizontal federated learning is a group of hospitals collaborating to build a model that can predict a health risk for their patients, based on agreed data. Vertical federated learning is used when multiple parties share not the features, but the subjects – for example, a telecom company collaborating with a home entertainment company (cable television provider), or an airline collaborating with a car rental agency.

In this paper, we consider a horizontal federated learning scenario. Figure 1 shows the general architecture of the federated model. The assumption is that all clients have the same local data structure and use a common machine learning model. They exchange with the server only coefficients describing the learned local models and parameters describing the classification quality, which are used only to determine the stopping point of the iteration process. The server performs model aggregation, that is, suitable aggregation of the coefficients. The server then returns the new coefficients to the clients.

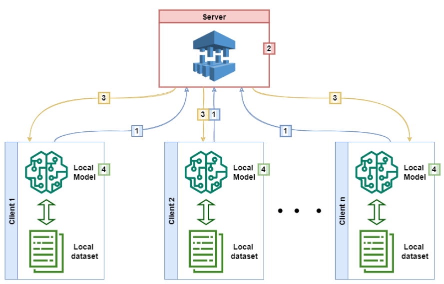

Fig. 1. Proposed federated model

In the original federated averaging method [12], the aggregation uses weighted averaging, where the weights depend on the amount of data available to each client. However, this is not the only solution. For instance, FedMA [18] combines the weights of neurons with similar characteristics based on the permutation invariant of the neural network, and makes efficient use of the communication rounds to improve the convergence speed. Another approach, inspired by active learning, assumes that the central server can evaluate every client's data and indicate the potential utility of training before aggregation, and select only the most useful clients to participate in the training [19]. Zhao [20] proposed that the central server should collect a small amount of data from each client for macro-distribution, resulting in more homogeneous data. There have also been experiments with different weighting schemes, such as [21], where the authors assigned weights to the clients according to a fairness measure, or [22], where the weights depended on the quality of the local models.

B. Interval-valued fuzzy set theory

Since 1965, when Zadeh [23] proposed fuzzy sets, and later since 1975 (Sambuc and Zadeh, [24], [25]) when the study of extensions of fuzzy sets began, we have been able to effectively model the uncertainty and imprecision of data or decisions. Here, we use the terminology of interval calculus to describe the data; in particular, we will denote by $L^I = \{[\underline{p}, \overline{p}] : \underline{p}, \overline{p} \in [0,1], \ \underline{p} \leq \overline{p}\}$ the family of intervals belonging to the unit interval. In many areas, data aggregation is required, to summarize information based on data. An aggregate function takes as its input a set, a multiset (bag) from some input domain, and creates outputs as elements of an output domain. For input data in the form of interval-valued fuzzy sets, thus incorporating the uncertainty aspect, we can find the definition of aggregations in [26], [27], [28]. Aggregation functions on L^I are a significant concept in numerous applications (e.g. [29], [30], [27]). Often applied in practice are Ordered Weighted Averaging (OWA) operators, introduced by Yager in 1988. OWA operators are a particular case of the more general aggregation functions called Choquet integrals. In [31], the class of linear orders on L^I is used to extend the definition of OWA operators for the interval-valued fuzzy setting in the following way.

Definition 1 ([31])**.** Let $\leq$ be an admissible order on L^I, and $w = (w_1, \ldots, w_n) \in [0,1]^n$, with $w_1 + \cdots + w_n = 1$. The interval-valued ordered weighted averaging (OWA) operator (IVOWA) associated with $\leq$ and w is a mapping $IVOWA_{\leq,w} : (L^I)^n \to L^I$, given by

$$IVOWA_{\leq,w}([\underline{x}_1, \overline{x}_1], \ldots, [\underline{x}_n, \overline{x}_n]) = \sum_{i=1}^{n} w_i \cdot [\underline{x}_{(i)}, \overline{x}_{(i)}],$$

where $[\underline{x}_{(i)}, \overline{x}_{(i)}]$, $i = 1, \ldots, n$, denotes the i-th greatest of the inputs with respect to the order $\leq$ and $w \cdot [\underline{x}, \overline{x}] = [w\underline{x}, w\overline{x}]$, $[\underline{x}_1, \overline{x}_1] + [\underline{x}_2, \overline{x}_2] = [\underline{x}_1 + \underline{x}_2, \overline{x}_1 + \overline{x}_2]$.

A special case of IVOWA is the interval arithmetic mean for weights of 0.5 (denoted here by $\mathcal{A}_{mean}$). Different interval arithmetics have also proved important in the development of the theory of uncertainty. The most common and most frequently used interval arithmetic is Moore arithmetic [32], [33]. In Moore arithmetic, basic operations on intervals $X = [\underline{x}, \overline{x}]$ and $Y = [\underline{y}, \overline{y}]$ are realized by formulae for sum, difference, and product:
$[\underline{x}, \overline{x}] + [\underline{y}, \overline{y}] = [\underline{x} + \underline{y}, \overline{x} + \overline{y}]$;
$[\underline{x}, \overline{x}] - [\underline{y}, \overline{y}] = [\underline{x} - \overline{y}, \overline{x} - \underline{y}]$;
$a * [\underline{x}, \overline{x}] = [a\underline{x}, a\overline{x}], \quad a \in R^+$;
$a * [\underline{x}, \overline{x}] = [a\overline{x}, a\underline{x}], \quad a \in R^-$; $[\underline{x}, \overline{x}] * [\underline{y}, \overline{y}] =$
$[\min(\underline{x} * \underline{y}, \overline{x} * \overline{y}, \underline{x} * \overline{y}, \overline{x} * \underline{y}), \max(\underline{x} * \underline{y}, \overline{x} * \overline{y}, \underline{x} * \overline{y}, \overline{x} * \underline{y})]$
for $\underline{x}, \overline{x}, \underline{y}, \overline{y} \in R$ and $\underline{x} \leq \overline{x}, \underline{y} \leq \overline{y}$.

Some limitations and drawbacks have been found in Moore interval arithmetic, such as the excess width effect problem. As an alternative to Moore arithmetic, we may use multidimensional interval arithmetic. as developed by Piegat [34]. Here, a given value x from the interval $X = [\underline{x}, \overline{x}]$ is described using the variable γ_x, with $\gamma_x \in [0,1]$, as shown:

$$Rep_\gamma(x) = \underline{x} + \gamma_x(\overline{x} - \underline{x}). \tag{1}$$

In this notation the interval $X = [\underline{x}, \overline{x}]$ is described in the form $X = \{Rep_\gamma(x) : Rep_\gamma(x) = \underline{x} + \gamma_x(\overline{x} - \underline{x}), \gamma_x \in [0,1]\}$. The variable γ_x makes it possible to obtain any value between the left boundary $\underline{x}$ and right boundary $\overline{x}$ of the interval X.

C. Fuzzy measure and Sugeno integral

There are several important fuzzy integrals, such as the Choquet or Sugeno integrals proposed in 1974 for fuzzy sets.

Let us recall the definition of fuzzy measure.

Definition 2. Let X be a finite set. A function $m : 2^X \to [0, 1]$ is a fuzzy measure (or monotone measure) if it satisfies the following properties:
1. $m(\varnothing) = 0$.
2. $m(X) = 1$
3. If $A, B \subseteq X$ and $A \subseteq B$ then $m(A) \leq m(B)$.

In particular, a fuzzy measure is called a Sugeno measure (λ-measure) if it satisfies for all $A, B \subseteq X$ and $A \cap B \neq \varnothing$

$$m(A \cup B) = m(A) + m(B) + \lambda(g(A)g(B)), \qquad (2)$$

where $\lambda > -1$.

For a discrete set X, $X = \{x_1, ..., x_n\}$ and $g^j = g\{x_j\}$ for $j = 1, ..., n$, then λ can be solved from:

$$1 + \lambda = \prod_{j=1}^{n} (1 + \lambda g^j). \qquad (3)$$

Sugeno measures are among the most extensively used and most successful fuzzy measures (e.g. [35]).

Based on a fuzzy measure we can define a very important measure: the integral fuzzy measure. In particular, we propose the Sugeno integral measure. We will use the Sugeno integral to aggregate the parameters of local models in the federated learning model.

Definition 3. Let g be a fuzzy measure and h be a function $h : X \to [0, 1]$. Assume that the (x_i) are ordered so that $h(x_1) \geq h(x_2) \geq ... \geq h(x_n)$. A discrete Sugeno integral (cf. [36]) of a function h with respect to g is a function $S_g : [0, 1]^n \to [0, 1]$ such that

$$S_g(x) = \max_{i=1,...,n} (\min(h(x_i), m(A_i))), \qquad (4)$$

where $A_i = \{x_i\}$, $h(x_i)$ is ordered antitonically.

III. Proposed method

We consider a horizontal federated learning scenario, where each client has its own independent data set $\{Y_i, x_{i1}, ...x_{ip}\}$ and $x_{ip} \in L^I$, $Y_i \in \{0, 1\}$ for $i = 1, ..., n$, n is the number of instances, and p is the number of attributes. Each client trains a set model on its data (n_k observations) in a specified number of internal iterations, and provides the training result in the form of a result vector according to the selected machine learning model of the trained parameters, β and ϵ,

$$y_i = \beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip} + \epsilon_i$$

for $i = 1, ...n_k$ and $\beta_k \in R$ for $k = 1, ...p$.

In our proposal, we follow the original algorithm as shown in Figure 1, but with a different aggregation step. Therefore, federated learning is initialized by the server, by sending the model to a subset of the clients. Next, the model is trained by executing the following four steps:

1) Each contacted client performs a few training steps of its own model on its local data and passes it to the server (parameters of models and their efficiency, e.g., Accuracy (ACC), Sensitivity (SENS), Specificity (SPEC), Precision (PREC), the Area Under the ROC Curve (AUC)) (line 1);
2) The server aggregates the models using the Sugeno measure based on the model's effectiveness for each client (line 2);
3) The server returns the new model to the clients (line 3);
4) Local models are updated (line 4).

Below we explain how to use the Sugeno integral to aggregate the model parameters.

First, we need to calculate the densities and λ. The densities depend on the model quality, denoted as Q_i. In this paper, we use ACC and AUC. The densities are calculated as:

1) If $\sum_{i=1}^{k} Q_i \leq 1$, then $g(x_i) = \frac{Q_i}{\sum_{i=1}^{k} Q_i}$ and $\lambda = 0$;
2) If $\sum_{i=1}^{k} Q_i > 1$, then $g(x_i) = Q_i$ and we calculate λ from (3) and $g(\{x_i\})$ from (2).

Next, we can aggregate the model parameters. The new parameters are the values of the Sugeno integral $S_g(x)$ of model parameters $x = (\beta_i^1, ..., \beta_i^k)$ for each $i = 1, ..., p$ and $h(x_i) \in \{\beta_i^1, ..., \beta_i^k\}$ for k models and the i-th parameters (all parameters are first normalized). To calculate the Sugeno integral $S_g(x)$, we use equation (4).

For the local learning process, we chose logistic regression with stochastic gradient descent, but modified for interval data. Calculation of the model response for each training sample is done according to the sigmoid function:

$$f(y_i) = \frac{1}{1 + e^{-Rep_\gamma(\beta_0 + \beta_1 \cdot x_{i1} + ... + \beta_p \cdot x_{ip} + \epsilon_i)}}$$

for $\gamma \in [0, 1]$ and $f : L^I \to R$.

We update the learning coefficients in the steps:

$$\beta_j = \beta_j + \alpha \cdot \bigtriangledown_{\beta_j} \mathcal{L}(y_i) \cdot x_{ij},$$

$$\beta_0 = \beta_0 + \alpha \cdot \bigtriangledown_{\beta_0} \mathcal{L}(y_i),$$

where α is the learning coefficient and $\bigtriangledown$ is the gradient, for $i = 1, .., n_k,\ j = 1, .., p$, and where Y_i is the current output value and

$$\mathcal{L}(y_i) = -\log(f(y_i)) \cdot Y_i - \log(1 - f(y_i)) \cdot (1 - Y_i).$$

IV. Experiments and results

In this section, we describe our initial evaluation of the proposed method. For evaluation purposes, we use the publicly available Wisconsin (diagnostic) breast cancer dataset. We compare the results of the proposed method with a centralized model, a local model, and other federated learning solutions. In subsection IV-A we describe the Wisconsin (diagnostic) breast cancer dataset, subsection IV-B contains the methodology for building the different models, and in subsection IV-C we present the results that we obtained.

A. Dataset used in the experiments

The dataset used is the Wisconsin (diagnostic) breast cancer dataset. This is one of the popular datasets from the UCI Machine Learning Repository [37]. It contains information on 569 medical cases. Features are calculated from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe the characteristics of the cell nuclei present in the image as the pair of the mean and standard deviation. Those features are:

- radius (mean of distances from the center to points on the perimeter),
- texture (standard deviation of gray-scale values),
- perimeter,
- area,
- smoothness (local variation in radius lengths),
- compactness,
- concavity (severity of concave portions of the contour),
- concave points (number of concave portions of the contour),
- symmetry,
- fractal dimension ($coastline\ approximation - 1$).

The pair mean and standard deviation is not most useful representation to be used in a model, therefore we transform it to an interval representation. We construct it in the following way:

$$[mean - standard\ deviation,\ mean + standard\ deviation]$$

after prior fuzzification of both values, "mean–standard deviation" and "mean+standard deviation", by normalization.

The decision attribute stores information about the diagnosis: malignant (0) or benign (1). The dataset consists of 212 malignant objects and 357 benign objects. Since the dependent variable, the explained variable, takes two dichotomous values of 0 and 1, the optimal model choice for decision prediction turned out to be the logistic regression model, which determines the probability of a given event occurring for the values of the predictors entered into the model.

Creating local models for the federated model. To simulate the data sets of a group of clients (three in this case), the data were randomly divided into three groups with decision-balanced and unbalanced behavior. The data of each client were then randomly split into a training set and a test set in a ratio of 90% to 10%.

B. Models built in the experiments

We built the five sets of models:

1) centralized model
2) local models
3) federated averaging model
4) federated learning – aggregation via weighted average
5) federated learning – aggregation via Sugeno integral

In all models, we used the logistic regression model, where we assumed $\epsilon_i = 0.001$, $\alpha = 0.01$, and $\gamma = 0.5$.

Model 1: centralized model. The first benchmark is a centralized model with no missing data (no uncertainty). The model is trained by 10-fold cross-validation with stratification using a standard logistic regression model.

Model 2: local models. The second benchmark consists of local models. The standard logistic regression models were trained on the training set containing 90% of the data by 5-fold cross-validation with stratification.

Model 3: federated averaging model. This model was trained with federated averaging as proposed in [12]. The learning rate was set to 0.01, there were 50 local learning epochs, and the stopping criterion was set to 100 aggregation cycles.

Model 4: federated learning – aggregation via weighted average. This model was trained with federated learning as proposed in [22]. In this model, in the aggregation step, we use a weighted average, based on the quality of the model, namely accuracy and AUC. The learning rate was set to 0.01, there were 50 local learning epochs, and the stopping criterion was set to 100 aggregation cycles.

Model 5: federated learning – aggregation via Sugeno integral. This model was trained with federated learning as proposed in this paper.

Balanced and unbalanced datasets – a method of building unbalanced datasets. When splitting the data across clients, we consider two scenarios: data divided in a stratified manner ("iid data") and data divided in a biased manner ("non-iid data"). In the case when data are divided in a stratified manner, each client has the same number of samples of each class as other clients. This means that the proportions between malignant and benign classes are preserved. In the case where the data were divided in a biased manner, Client 1 had a 50–50% division, while Clients 2 and 3 had 70% of the classes malignant and benign, respectively.

C. Results

Here, we present the results of different federated models that we obtained in case of stratified and biased data division. However, we first give the results of the centralized model, which can be considered a good benchmark.

Model 1: centralized model

TABLE I
PERFORMANCE OF BENCHMARK MODEL

	ACC	SENS	SPEC	PREC
Complete data	0.965	0.972	0.935	0.965

The results of Model 1 are shown in Table I. The performance of the centralized model is good; however, some improvement is still possible.

As mentioned before, we consider two data partitions across the federation. Below are the results for the scenario with iid data.

Model 2: local models

The results of Model 2 are shown in Table II. We can see that the performance of the models trained only on the local data is inferior to that of the centralized model. This creates an opportunity for federated learning to show its value.

TABLE II
PERFORMANCE OF LOCAL MODELS WITH IID DATA

Dataset	ACC	SENS	SPEC	PREC	AUC
Client 1	0.94	0.96	0.89	0.94	0.938
Client 2	0.92	0.95	0.86	0.92	0.831
Client 3	0.92	0.92	0.94	0.96	0.867

TABLE III
PERFORMANCE OF FEDERATED AVERAGING MODEL WITH IID DATA

Dataset	ACC	SENS	SPEC	PREC	AUC
Client 1	0.94	0.96	0.89	0.94	0.938
Client 2	**0.921**	0.95	0.86	0.92	**0.832**
Client 3	0.92	0.922	0.94	0.96	0.868

Model 3: federated averaging model

The results of Model 3 are shown in Table III. When using federated averaging, we observe only a very small improvement compared with the local models.

Model 4: federated learning – aggregation via weighted average

TABLE IV
PERFORMANCE OF MODEL 4 WITH IID DATA

weights defined by the accuracy					
Dataset	ACC	SENS	SPEC	PREC	AUC
Client 1	0.94	0.96	0.89	0.94	0.938
Client 2	0.92	0.95	0.86	0.92	0.932
Client 3	0.92	0.92	0.94	0.96	0.957
weights defined by the AUC					
Dataset	ACC	SENS	SPEC	PREC	AUC
Client 1	0.94	0.96	0.89	0.94	0.919
Client 2	**0.93**	0.95	0.86	0.92	**0.933**
Client 3	0.92	0.922	0.94	0.96	0.949

The results of Model 4 are shown in Table IV, with weights defined by the accuracy and AUC. In this case, the improvement is larger, especially in terms of AUC and ACC for Client 2.

Model 5: federated learning – aggregation via Sugeno integral

TABLE V
PERFORMANCE OF MODEL 5 WITH IID DATA (SUGENO)

Dataset	ACC	SENS	SPEC	PREC	AUC
Client 1	**0.96**	0.94	**0.99**	**0.99**	**0.96**
Client 2	**0.94**	**0.96**	**0.99**	**0.99**	**0.96**
Client 3	**0.95**	**0.98**	0.9	0.95	**0.95**

The results of Model 5 are shown in Table V. In this case, we observe the greatest improvement. Model 5 outperforms even the central model in terms of precision. Moreover, during FL we improve the remaining efficiency parameters of the local models.

Non-balanced cases

Now, we simulate cases with non-iid data (where Client 1 had a 50–50% division, and Clients 2 and 3 had 70% of the classes malignant and benign, respectively).

TABLE VI
PERFORMANCE OF LOCAL MODELS WITH NON-BALANCED DATA

Dataset	ACC	SENS	SPEC	PREC	AUC
Client 1	0.94	0.96	0.89	0.94	0.938
Client 2	0.81	0.86	0.89	0.89	0.802
Client 3	0.70	0.89	0.78	0.88	0.826

Model 2: local models with non-balanced data

The results of Model 2 in the case where Clients 2 and 3 have non-balanced local data are shown in Table VI. We observe that the performance achieved with models trained only on the local data is inferior to that of the centralized model, and is very significantly lower in the case of the non-balanced data of Clients 2 and 3. This creates an opportunity and need for the implementation of federated learning.

Model 3: federated averaging model with non-balanced data

TABLE VII
PERFORMANCE OF FEDERATED AVERAGING MODEL WITH NON-BALANCED DATA

Dataset	ACC	SENS	SPEC	PREC	AUC
Client 1	0.94	0.96	**0.94**	**0.96**	**0.954**
Client 2	**0.916**	**0.89**	**0.98**	**0.98**	**0.943**
Client 3	**0.890**	0.89	0.78	0.88	**0.912**

The results of Model 3 for the case where Clients 2 and 3 have non-balanced local data are shown in Table VII. When using federated averaging, we can see some improvement compared with the local models, especially for non-balanced data.

Model 4: federated learning – aggregation via weighted average with non-balanced data

TABLE VIII
PERFORMANCE OF MODEL 4 WITH NON-BALANCED DATA AND AVERAGING WEIGHTED BY ACC

weights defined by accuracy					
Dataset	ACC	SENS	SPEC	PREC	AUC
Client 1	0.94	0.96	0.94	0.96	0.954
Client 2	**0.922**	0.89	**0.99**	0.97	**0.988**
Client 3	0.801	0.97	0.48	0.76	**0.943**
weights defined by AUC					
Dataset	ACC	SENS	SPEC	PREC	AUC
Client 1	0.94	0.96	0.94	0.96	0.954
Client 2	**0.93**	0.89	0.99	**0.99**	**0.991**
Client 3	**0.92**	**0.99**	0.72	0.88	**0.953**

The results of Model 4 for the case where Clients 2 and 3 have non-balanced local data are shown in Table VIII, with weights defined by the accuracy and AUC. In this case, improvement is achieved in terms of ACC and AUC, especially for non-balanced Clients 2 and 3.

Model 5: federated learning – aggregation via Sugeno integral with non-balanced data

The results of Model 5 (with the Sugeno integral as the averaging method) for the case where Clients 2 and 3 have non-balanced local data are shown in Table IX. In this case,

TABLE IX
PERFORMANCE OF MODEL 5 WITH NON-BALANCED DATA

Dataset	ACC	SENS	SPEC	PREC	AUC
Client 1	**0.955**	0.94	**0.99**	**0.99**	0.953
Client 2	0.93	0.89	0.99	0.99	0.88
Client 3	**0.95**	0.97	**0.91**	0.95	0.951

improvement is achieved in terms of accuracy, sensitivity, and precision for the non-balanced local clients, but the improvement also concerns an increase in efficiency in the model with decision-balanced data. The improvement with the use of the Sugeno integral is observed not only in relation to other aggregation methods used in FL, but also in relation to the central model.

V. CONCLUDING REMARKS

We have examined the impact of various aggregations in improving the prediction performance of models included in the federation. In the case of decision-balanced data, we obtain a very small improvement in the efficiency of local models for individual aggregations, but in the case of unbalanced data, the Sugeno integral significantly outperforms other aggregation methods, because not only does FL then improve models with unbalanced data, it also improves even the model with balanced data (cf. values of ACC), which indicates the high compatibility of the Sugeno integral with FL. In addition, we observed the high stability of the Sugeno integral in the context of various combinations of the number of epochs in local model training and the number of aggregations in the federated model. In future research, a more extensive examination of this issue will be made in relation to various types of models, including deep neural networks and data, including problems arising in industry or finance.

REFERENCES

[1] "Worldwide IDC global datasphere forecast," https://www.idc.com/getdoc.jsp?containerId=US49018922.

[2] "Global big data analytics market – forecast to 2023," https://www.researchandmarkets.com/reports/4702627/global-big-data-analytics-market-forecast-to.

[3] "IoT analytics market," https://www.marketsandmarkets.com/Market-Reports/iot-analytics-market-52329619.html.

[4] "Regulation (EU) 2016/679, OJ L 119, 4 May 2016, pp. 1–88."

[5] P. Kairouz, B. McMahan *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, pp. 1–210, 2021.

[6] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, and B. He, "A survey on federated learning systems: vision, hype and reality for data privacy and protection," *arXiv preprint arXiv:1907.09693*, 2019.

[7] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018.

[8] T. M. Deist, A. Jochems, J. van Soest, G. Nalbantov, C. Oberije, S. Walsh, M. Eble, P. Bulens, P. Coucke, W. Dries *et al.*, "Infrastructure and distributed learning methodology for privacy-preserving multi-centric rapid learning health care: euroCAT," *Clinical and Translational Radiation Oncology*, vol. 4, pp. 24–31, 2017.

[9] W. Zheng, L. Yan, C. Gou, and F.-Y. Wang, "Federated meta-learning for fraudulent credit card detection," in *IJCAI-20*, 2020.

[10] P. Grefen, H. Ludwig, S. Tata, R. Dijkman, N. Baracaldo, A. Wilbik, and T. D'hondt, "Complex collaborative physical process management: a position on the trinity of BPM, IoT and DA," in *Working Conference on Virtual Enterprises*. Springer, 2018, pp. 244–253.

[11] T. d'Hondt, A. Wilbik, P. Grefen, H. Ludwig, N. Baracaldo, and A. Anwar, "Using BPM technology to deploy and manage distributed analytics in collaborative IoT-driven business scenarios," in *IoT 2019*, 2019, pp. 1–8.

[12] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *AISTATS 2017*, 2017.

[13] J. Konecný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *ArXiv*, vol. 1610.02527, 2016.

[14] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *ArXiv*, vol. 1610.05492, 2017.

[15] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM TIST*, vol. 10, no. 2, 2019.

[16] A. Wilbik and P. Grefen, "Towards a federated fuzzy learning system." FUZZ-IEEE 2021, 2021, pp. 1–6.

[17] H. Yan, L. Hu, X. Xiang, Z. Liu, and X. Yuan, "Privacy-preserving collaborative learning for mitigating indirect information leakage," *Information Sciences*, vol. 548, pp. 423–437, 2021.

[18] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," *arXiv preprint arXiv:2002.06440*, 2020.

[19] J. Goetz, K. Malik, D. Bui, S. Moon, H. Liu, and A. Kumar, "Active federated learning," *arXiv preprint arXiv:1909.12641*, 2019.

[20] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.

[21] T. Li, M. Sanjabi, A. Beirami, and V. Smith, "Fair resource allocation in federated learning," *arXiv preprint arXiv:1905.10497*, 2019.

[22] A. Wilbik, B. Pękala, K. Dyczkowski, and J. Szkoła, "A comparison of client weighting schemes in federated learning," in *IWIFSG'2022, Springer*, to appear.

[23] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[24] R. Sambuc, "Fonctions ϕ-floues: Application á l'aide au diagnostic en pathologie thyroidienne," Ph.D. dissertation, Faculté de Médecine de Marseille, 1975, (in French).

[25] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning–I," *Information Sciences*, vol. 8, no. 3, pp. 199–249, 1975.

[26] H. Zapata, H. Bustince, S. Montes, B. Bedregal, G. Dimuro, Z. Takáč, M. Baczyński, and J. Fernandez, "Interval-valued implications and interval-valued strong equality index with admissible orders," *International Journal of Approximate Reasoning*, vol. 88, pp. 91–109, 2017.

[27] G. Beliakov, H. B. Sola, and T. C. Sánchez, *A practical guide to averaging functions*, ser. Studies in Fuzziness and Soft Computing. Springer, 2016, vol. 329.

[28] M. Komorníková and R. Mesiar, "Aggregation functions on bounded partially ordered sets and their classification," *Fuzzy Sets and Systems*, vol. 175, no. 1, pp. 48–56, 2011.

[29] K. Dyczkowski, A. Wójtowicz, P. Żywica, A. Stachowiak, R. Moszyński, and S. Szubert, "An Intelligent System for Computer-Aided Ovarian Tumor Diagnosis," in *Intelligent Systems'2014*, 2015, pp. 335–343.

[30] B. Pękala, *Uncertainty Data in Interval-Valued Fuzzy Set Theory: Properties, Algorithms and Applications*. Springer, 2019.

[31] H. Bustince, M. Galar, B. Bedregal, A. Kolesárová, and R. Mesiar, "A new approach to interval-valued Choquet integrals and the problem of ordering in interval-valued fuzzy sets applications," *IEEE TFS*, vol. 21, no. 6, pp. 1150–1162, 2013.

[32] R. E. Moore, *Interval analysis*. Prentice Hall, 1966.

[33] ——, *Methods and applications of interval analysis*. SIAM, 1979.

[34] A. Piegat and M. Landowski, "Multidimensional approach to interval uncertainty calculations," in *New Trends in Fuzzy Sets, Intuitionistic: Fuzzy Sets, Generalized Nets and Related Topics, Volume II: Applications*, K. Atanassov *et al.*, Eds., 2013, p. 137–151.

[35] G. Beliakov, A. Pradera, and T. Calvo, "Aggregation functions: A guide for practitioners," in *Studies in Fuzziness and Soft Computing*, 2007.

[36] M. Sugeno, "Theory of fuzzy integrals and its applications," Ph.D. dissertation, Tokyo Institute of Technology, 1974.

[37] "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml