Computers and Electrical Engineering 123 (2025) 110259

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

A federated learning model with the whale optimization algorithm for renewable energy prediction

Viorica Rozina Chifu, Tudor Cioara, Cristian Daniel Anitei, Cristina Bianca Pop, Ionut Anghel [*], Liana Toderean

Computer Science Department, Technical University of Cluj-Napoca, Memorandumului 28, 400114 Cluj-Napoca, Romania

A R T I C L E I N F O

Keywords:
Federated learning
Whale optimization algorithm
Renewable energy prediction
Smart grid
LSTM neural network
K-Means clustering

A B S T R A C T

Federated prediction models for energy prosumers create a global model by combining insights from local machine learning models trained on-site without centralizing the data. For time series energy data, this collaborative approach faces challenges due to the non-IID nature of the data, variations in generation patterns, the high number of model parameters, and convergence issues, leading to poor prediction accuracy. This paper introduces a novel federated learning model, FedWOA, which uses the whale optimization algorithm to determine optimal aggregation coefficients based on the local model weight vectors by pondering the updates considering the model performance and data dimensionality construct the global shared model. To handle the non-IID data the prosumers were clustered based on the similarity of their energy profiles using K-Means. FedWOA improves the prediction quality at the prosumer site, with a 16 % average reduction of the mean absolute error compared to FedAVG while demonstrating good convergence and reduced loss.

1. Introduction

To be effective machine learning models use huge amounts of data in the training processes. As the data may contain sensitive information about individuals ensuring privacy in learning while still obtaining good performance is crucial [1]. This is valid for many sectors where data privacy and decentralization are important such as healthcare, the Internet of Things (IoT), or energy. In the healthcare domain, large volumes of medical data are needed to train machine learning or deep learning algorithms to aid the healthcare professional's decision processes and generate predictions or classifications [2]. Combined with digital twins' models, these techniques can contribute to the envisioned metaverse technologies in the context of simulating medical procedures or allowing augmented virtual interactions [3]. In the energy domain, the transition towards renewable energy shifted the focus to citizens and energy prosumers who need to be actively involved in the energy management process [4]. More decentralized energy ecosystems are emerging that require the active participation of individuals and communities in managing their energy production and consumption [5]. To balance energy supply and demand and manage the local grid stability, access to households' prosumers' energy data is required for training energy prediction machine learning models to support the development of adaptation and proactive optimization strategies [6]. Thus, lately, data-driven machine learning modes have been widely proposed for energy predictions [7,8] like

---

* Corresponding author.
E-mail addresses: viorica.chifu@cs.utcluj.ro (V.R. Chifu), tudor.cioara@cs.utcluj.ro (T. Cioara), anitei.va.cristian@student.utcluj.ro (C.D. Anitei), cristina.pop@cs.utcluj.ro (C.B. Pop), ionut.anghel@cs.utcluj.ro (I. Anghel), liana.toderean@cs.utcluj.ro (L. Toderean).

https://doi.org/10.1016/j.compeleceng.2025.110259
Received 27 September 2023; Received in revised form 16 January 2025; Accepted 7 March 2025

regression models, good in analyzing historical energy and weather data, deep learning models that may identify long-term dependencies in energy data, or other machine learning models that can capture complex relationships between input features and energy production. For better accuracy and effectiveness, they use a combination of features derived from energy data, timestamps, and contextual weather information. However, the households' prosumers' energy data contains sensitive information that is rather private. They need to be carefully handled and protected when used in energy management applications thus techniques like data anonymization [9], encryption [10], and differential privacy [11] have recently emerged. But even with strong privacy and security guarantees the citizen are often reluctant in giving access to their energy data to be moved in centralized silos in the cloud to be further processed and used in training by taking advantage of the cloud's potentially unlimited computing resources. Moreover, in Europe, it is crucial to comply with the General Data Protection Regulation (GDPR) when using machine learning in the cloud aspects like data minimization, retention, deletion, or cross-border transfer affecting the effectiveness of the training processes [12].

In this context, federated learning (FL) models have been proposed lately specifically for cases in which the data owners do not want to share their data due to privacy concerns [13]. They are effective in ensuring the confidentiality and security of prosumers data and they enable multiple local nodes to collaboratively train a machine learning model without sharing their data, which is kept locally [14]. The local machine models are trained on the citizen's devices and are then transmitted to a central server that aggregates and integrates them into a shared global model. The central server sends back the learned global model to the citizen's devices for further refinement and eventually usage. The FL cycle is repeated several times before the global model reaches the desired optimal accuracy. As it is a critical process for the creation of the global federated models several techniques have been proposed lately, the most representative ones being federated averaging of weights (FedAVG), Federated Stochastic Gradient Descent (FedSGD), and their combinations [15]. In the FedAVG case, the aggregation of the local models learned to infer the global shared model is done by averaging (e.g., weighted, median, trimmed, etc.) their model updates in terms of weights while for the FedSGD the local nodes compute and share the gradients of the loss function concerning their local model parameters using their local data. However, even if these approaches generally provide acceptable solutions, there are situations in which the global model obtained with FL has lower accuracy than the locally learned models [16]. This may be caused by the rather simplistic approach of combining the learned weights from the local nodes. Challenges are reported in generating the global model due to local energy data heterogeneity, variations in generation patterns, and the high number of parameters that must be considered especially in the case of deep learning making the averaging not appropriate [17].

Novel global model determination methods need to be researched and developed to deal with the complexity of the problem search spaces in the case of time series models requiring more adaptive aggregation techniques to maintain prediction accuracy across varying temporal patterns [18]. In this context, we consider the bio-inspired population-based heuristics good candidates for performing more optimized aggregation at the central node level than the averaging-based solutions. The population-based metaheuristics use the initialization and iterative evolution phases to explore the solution space and produce high-quality solutions [19]. The initial population is generated and evaluated, with the best individual identified. Search agents, inspired by living organisms, are generated randomly, or using some prior knowledge. A fitness function measures everyone's quality in the search space. Iterative evolution involves updating a population using a mathematical model inspired by animal behavior. Fitness is evaluated to find the best search agent, which is then iteratively updated if a better one is found. The best individual from all populations is returned as the solution. As a result, the population-based metaheuristics guide the search process, efficiently explore the solution space, prevent local optima, use a wide range of algorithms, are not specific to certain problems, and can use domain-specific knowledge through local heuristics controlled by a higher level. These heuristics can be hybridized with different optimization algorithms to reduce computational complexity, improve accuracy, or deal with overfitting problems. Among the metaheuristics proposed in the last years, one of the most promising and that has shown effectiveness in some cases in energy engineering and optimization cases is the Whale Optimization Algorithm (WOA) [20]. It is inspired by the hunting behaviour of humpback whales that balances the exploration and exploitation of the solution space by guiding the whales to search for new regions while converging towards an optimum solution. WOA ensures a good balance between exploration and exploitation, with low computational complexity and a good convergence rate in a variety of optimization use cases [21]. Given these qualities, WOA emerges as a promising candidate for addressing the optimization challenges posed by federated models.

WOA can improve the global model aggregation process in federated learning by determining optimal aggregation coefficients for combining the local model weight vectors going beyond the usage of averaging as in traditional methods such as FedAVG. Therefore, we address the problem of sub-optimal energy prediction accuracy of the global model determined by averaging the local model updates in case of non-IID (independent and identically distributed) energy consumption time series of prosumers. We propose FedWOA, a novel FL model for time series energy data that uses the WOA to construct the global shared prediction model by searching and identifying the optimal set of aggregation coefficients in the search space of the local model weight vectors. We aim to address the specific challenge in federated learning-based energy prediction: while local models often capture well the energy generation patterns of individual prosumers, the global aggregation can smooth out these time-series patterns if not optimized. Moreover, simple federated averaging may smooth the localized peaks and valleys in prosumer energy profiles, which are crucial for accurate time-series prediction. Therefore, we have used household energy data from energy prosumers to train local Long Short-Term Memory (LSTM) models. The vector of weights for each locally trained model is used to create the initial population for the global optimization phase, implemented using the WOA. Afterwards, the algorithm iteratively updates the population by simulating three whale behaviors: searching for prey, encircling the prey, and executing the bubble-net attack. These steps adjust each local prediction model in the population, balancing exploration and intensification to progressively improve accuracy in finding an optimal global federated prediction model. At each iteration, the new prediction models generated are evaluated by a fitness function that aims to minimize the prediction loss on each prosumer. This process ensures that models become progressively more accurate for individual prosumer data as optimization continues. To address the problem of non-IID data in renewable energy production where we need to deal with data heterogeneity and variations in energy generation patterns, we apply the K-Means clustering algorithm to group prosumers with a similar scale of energy data into clusters. Experimental results show that FedWOA achieves better accuracy in energy prediction compared to FedAVG for both Mean Square Error (MSE) and Mean Absolute Error (MAE) metrics.

The paper brings the following contributions to the state of the art:

- An optimal global model inference technique for federated energy prediction of prosumers that captures well the energy generation patterns of individual prosumers minimizing the prediction loss while ensuring that models become progressively more accurate as optimization continues.
- Application of WOA for global model inference in federated learning energy prediction by defining the whale behaviors of searching for prey, encircling the prey, and executing the bubble-net attack for LSTM models trained on prosumer data.
- Use of K-Means clustering algorithm to address the problem of non-IID data in renewable energy production grouping prosumers based on data heterogeneity and variations in energy generation patterns.

The rest of the paper is structured as follows: Section 2 presents the state-of-the-art models for federated machine learning in different domains with a focus on energy prediction cases, Section 3 describes the FL model using WOA for global model construction, Section 4 presents the relevant results in the context of prosumers energy prediction comparing the accuracy with FedAVG models, Section 5 discusses the convergence rate, diversity and overhead of WOA, while Section 6 presents conclusions and future work.

2. Related work

FL techniques have been used primarily in fields, such as healthcare, autonomous driving, IoT, and edge computing, demonstrating good potential to share knowledge and insights among different local parties while simultaneously preserving the privacy of sensitive data. Brisimi et al. [22] integrate a distributed sparse Support Vector Machine (SVM) approach that works well with a small number of features extracted from electronic health records (EHR) data and Primal-Dual Splitting clustering to differentiate between patients who are likely to be hospitalized. Similarly, in [23] sequential and Empirical Bayes based Hierarchical Bayesian federated models are proposed for heart rate prediction. Brophy et al. [24] define a FL-based method for estimating continuous blood pressure from optical photoplethysmogram signals. The global model demonstrates effectiveness for newly collected data and is responsible for updating the client nodes with the accumulated global weights. A similar FL architecture is proposed in [25] for referable diabetic retinopathy classification, while Chen et al. [26] propose a federal transfer-based learning framework for classifying human activities using Convolutional neural network (CNN)-based transfer learning for model customization. FL models were proposed for the detection of lung lesions after COVID-19 from CT images using CNN and federated average [27]. FL has been applied to predict the driver's behavior, the vehicle trajectory, and traffic flow or to detect objects in traffic, such as pedestrians, traffic signs, or other cars, to reduce the risk of accidents and increase people's safety. FL with homomorphic encryption for improved security [28], spike neural networks-based architecture, or one-class support vector machine with federated average aggregation [29] were proposed in the literature. Qi et al. [30] propose a FL solution for traffic flow prediction that stores the local model updates on blockchain to improve global model traceability and uses a differential privacy method with a noise-adding mechanism to improve the confidentiality of the model against attacks of data poisoning. In the IoT and edge computing domains, which are relevant to smart grid management, FL is employed for purposes such as anomaly and intrusion detection, as well as data privacy protection. Wu et al. [31] propose a personalized FL-based framework for IoT applications to manage heterogeneous data while maintaining data privacy. FedAVG is used to learn a global shared model based on the local model updates provided by the client nodes and then is customized based on personal data. Lazzarini et al. [32] use FL for IoT intrusion detection. The client nodes train local models using a shallow artificial neural network and the server node aggregates local models using FedAVG. In the context of resource management or network communication, FL-based solutions address problems such as malicious node detection, unreliable communication, and bandwidth limitation. Fu et al. [33] use blockchain in FL to ensure security for exchanging the local model parameters while Yang et al. [34] put forward three scheduling policies to evaluate the performance of FL in a resource-limited setting with restricted bandwidth and transmission interference. The problem of ultra-reliable low-latency communication in vehicular networks is addressed in [35] by combining FL with the Lyapunov Optimization Algorithm to reduce the delays.

FL strategies have been applied in the context of smart grid management, and these strategies primarily focus on areas such as privacy-preserving energy prediction, demand response, and energy optimization. Fekri et al. use FedSGD and FedAVG to predict the loads in an energy network [36]. During training, the FedSGD sends updates after each round, while the FedAVG uses multiple batches per node. A secure federated deep learning solution for heating load prediction in a building environment is proposed in [37]. To prevent overfitting and maintain confidentiality of data, a dropout regularization technique is used to transmit gradient updates to the central node. Savi et al. [38] apply FL in edge computing scenarios using LSTM models on users' devices and user-specific historical energy consumption data which then are aggregated into a global model on a central node. The novelty lies in the users' clustering based on socioeconomic and consumption similarities to form smaller federations that can be trained separately to produce specific FL models. Fernandez et al. [39] apply FL approaches in different scenarios that focus on the privacy of the data. Each local node adds noise to the data, a differential privacy technique to ensure that attackers cannot reconstruct the original data from gradients communicated over the network. A secure aggregation algorithm is introduced to encrypt the weights of the local models. FL is applied on different campuses to predict the load of combined cooling, heating, and power systems [40]. Several models have been implemented using different strategies and then are comparatively analyzed, the FedAdagrad providing the best prediction results under a

Table 1
Comparison of existing works in FL for smart grids.

Approach	Solution objective	FL strategy	Local ML models	Method for dealing with non-IID data
[36]	Electricity load forecasting on smart meters (short & long term)	FedAvg & FedSGD	LSTM	Adaptive learning rate
[37]	Heating load demand prediction for smart buildings	Cyber-Secure Federated Deep Learning	CNN	N/A
[38]	Energy consumption forecasting at the smart grid edge (short term)	FedAVG	LSTM	Clustering
[39]	Residential buildings electricity load forecasting (short term)	FedAVG	CNN+LSTM	Clustering
[40]	Multi-energy load forecasting	FedAvg, FedAdagrad, FedYogi & FedAdam	CNN-Attention-LSTM	N/A
[41]	Residential buildings electricity load forecasting (short term)	Deep federated adaptation	CNN+BiLSTM	Transfer Learning
[42]	Electricity load forecasting (short term)	FedAvg	LSTM	Clustering
[43]	Consumer Electricity Load Forecasting	Customized FedAdam	ANN	Clustering
[44]	Electricity Load Forecasting	Customized FedSGD	Improved Gate Recurrent Unit (iQGRU)	N/A
[45]	Energy consumption prediction	Customized FedSGD	DNN	N/A
[46]	Wind power forecasting	FedProx	LSTM	Data Normalization
[47]	Building energy forecasting	Secure federated algorithm	ANN	N/A
[48]	Consumer Electricity Consumption	FedAvg	CNN+LSTM	N/A
[49]	PV power forecasting	DP-FedAvg	LSTM	Clustering
[50]	Energy load forecasting	FedAvg	LSTM	Clustering
Current approach	Renewable energy forecasting	FedWOA	LSTM	K-Means clustering

Factitious disorder imposed on another (FDIA) attack. Shi et al. [41] combine FL with transfer learning to improve the accuracy of residential short-term load forecasting. FL is used for issues related to accessibility and privacy of the data, while transfer learning is used to handle the challenge of working with non-identically distributed data. The FL architecture is built on a hybrid model that combines a CNN with a bi-directional LSTM network. Gholizadeh et al. forecast both individual and aggregated electricity demand using FedAVG and LSTM on local nodes [42]. Before training the consumers are clustered using the energy consumption data similarity. Similarly, in [43], two steps are defined to forecast the individual consumer electricity load. First, the local models are trained on nodes and then aggregated into a global model on the central node. Second, the clients customize the global model by training it on their data, as the data is not independent and identically distributed. Liu et al. use a FL framework based on an improved Gate Recurrent Unit for forecasting distributed short-term individual load [44]. The framework guarantees the data privacy of consumers and the usage of computing capacity in edge devices by transmitting the forecasting models during model training instead of the actual load data. BuildFL platform is used to predict the distributed energy resources demand and consumption in [45]. The server node stores the parameters of the global model and sends it to the client nodes to train the local models using their datasets and update the model parameters. The updates are sent back to the global model so that consumer data privacy is preserved. Wang et al. [46] approach the problem of renewable generation uncertainty by proposing a wind power prediction method that uses an adaptive local updating algorithm managed through a secure FL method. Decentralized multiclient functional encryption is employed to achieve data privacy for the FL model that shows better accuracy than classical wind forecasting techniques based on CNN or MLP. Li et al. [47] propose the use of transfer learning for applying FL to building data while maintaining privacy. A secure aggregation algorithm is employed for managing local Artificial Neural Network (ANN) models and the evaluation results show that better training times and accuracy can be obtained using data from the Building Data Genome Project. Using the same dataset a FL method based on CNN-LSTM local nodes to forecast consumer energy consumption is proposed by de Moraes Sarmento et al. [48]. The approach is developed in the edge computing context to be able to run locally on limited computational resources. For model aggregation and orchestration, the authors use the well-known Flower FL framework. Michalakopoulos et al. [49] address privacy-preserving in FL for a residential PV production forecasting technique. The suggested approach combines the weights of locally trained LSTMs through the FedAvg algorithm while integrating a differential privacy aggregator. The non-IID data issue is addressed by using Hierarchical Agglomerative Clustering. Finally, Dogra et al. [50] combine time-series pattern analysis with clustering and FL to preserve consumer data privacy, to improve load prediction accuracy. Affinity clustering groups consumers based on consumption patterns, while FL estimates energy load and improves communication efficiency.

We have identified only a few state-of-the-art FL solutions that focus on global model optimization using heuristics, but neither of these solutions has been applied in the context of smart grid management. This highlights the current gap or lack of specific solutions in the smart grid management domain on using heuristics for FL optimization. Vaiyapuri et al. [51] combine FL with the Bird Swarm Algorithm to perform intrusion detection in IoT environments. The local nodes integrate a social group optimization algorithm kernel extreme learning machine for model training, and the bird swarm algorithm is used to select the most relevant features for training the algorithms. A combination of FL and particle swarm optimization (PSO) was proposed in [52] to enhance network communication and decrease the amount of data being transmitted. Unlike traditional methods where the weights of the models learned on local nodes are sent to the server, this approach uses the PSO algorithm to deduce the score values (i.e., accuracy or loss) of the learned models.

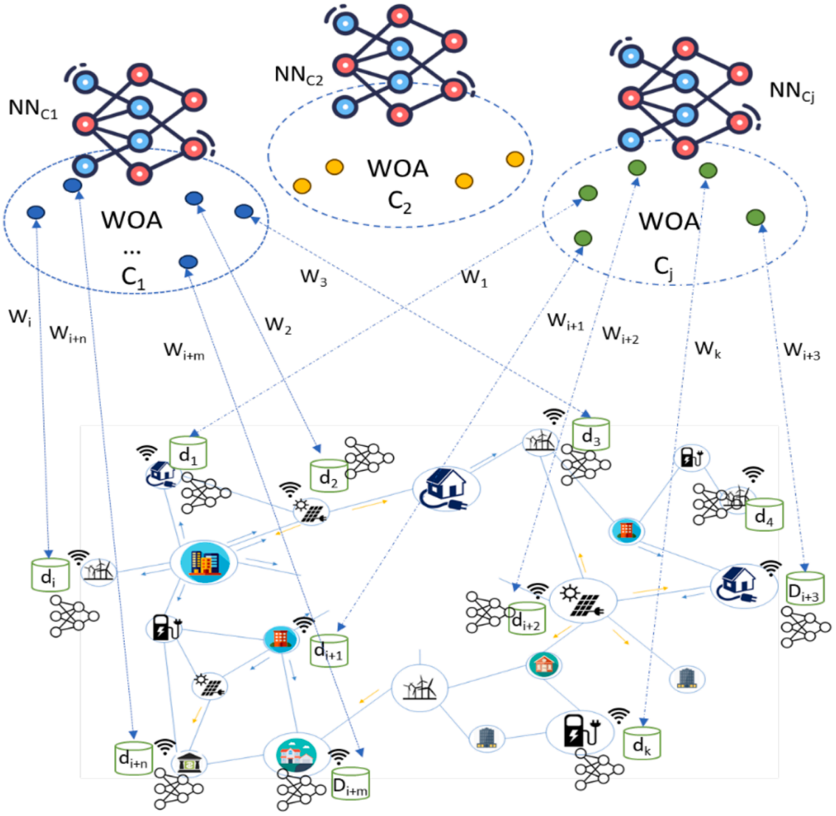


Fig. 1. FL architecture.

Finally, a method for detecting anomalies in pedestrians' walk paths from remote sensing data was proposed in [53], combining FL with Harris Hawks Optimizer (HHO). FL is integrated with a deep faith network architecture to perform anomaly detection, while the HHO is employed to adjust network hyperparameters. In [54] simulated annealing metaheuristic is used for selecting the hyperparameters and participants in aggregation rounds of a FL-based intrusion detection system. The proposed strategy, FedSA optimizes the convergence of the global model without requiring fine-tuning of hyperparameters. Połap et al. [55] approach the image classification problem by using the Red-Fox optimization algorithm (RFOA) metaheuristic for managing aggregation and separation of local models in FL. Additionally, the authors test different metaheuristics (PSO, WOA, dragonfly, etc.) showing that the best results when managing ANN/CNN local models have been achieved by the Red Fox Optimization Algorithm and WOA. Grey Wolf Optimizer algorithm and Markov Chain are combined by FedWolf, a FL approach for improving performance on local long-tailed heterogenous data [56] while in [57] a modified SGD heuristic algorithm based on loss history that uses clients' loss histories to optimize fairness is proposed. One of the few approaches that use WOA for global-level orchestration and aggregation is proposed in the context of mobile data offloading between cellular network devices by Jadav et al. [58]. WOA is used for optimal resource allocation to discover the optimal nodes participating in the group using the social behavior of whales while for decentralized learning a CNN model is employed.

Table 1 provides a comparison of FL approaches for smart grid applications in terms of FL strategies and ML algorithms used. As a conclusion of state-of-the-art analysis, an important challenge is related to generating the global model due to local energy data heterogeneity, and variations in generation patterns. Consequently, when such diverse local machine learning models are combined into a single global model, decreases in energy prediction accuracy can be noticed. To address this challenge, we have adapted and used the WOA to construct the global shared model by identifying the optimal aggregation coefficients for the local node models to enhance the energy prediction accuracy. Another important shortcoming highlighted in the literature is the issue of non-IID data that was addressed in our approach by initially performing clustering of the prosumers to group local nodes with similar scales of energy data. Finally, most of the research approaches tackle model aggregation by using FedAvg, FedSGD, or custom variations of these algorithms through different FL frameworks such as Flower. Few works develop and integrate new and personalized aggregation strategies for energy time series that optimize the general data balancing, communication, and aggregation of local models. In this paper, we propose the integration of a novel FL aggregation algorithm based on WOA for renewable energy prediction at the smart grid level that provides a novel approach to model aggregation which from our knowledge, has not been approached in the literature.

The experimental results show that our WOA-based FL approach yields better results compared with the FedAVG algorithm when applied to the prosumer energy generation prediction problem. Also, FedWOA can efficiently explore the search space to identify a solution close to the global optimum achieving a better loss (fitness) in just ten iterations, whereas FedAVG requires 50 epochs to achieve a similar result.

3. FL model

The FL architecture is presented in Fig. 1 consisting of multiple local nodes geographically distributed. Each node acquires data using IoT metering devices in the form of time-series and stores it locally. Each cluster groups a set of local nodes that have homogeneous data and has a central node responsible for node coordination and federated model aggregation.

3.1. Model assumptions

We have considered that the energy data used in the learning process is distributed across several local nodes and each node is owned by a different user or organization. The data is geographically distributed and stored in$K$ local nodes and not shared with the central server or other participants, respecting the privacy-preserving requirements ϑ_i while allowing for better utilization of resources, easier maintenance, and avoiding a single point of failure:

$$x_i = \{data \mid stored\ by\ n_i\ under\ privacy\ policy\ \vartheta_i \&\ i \in \{1,\ ..K\} \tag{1}$$

The data explored in this model is in the form of a time series, the data points being ordered and indexed by time, t. It represents a stream of energy observations, o, recorded at specific and equal time intervals by IoT measuring devices:

$$data = \{(t, o) : t \in T\ \wedge\ o \in \mathbb{R}\} \tag{2}$$

On each local node the energy data, x_i is used to train a neural network model (*NN*). Afterwards only the local NN model vector of weights is shared to follow the privacy and data security policy ϑ_i:

$$NN_i : y_i = \phi(W_{\mathrm{i}} \cdot x_i + B_i) \tag{3}$$

where ϕ is the activation function, W_{i} is the local model vector of weights and B_i the vector of biases. We assume that in each node the same neural network model is used in training, and the vector of weights W_{i} is represented as:

$$W_{\mathrm{i}} = \left[w_{i,1}, \ldots, w, \ldots w_{i,M}\right],\ i \in \{1, ..K\} \tag{4}$$

where $w_{i,j} \in \mathbb{R}$ is a model parameter and M is the maximum number of weights in the vector of the local model NN_i. The objective of the training process at the level of each local node n_i is to identify a weight vector W_{i} that minimizes a loss function of energy prediction for future observations o:

$$L_i : M \text{ x } K \rightarrow \mathbb{R},\ L_i(y_i, o) = MSE(\phi(\Omega_{\mathrm{i}} \cdot x_i + B_i), o) \tag{5}$$

where MSE is the Mean Squared Error, are o is the actual energy value provided by the IoT device.

In the case of smart grid management scenarios, it is common to have time series energy data in the local nodes that can differ in characteristics and distributions representing different energy profiles. To manage this, we have considered the use of clustering algorithms to group in a cluster C, the nodes n_c that have energy data x_c with similar profiles before the training of the local NN model:

$$C = \{n_c,\ c \in 1..k \wedge f_d\ (x_c, x_{c+1}) < \varepsilon\} \tag{6}$$

where f_d is a function that measures the similarity among energy data of two nodes using various criteria such as maxim, minim, medium values, pattern matching, etc. Also, we considered the clusters to be distinct and there is no overlap among them:

$$\forall\ C_i,\ C_j \in \mathbb{C},\ C_i \cap\ C_j = \varnothing \tag{7}$$

At the level of each cluster, we aim to infer a globally federated model $\overrightarrow{W_C}$ by optimally aggregating each local model vector of weights $\overrightarrow{W_c}$ and pondering the local nodes' contributions based on the local loss values L_c:

$$NN_C :\ \overrightarrow{W_C} = AGG\left(\overrightarrow{W_c}\right),\ \forall\ n_c \in C \tag{8}$$

In each cluster, we have a central node that runs a metaheuristic algorithm to identify the best vector of aggregation coefficients $\overrightarrow{W_C}$ that produces the vector of weights for the cluster-level global model with the minimum loss L_C:

$$L_C = \underset{WOA}{min} L_c(y_c, o),\ for\ all\ n_c \in C \tag{9}$$

The optimization in this case involves searching in the solutions' search space for an optimal vector of aggregation coefficients that minimizes the cluster level loss under certain constraints:

$$(S,\ L,\ \Psi) \tag{10}$$

where S is the set of candidate solutions, L is the loss function that need to be minimized, and Ψ is a set of constraints associated. The domain of aggregation coefficients values must be positive and forms the search space of the optimization problem at the cluster level:

$$\sum \overrightarrow{W_C} = 1 \tag{11}$$

The aggregation process and cluster model construction are optimal in preserving overall accuracy and performance therefore the local models are updated if the cluster model yields a better performance:

$$L_C \geq L_c \ \forall \ n_c \in C \tag{12}$$

3.2. *WOA for distributed learning*

WOA is applied at each cluster to determine the optimal vector of aggregation coefficients by pondering each local model contribution based on loss value and data dimensionality. The K-Means clustering is used to ensure that each node in the cluster has a similar data distribution, and WOA finds the optimal aggregation coefficients that fit the characteristics of the data distribution in that cluster. This strategy reduces the risk that the heterogeneity of data distributions between nodes degrades the performance of the global model, adapting the model to the specificities of each data subset and thus improving the generalization of the final model.

In our approach, an individual (i.e. solution) is represented by a vector of weights $\overrightarrow{W_C}$ associated with the neural network model. This vector of weights defines the position of an individual in the solutions space, and is defined as follows:

$$\overrightarrow{W_C} = \left[w_{C,1}, \ldots, w_{C,M}\right] \tag{13}$$

where $w_{i,j} \in \mathbb{R}$ is a model parameter and M is the maximum number of parameters of the neural network model. We denote S as the space of all possible solutions:

$$S : MxM, S = \left\{ \overrightarrow{W_c} \ \middle| \overrightarrow{W_c} = \left[w_{c,1}, \ldots, w_{c,M}\right], \ w_{i,M} \in \mathbb{R}, \ n_c \in C\right\} \tag{14}$$

At the beginning of the search process, a global vector of weights ($\overrightarrow{W_C}$) is randomly initialized on the central node of the cluster and distributed to each local node n_c. In the initialization process the number of nodes in the cluster C as well as the number of weights in the weight vector associated with the neural network model M are considered. The values $w_{c,M}$ are randomly initialized with values between 0 and 1 while respecting the constraint (11):

$$w_{c,M} = Random\ (0, 1) \tag{15}$$

Each local node n_c uses the weights vector ($\overrightarrow{W_C}$) and performs training on its data, generating a set of locally adjusted weights ($\overrightarrow{W_{n_c}}$):

$$\overrightarrow{W_{n_c}} = f\left(\overrightarrow{W_C}, x_{n_c}\right) \tag{16}$$

where f is the optimization function (which in our case is gradient descent) and x_{n_c} is the data set of the local node n_c.

Thus, the initial population of the WOA algorithm consists of the set of all weight vectors generated at the level of each local node, n_c:

$$\left[\overrightarrow{W_{n_1}}, \overrightarrow{W_{n_2}}, \ldots., \overrightarrow{W_{n_c}}\right] \tag{17}$$

where c is the number of the local nodes of the cluster. Each candidate solution in the population is evaluated using a fitness function that aims at minimizing the loss of the models:

$$fitness = \min \frac{1}{c} \sum_{i=1}^{c} \alpha_i * L_{n_i}\left(\overrightarrow{W_{n_i}}\right) \tag{18}$$

where α_i is a value that is proportional to the number of training data on the local node n_i and $L_{n_i}\left(\overrightarrow{W_{n_i}}\right)$ is the loss function associated with the local node n_i computed for $\overrightarrow{W_{n_i}}$.

The value of α_i is computed with the following formula:

$$\alpha_i = \frac{|x_{n_j}|}{\sum_{k=1}^{c} |x_{n_k}|} \tag{19}$$

where

- $|x_{n_j}|$ represents the cardinality of the data set on the local node n_j
- $\sum_{k=1}^{m} |x_{n_k}|$ is the sum of the cardinalities of the datasets on all local nodes in the cluster.

The fitness values of individuals are used to guide the search process for the best solution.

```
        $p = RANDOM\_NUMBER(0,1);\ l = RANDOM\_NUMBER(-1,1)$
        $\vec{r} = RANDOM\_VECTOR(0,1)$
        $\vec{A} = UPDATE(\vec{r}, a);\ \vec{C} = UPDATE(\vec{r})$
        if (p < 0 .5)
            if (|A| < 1) $\overrightarrow{W_i} = UPDATE(\overrightarrow{W_i},\ \overrightarrow{W_{best}},\ \vec{A}, \vec{C})$
            else $\overrightarrow{ws_i} = TRAIN\_ON\_NODE\ (NN_i, x_i)$
        else $\overrightarrow{W_i} = UPDATE(\overrightarrow{W_i}, \overrightarrow{W_{best}}, b, l)$
    end for
    Foreach $\overrightarrow{W_i}$ in Population do
        $L_{n_i} = COMPUTE\_LOCAL\_LOSS(n_i, \overrightarrow{W_i}, x_i)$
    End for
    $GL_{n_i} = COMPUTE\_GLOBAL\_LOSS(L_{n_i}, \overrightarrow{W_i}, |C|)$
    $\overrightarrow{W_{best}} = SELECT\_MINIMUM\_LOSS(Population,\ GL)$
    $t = t + 1$
end while
return $\overrightarrow{W_{best}}$
End
```

Fig. 2. Federated model optimization using WOA.

The WOA algorithm starts with an initial population of candidate solutions and iteratively updates the population by reproducing the three phases of whale behaviour: search and circling the prey, and bubble-net attacking. These steps are used to update the position of each solution within the population.

The Bubble-net attacking phase has two main steps: shrinking encircling and spiral updating position. In the shrinking encircling step, everyone within the population tries to improve their position relative to the best individual. This means that each member of the population tries to adjust their position ($\overrightarrow{W_i'}$) to approximate the location occupied by the best individual within the population:

$$\overrightarrow{W_i'} = \overrightarrow{W_{best}} - \overrightarrow{A} * \overrightarrow{D} \tag{20}$$

where A and D are vectors of the same length as $\overrightarrow{W'_i}$, while $\overrightarrow{W_{best}}$ is the vector of weights corresponding to the best individual i at the current iteration. We assess the loss of all individuals based on their weight vector to determine the best individual. The two vectors $\overrightarrow{A}$ and $\overrightarrow{D}$ and are defined as follows:

$$\overrightarrow{D} = \left|\overrightarrow{C} * \overrightarrow{W_{best}} - \overrightarrow{W_i}\right| \tag{21}$$

$$\overrightarrow{A} = 2 * a * \overrightarrow{r} - a \tag{22}$$

$$\overrightarrow{C} = 2 * \overrightarrow{r} \tag{23}$$

where $\overrightarrow{W_{best}}$ is the vector of weights that corresponds to the best individual in the current iteration and $\overrightarrow{W_i}$ is the vector of weights that corresponds to the individual i in the current iteration, $\overrightarrow{r}$ is a vector of random values generated in the [0, 1] with the same length as $\overrightarrow{W_i}$, and a is a scalar value that linearly decreases in each iteration from an initial value to zero.

In the spiral position update step, individuals in the population update their position based on their current position and the position of the best individual in the population:

$$\overrightarrow{W_i'} = \overrightarrow{D}' * e^{bl} * \cos(2\pi l) + \overrightarrow{W_{best}} \tag{24}$$

where, b is a constant, l is a random value in $[-1, 1]$, and $\overrightarrow{D}'$ is a weights vector defined as:

$$\overrightarrow{D}' = \overrightarrow{W_{best}} - \overrightarrow{W_i} \tag{25}$$

The Bubble-net attacking phase combines the encircling and spiral updating position steps using the formula below:

$$\overrightarrow{W_i'} = \begin{cases} \overrightarrow{W_{best}} - \overrightarrow{A} * \overrightarrow{D},\ if p < 0.5 \\ \overrightarrow{D}' * e^{bl} * \cos(2\pi l) + \overrightarrow{W_{best}},\ otherwise \end{cases} \tag{26}$$

Table 2
Features of the energy profiles clustering process.

Features		Cluster 0	Cluster 1	Cluster 2
LENGTH	No. of time series data points	145,912	lower bound: 74,504 upper bound: 84,576	lower bound: 134,304 upper bound: 137,280
ACTIVE LOAD	MIN	0.000	0.000	0.000
	MAX	lower bound: 4.925 upper bound: 302.000	lower bound: 0.551 upper bound: 4.268	lower bound: 5.360 upper bound: 121.328
	MEAN	lower bound: 0.327 upper bound: 71.903	lower bound: 0.024 upper bound: 0.779	lower bound: 0.426 upper bound: 9.056
	STD	lower bound: 0.262 upper bound: 50.950	lower bound: 0.023 upper bound: 0.537	lower bound: 0.152 upper bound: 6.615
DHH (yyyyMMddhhmm)	MIN	201,501,010,000	201,610,010,000	lower bound: 201,504,010,000 upper bound: 201,505,010,000
	MAX	201,902,282,345	lower bound: 201,901,312,345 upper bound: 201,902,282,345	201,902,282,345

where p is a random number in [0, 1].

A subset of whales will move towards the current best solution in the population, as in the original WOA, while for the remaining whales (those not participating in the encircling prey phase) a modified update strategy was defined. Instead of adjusting the position of the current search individual towards another randomly chosen one (which might cause heavy loss drift), we propose to update the weights vector of an individual by training it on its local data using gradient updates:

$$\overrightarrow{W_i'} = \overrightarrow{W_i} - \eta * \nabla L\left(\overrightarrow{W_i}\right) \tag{27}$$

$\overrightarrow{W_i}$ is the weights vector in the current iteration, $\overrightarrow{W_i'}$ is the updated weights vector of the individual for the next iteration, η is the learning rate of the local model, and $\nabla L\left(\overrightarrow{W}\right)$ is the gradient of the loss function concerning the local model wights during local training.

The updated strategy has a regularizing effect. Each agent during search space exploration will try to pull the federated solution in the direction of its local optimum, thus learning more from the local data. This exploration will be valid only in the first half of the optimization process, as given by the value of A that decreases from 2 to 0 across a fixed number of iterations. Thus, only when the random number p is less than 0.5 and the magnitude of the vector A is greater than or equal to 1, the local training will be used. These conditions will ensure a balance between exploration and exploitation of the existing solutions. After the exploration and local search phases, the positions of all whales are updated based on the solutions obtained from both the encircling prey and the local gradient-based update.

Fig. 2 shows the pseudocode for the adapted WOA for FL on time series data consisting of initialization and iterative search phases. During the initialization phase, an initial population of a set of weight vectors is built by training models on local nodes using local data (see lines 2–5).

The global loss is calculated for each weight vector in the population using the local loss values. From the set of all weight vectors in the population, the algorithm chooses the weight vector with the lowest overall loss value that represents the best individual (see lines 6–10). In the iterative phase, the algorithm updates the population of weight vectors by applying exploration and exploitation strategies to each vector of weights in the population (see lines 13–22). These strategies aim to explore new solutions and exploit the most promising ones. At the end of each iteration, the algorithm identifies the best weight vector within the updated population by evaluating the global loss values of the weight vectors and selecting the one with the lowest loss as the best vector of weights (see lines 23–27).

4. Evaluation results

To evaluate the effectiveness and prediction accuracy of our federated solution, we used a dataset comprising information from more than 30 small-scale prosumers, each with its own photovoltaic (PV) energy production [59]. The FL model has been applied to renewable energy prediction, allowing multiple local prosumer nodes to collaboratively train a global prediction model while keeping the data decentralized. The data set consists of daily energy production profiles over 4 years with time series values collected every 15 min. To handle non-IID data we organized the prosumers with similar power generation profiles, by applying the K-Means clustering algorithm. The following features were used to capture the temporal variability of the time series records (see Table 2): (i) the number of energy time series data points for each prosumer, (ii) the minimum and maximum limits of active loads, as well as their average energy value and standard deviation from the average, and (iii) the start and end timestamps associated with the energy profiles.

These features enable the formation of homogeneous clusters, reducing the impact of data heterogeneity on the federated learning process by grouping prosumers with similar data. To ensure a balanced distribution of features in the clustering process, the energy values were normalized using the min-max scaling method [60]:

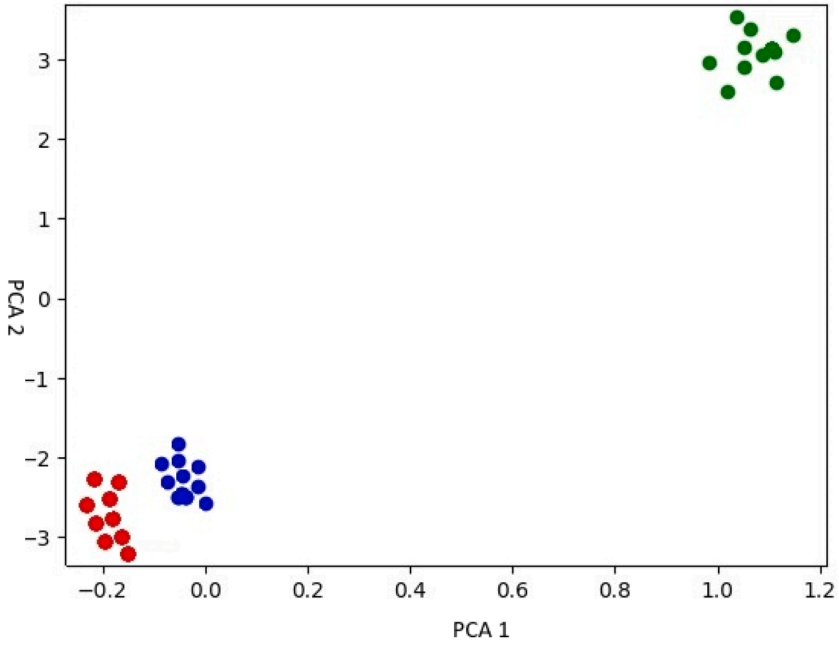


Fig. 3. Prosumers cluster visualization using PCA.

$$E_{norm} = \frac{E - E_{min}}{E_{max} - E_{min}} \tag{28}$$

where E_{norm} is the normalized energy value, E_{max} and E_{min} are the maximum and minimum energy values within the set of energy values that correspond to the four measurements collected at 15 min intervals during the hour.

The min-max scaling to normalize the energy values is a suitable approach when dealing with renewable energy curves that do not follow a Gaussian distribution. The deviation from such distribution can be attributed to several factors, such as clear sky index, azimuth Angle, shading, solar panel placement, or meteorological conditions. The normalized inputs improve the convergence of the local model learning while mitigating the impact of outliers in data. Also, the model can generalize better for new data because the training was done on normalized features that are consistent across different scales. In addition, we used Principal component analysis (PCA) to reduce dimensionality. Dimensionality reduction facilitates differentiation between clusters and simplifies their analysis. After applying K-Means, three homogeneous clusters resulted: one with 9 local nodes, one with 10, and one with 12 (see Fig. 3).

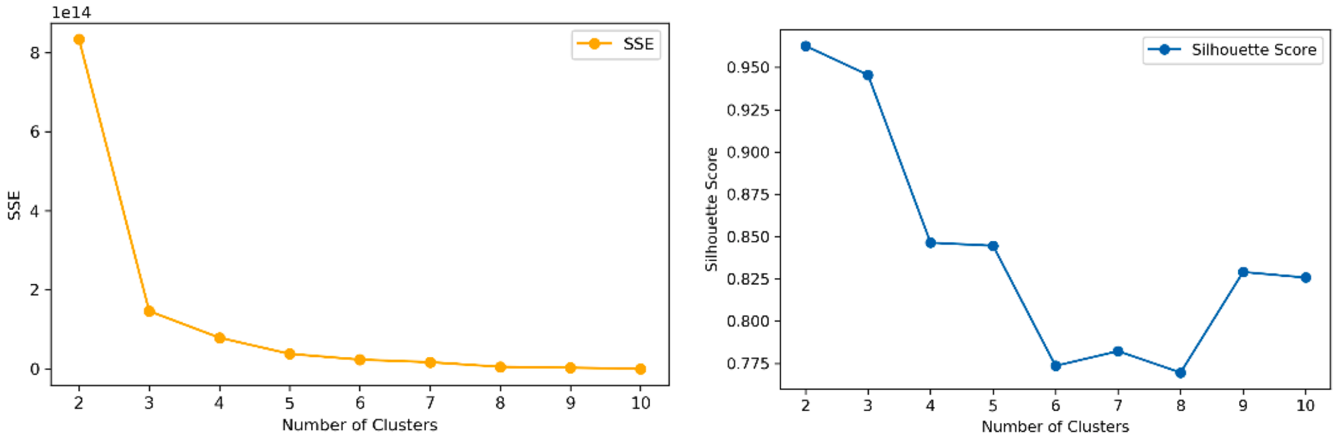
To evaluate the quality of the clustering process we have used Inertia (i.e. the sum of squared Euclidean distances between data points and centroid of each cluster) and Silhouette Score (i.e. score for how well the data points fit in each cluster) as indicators. Fig. 4 presents the indicators values for a number of clusters varying between 2 and 10. The elbow point (first significant decrease of the inertia value) is for a number of 3 clusters which suggests an efficient separation between clusters, without significant overlaps, while maximizing the compactness of the clusters. The high silhouette score indicates cluster cohesion and separation while avoiding overfitting and excessive fragmentation, as for a higher number of clusters the score decreases significantly.

A LSTM neural network is trained on each local prosumer node. It has two layers of LSTM cells, and each LSTM cell has a hidden size of 64, which denotes that it has 64 neurons (see Fig. 5). To prevent the overfitting phenomena, a dropout layer with a rate of 0.5 is applied on each layer. In addition to LSTM layers, we have used two fully connected layers, each of them with 64 features. Between the fully connected layers, a dropout layer with a dropout rate of 0.5 is also placed. The Adam optimizer with a learning rate of 0.0003 is used to optimize the LSTM network.

The neural network architecture was implemented in PyTorch [61]. To efficiently handle the computation of the loss function on each local prosumer node, we leverage PyTorch's chart compilation feature. This allows us to precompile and optimize a computation graph that includes all phases of the loss function, reducing wasted calculations, improving the overall efficiency, and mitigating timing-related issues of our model.

The neural network is used to predict energy values with a lead time of 4. This means that we forecast energy values one hour ahead of the current time because our dataset contains four samples at 15 min intervals for each hour. As input, we will consider sequences with the length set to 96 energy readings, which means that for each prediction, we consider the values of the energy readings from the entire day before, encompassing 24 h. A batch size of 16 is used during training.

The amount of energy data samples (d_i) used to train the LSTM neural network on each local node (n_i) are varying in size and are determined as:



(a) (b)

Fig. 4. Energy profiles clusters metrics: (a) Inertia and (b) Silhouette Score.

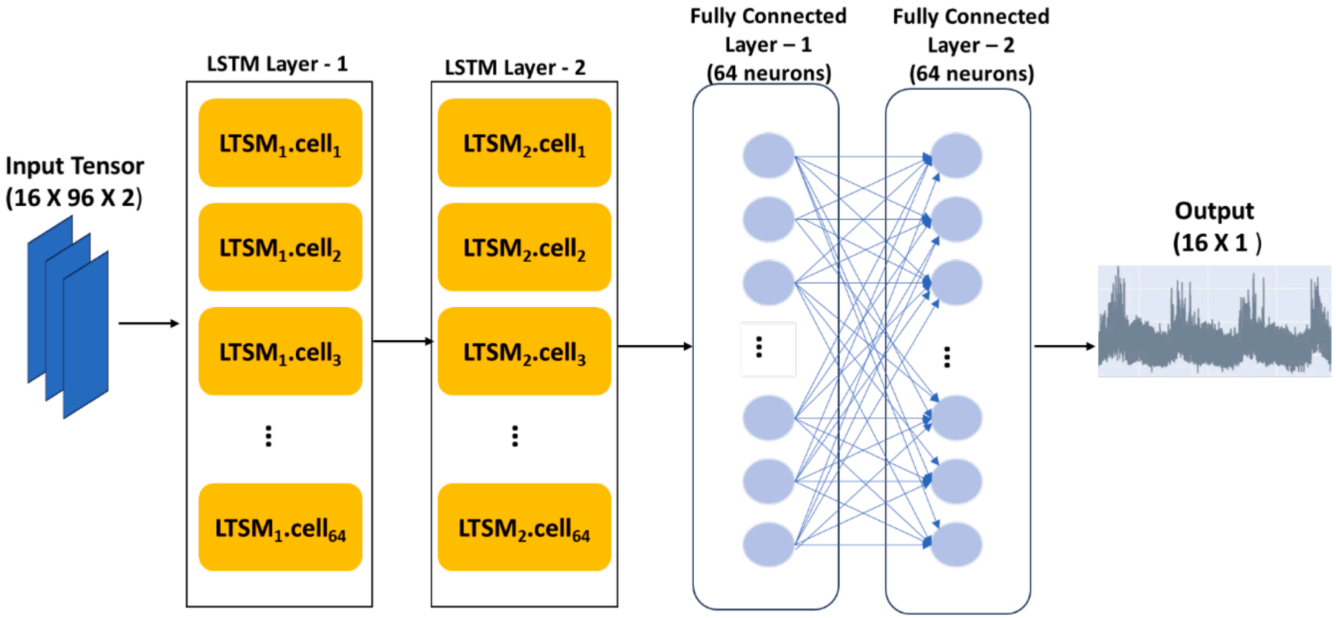


Fig. 5. The neural network architecture used for energy prediction.

Table 3
FedWOA accuracy compared with FedAVG for clusters with ID 0 nodes.

Node ID	MSE				MAE			
	Train		Test		Train		Test	
	FedWOA	FedAVG	FedWOA	FedAVG	FedWOA	FedAVG	FedWOA	FedAVG
P_000004	**0.008284**	0.037937	**0.019984**	0.070258	**0.069696**	0.166328	**0.114504**	0.247359
P_000006	0.006901	**0.002212**	**0.007506**	0.010177	0.078411	**0.033689**	0.069709	**0.069060**
P_000010	0.013317	**0.008977**	0.012887	**0.008231**	0.101358	**0.038855**	0.099493	**0.035694**
P_000013	**0.004946**	0.023111	**0.006870**	0.029826	**0.049691**	0.129460	**0.057932**	0.148497
P_000107	**0.003999**	0.024867	**0.007489**	0.039646	**0.044039**	0.138278	**0.056876**	0.173869
P_0048AB	**0.003603**	0.018498	**0.008010**	0.017770	**0.045826**	0.119876	**0.071150**	0.097609
P_004CB7	0.005758	**0.002851**	**0.006722**	0.007378	0.070555	**0.038802**	0.072343	**0.058888**
P_00506E	**0.009390**	0.029393	**0.009049**	0.028525	**0.079284**	0.129961	**0.075593**	0.128199
P_00509B	**0.016553**	0.039929	**0.023252**	0.049697	**0.104738**	0.148296	**0.116029**	0.158162
P_005169	**0.010605**	0.030274	**0.003903**	0.016410	**0.078645**	0.125971	**0.026814**	0.112718

Table 4
FedWOA accuracy compared with FedAVG for the cluster with ID 1 nodes.

Node ID	MSE				MAE			
	Train		Test		Train		Test	
	FedWOA	FedAVG	FedWOA	FedAVG	FedWOA	FedAVG	FedWOA	FedAVG
P_0005FT	**0.003829**	0.005834	**0.008828**	0.011027	**0.035584**	0.047204	**0.056174**	0.061671
P_0008D9	**0.002339**	0.017111	**0.001618**	0.012010	**0.025652**	0.112864	**0.026775**	0.097650
P_00119D	**0.003394**	0.013606	**0.004573**	0.005517	**0.029788**	0.093690	**0.043059**	0.043576
P_001236	**0.003529**	0.009005	**0.001962**	0.009590	**0.032112**	0.067071	**0.031365**	0.079683
P_001A05	**0.004125**	0.006001	**0.012879**	0.013544	**0.031095**	0.049709	**0.065968**	0.067660
P_00280E	**0.002557**	0.006015	**0.002921**	0.026353	**0.028948**	0.058552	**0.031604**	0.134193
P_003F4D	**0.001874**	0.009889	**0.003749**	0.010357	0.021277	**0.090896**	**0.035885**	0.085141
P_00410C	**0.002537**	0.006280	**0.002959**	0.003047	**0.026860**	0.055113	0.038478	**0.031183**
P_004613	0.005511	**0.027361**	**0.004873**	0.029442	**0.035097**	0.117643	**0.044560**	0.136422
P_0056CD	**0.002485**	0.081751	0.001650	**0.008533**	0.027262	**0.256203**	**0.028540**	0.075401
P_005D9A	**0.001543**	0.001642	0.004156	**0.003016**	0.027777	**0.026606**	0.043456	**0.026326**
P_007C29	**0.009848**	0.048921	0.001579	**0.000679**	**0.039781**	0.189627	0.031253	**0.013336**
P_0080F1	**0.006916**	0.018270	**0.002022**	0.017550	**0.042904**	0.099059	**0.029381**	0.116548

Table 5
FedWOA accuracy compared with FedAVG for the cluster with ID 2 nodes.

Node ID	MSE				MAE			
	Train		Test		Train		Test	
	FedWOA	FedAVG	FedWOA	FedAVG	FedWOA	FedAVG	FedWOA	FedAVG
P_000001	**0.014452**	0.050956	**0.035215**	0.069809	**0.099791**	0.182951	**0.153632**	0.186752
P_000410	**0.005756**	0.042573	**0.012841**	0.033146	**0.062664**	0.188013	**0.093337**	0.137788
P_004A72	**0.012650**	0.020478	**0.030867**	0.001029	**0.096708**	0.104756	0.173106	**0.022626**
P_00680A	**0.007578**	0.012626	**0.021141**	0.044094	**0.077020**	0.100150	**0.118028**	0.165886
P_006A55	**0.006136**	0.043906	**0.023435**	0.095025	0.057145	**0.191260**	**0.122700**	0.265007
P_007209	**0.004594**	0.028107	**0.012049**	0.037411	**0.052845**	0.153329	**0.089790**	0.158108
P_0091D7	**0.002624**	0.037981	**0.007879**	0.044147	**0.037102**	0.186407	**0.068846**	0.186897
P_009FA7	0.015610	**0.010084**	**0.022303**	0.017734	0.108311	**0.067072**	0.134834	**0.076131**
P_00B211	**0.006312**	0.041473	0.014426	**0.003195**	**0.051929**	0.190132	0.116661	**0.040575**

$$d_i = \frac{|dataSet| * ratio_{n_i}}{\sum_{j=1}^{N} ratio_{n_i}} \tag{29}$$

where: $ratio_{n_i}$ is the percentage of energy data from the entire cluster available in the node n_i.

The communication process, at the level of each cluster, was implemented with a specific number of communication rounds between the local nodes and the central node. The process involves the central node initially sending the models to the local nodes, and then the local nodes sending back adjusted model parameters W.

On each cluster, we have used the defined WOA to learn the optimal global model. During WOA optimization for $\frac{T}{2}$ iterations (where T represents the total number of iterations of WOA), each local node had a 50 % chance of sending its updated model parameters due to local training. This stochastic process allows for training on local data at each iteration while estimating the loss by exchanging the

Table 6
Metrics values FedWOA vs FedAVG.

Cluster ID	Solution	MSE_TRAIN	MSE_VAL	MSE_TEST	MAE_TRAIN	MAE_VAL	MAE_TEST
C0	FedWOA	**0.008336**	**0.010612**	**0.010567**	**0.072224**	**0.079599**	**0.076044**
	FedAVG	0.021805	0.025414	0.027792	0.106952	0.110631	0.123006
C1	FedWOA	**0.003888**	**0.005203**	**0.004136**	**0.031087**	**0.036358**	**0.038961**
	FedAVG	0.019360	0.011760	0.011590	0.097249	0.067437	0.074522
C2	FedWOA	**0.008413**	**0.014036**	**0.020017**	**0.071503**	**0.097329**	**0.118993**
	FedAVG	0.032021	0.038669	0.038399	0.151563	0.153327	0.137752

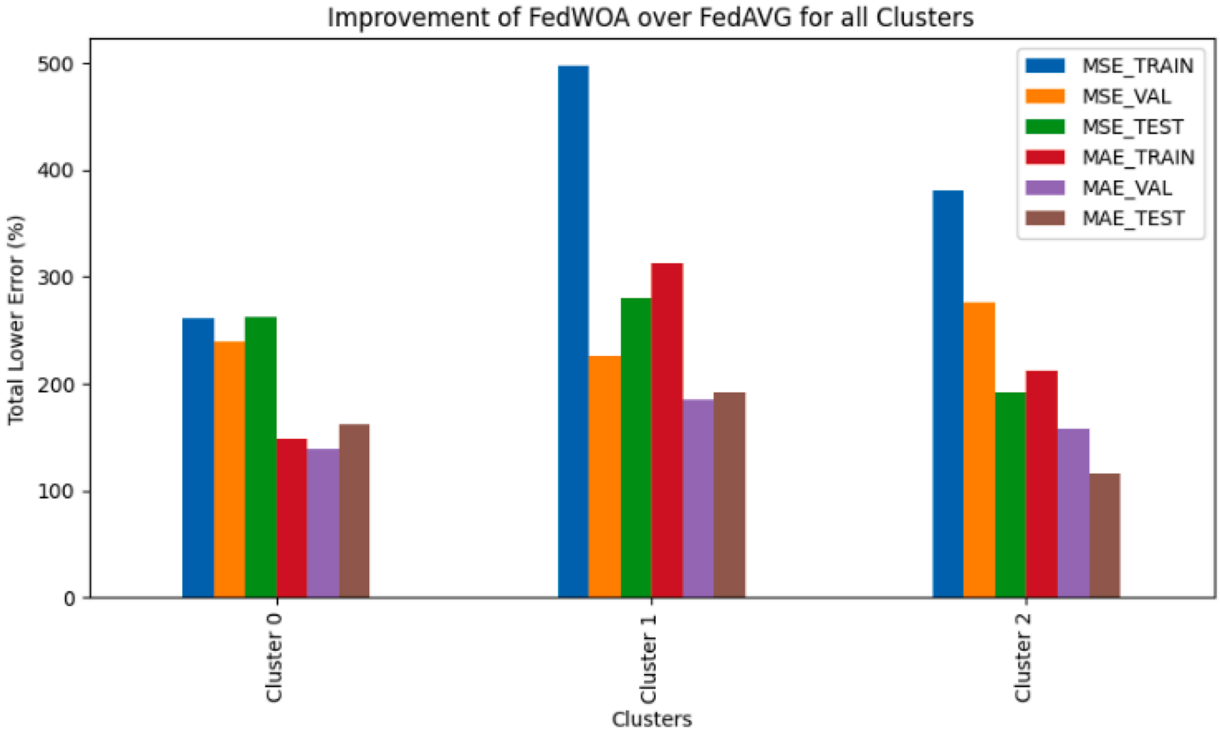


Fig. 6. Improvement of FedWOA over FedAVG.

models among all local nodes within the same cluster. The number of communication rounds is:

$$\gamma = n^2, \ n = |C| \tag{30}$$

where n represents the number of local nodes within the cluster.

To determine the performance of the global prediction model using the best weight vector identified by the WOA algorithm, we calculated the mean square error (MSE) and the mean absolute error (MAE) for the training and testing. The MSE offers a view on the global model's ability to reduce large errors significantly and prioritize precise predictions, while the MAE shows model robustness to outliers and overall accuracy in predicting energy values. The results obtained have been compared with a popular state-of-the-art method the federated average where the model parameters are aggregated by averaging them across the participating nodes [15]. We have maintained a consistent data distribution between the two evaluations to make a fair comparison of the results.

Tables 3–5 present the values obtained for these metrics for each local nod in each cluster considering both FL with WOA and federated average.

The results show that in most of the evaluation cases, our proposed FL model outperforms the federated average being more effective for the problem of energy prediction of prosumers renewable generation.

Table 6 shows comparatively the results on average per cluster of the FL with WOA providing more accurate energy prediction results.

We calculated, for each cluster, the percentage of improvement that our algorithm brings compared to the classic variant using:

$$improvement = \frac{error_{avg}}{error_{WoA}} * 100 \tag{31}$$

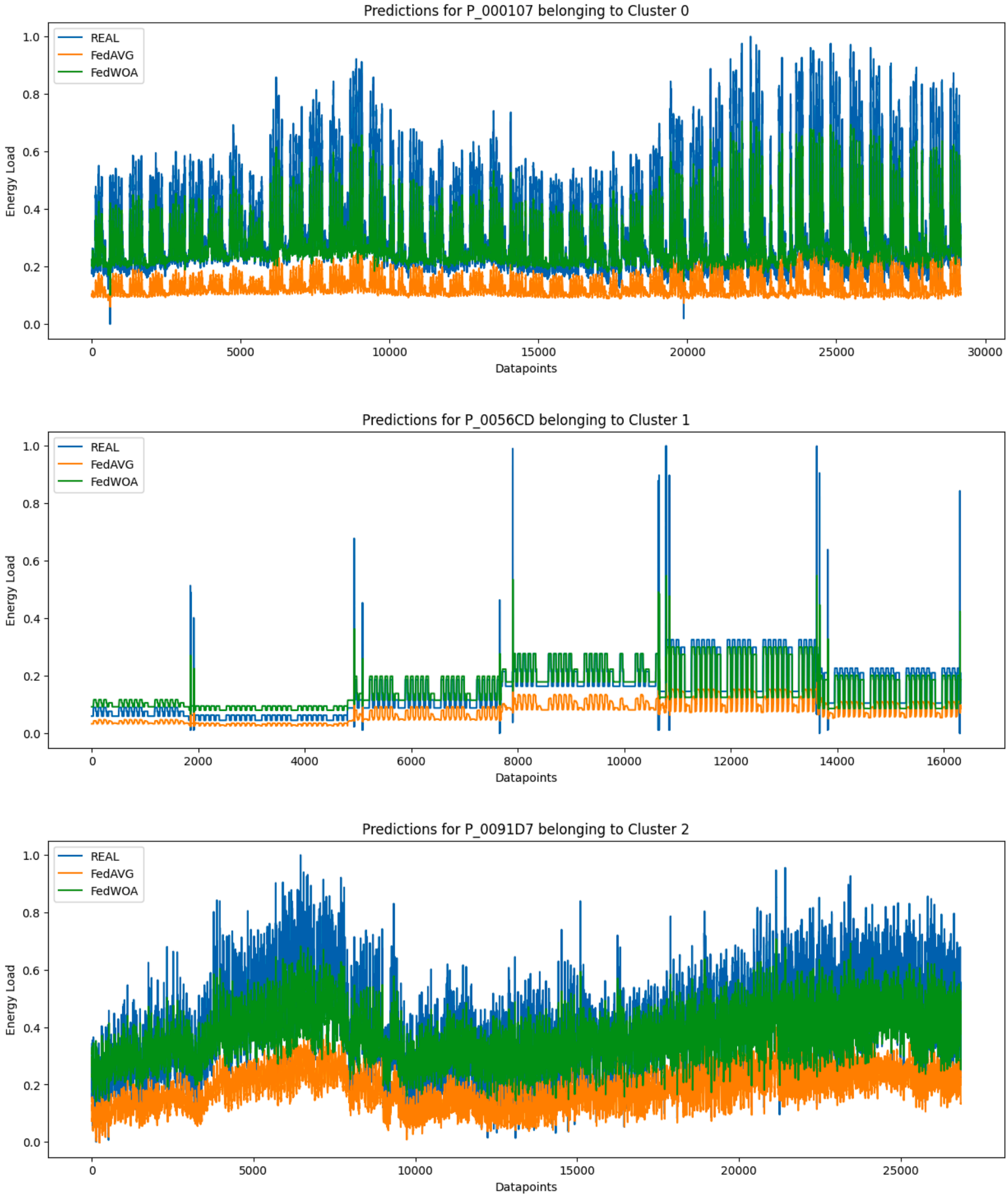


Fig. 7. Energy curve predicted for different prosumers of clusters 0, 1, and 2.

where the error represents the average MSE or MAE error computed for training, validation, or testing. Fig. 6 presents the percentage of improvement computed for MSE, and MAE per cluster of prosumers. The results show the effectiveness of our FedWOA algorithm, compared to the state-of-the-art FedAVG. FedWOA outperforms FedAVG in terms of Mean Square Error and Mean Absolute Error metrics for the federated energy prediction processes. The improvements of FedWOA compared with FedAVG are between, 22,6 % in the worst case and 27,5 % in the best case for MSE in the prediction validation phase and between 13,8 % in the worst case and 18,5 % in the best case for MAE for prediction validation.

Table 7
Data heterogeneity and clustering quality impact on the FedWOA prediction accuracy.

FedWOA applied to	MSE_TRAIN	MSE_VAL	MSE_TEST	MAE_TRAIN	MAE_VAL	MAE_TEST
IDD clusters	0.006879	0.009950333	0.011573333	0.058271333	0.071095333	0.077999333
non-IDD clusters	0.097154514	0.12345703	0.097378044	0.264658586	0.284283468	0.245306446

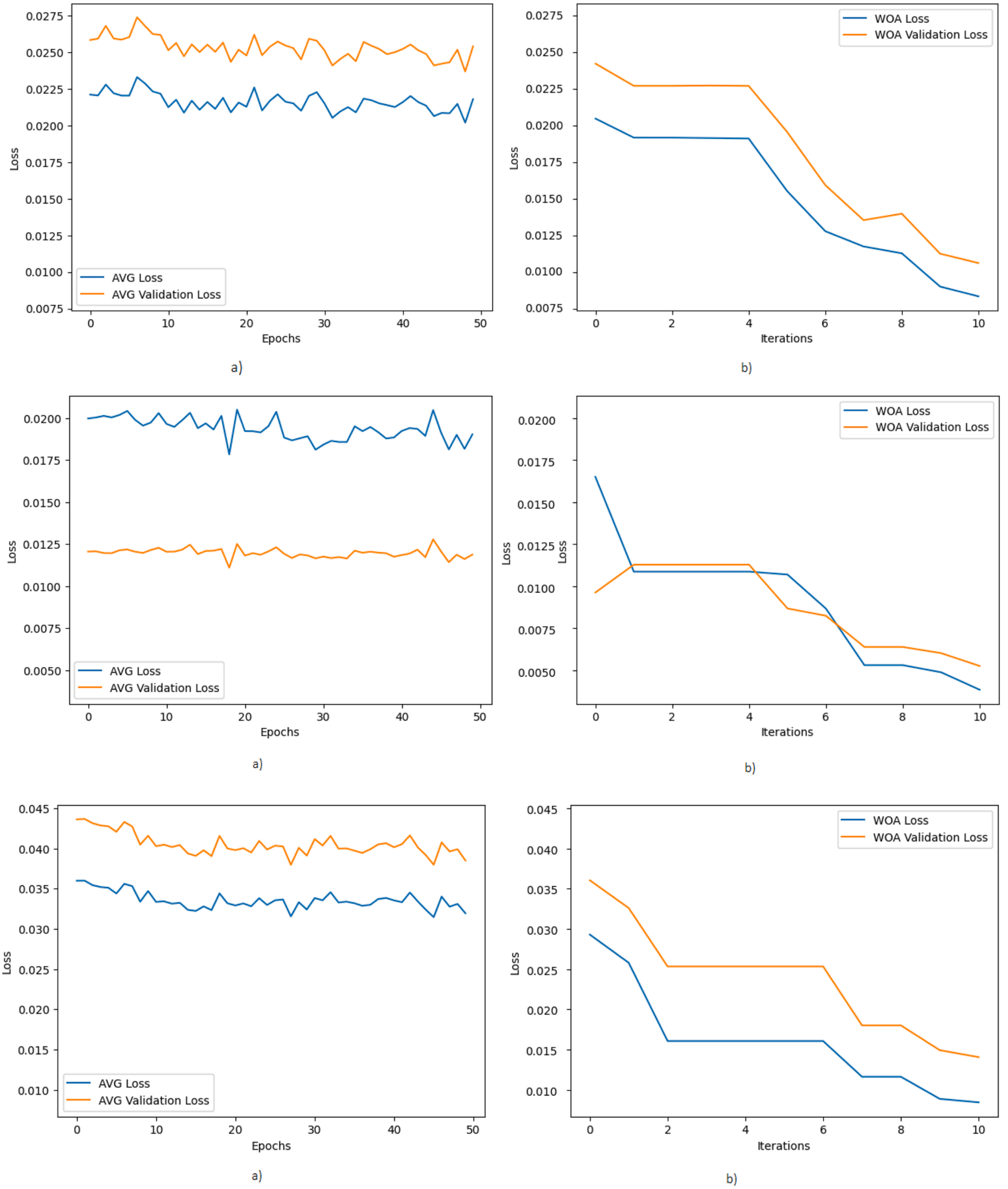


Fig. 8. The fitness variation for each cluster of prosumers.

We also compared FedWOA with the FedAVG considering the predicted energy curve. Fig. 7 shows, for each of the three clusters, the actual energy curve to be predicted, the energy curve predicted with WOA, and the energy curve predicted with the classical FL approach. Analyzing these graphs, we can see that the energy curve predicted with WOA more accurately approximates the real energy curve than the one predicted with the classical FL approach.

To evaluate the impact of energy data heterogeneity and handling non-IID data on the prediction accuracy of the FedWOA we have considered two cases. In the first case, FedWOA was applied to the energy data divided into the optimal number of clusters using K-Means while in the second case, the energy profiles were divided into the same number of clusters, randomly therefore non-IID distribution. We aim to assess how cluster configuration influences the performance of FedWOA in the context of non-IID data and explore the impact of the clustering quality on model accuracy. Table 7 shows a comparison of the prediction accuracy results obtained (i.e., MSE and MAE) for training, validation, and test sets. The results indicate that the heterogeneity and its reflection on the distribution of energy data profiles within clusters can significantly impact the quality of the energy prediction process in our federated learning solution.

5. Discussion

In this section, we discuss the convergence rate, diversity, and overhead of the WOA for FL considering the prediction of the prosumer energy generation use case. Such metrics offer a good view of how fast the WOA provides a good solution for FL-based energy prediction and how diverse are the proposed solutions across different iterations. Experiments have been conducted to comparatively determine the fitness value, convergence rate, and diversity on different configurations.

Fitness value measures the quality of the solution returned by WOA and is computed using the defined loss function L for the local and global federated models. The convergence rate is used to quantify the improvement rate in fitness values during the algorithm iterations and it is computed as [62]:

$$rate_{convergence} = 1 - \left(\left| \frac{L_{opt} - L_t}{L_{opt} - L_{Goal}} \right| \right)^{1/T} \tag{32}$$

where L_{opt} is the best fitness obtained by our algorithm in the T iterations, L_t is the fitness at the current iteration t, L_{Goal} is the target fitness that in our case has the ideal value of (i.e. no loss in the prediction process) and T is the number of iterations. The diversity is used to assess how well the WOA explores the search space for determining the optimal vector of parameters for the global federated model out of the local models learned [63]. The diversity is computed for each iteration using the Euclidean distance:

$$div = \sum_{i=1}^{N} \sum_{j=1}^{M} \sqrt{\left(\overrightarrow{W_i} - \overrightarrow{W_j} \right)^2} \tag{33}$$

where $\overrightarrow{W_i}$ and $\overrightarrow{W_j}$ are the positions of the individuals (i.e., weights vectors) in the search space.

Fig. 8 shows the evolution of average loss (i.e., the fitness value of the best individual) during FedWOA iterations for each of the three clusters. Analysing these plots, we notice that, for each of the three clusters FedWOA can find, after ten iterations, a solution with a fitness value very close to zero thus it efficiently explores the search space to identify a solution that is very close to the global optimum. However, in the case of the second cluster, we notice that although in the first iteration, the loss validation function decreases, starting with the second iteration, it increases and remains constant until interaction with the fourth one, when it begins to decrease again. This is due to the way the dataset was divided, namely that the validation dataset has less noise than the training dataset.

We also made a comparison in terms of the average training loss and the average validation loss of the FedWOA compared with FedAVG. The results show the evolution of average training loss and average validation loss from one epoch to another for each of the three clusters in the case of the FedAVG. Analysing these plots, we notice that our FedWOA can reach a better loss (i.e., a better fitness) in only ten iterations as opposed to the 50 epochs ran in the case of FedAVG. Also, it can be observed that in the FedAVG approach, the loss slightly reduces and continues to fluctuate from one epoch to another, while in the case of FedWOA, the loss reduces significantly and continues to decrease from one iteration to another.

Fig. 9 represents the average convergence rate chart for the FedWOA solution for each of the three clusters. We observe an improvement in fitness value from one iteration interval to the next. Also, there are no fluctuations in the convergence rate, but only intervals in which it stagnates and then increases again and approaches the optimal value. This indicates that the algorithm is converging fast towards the best solution. The plateau recorded in the graphs means that the best individual can remain the same for several iterations until another better individual is identified (i.e., an individual with a lower loss value).

The diversity measurements for each cluster are presented in Fig. 10. What is observed from all three graphs corresponding to the three clusters in which the WOA is applied, is that there are stages where the algorithm's diversity decreases indicating exploitation of the current search area, or stages where the algorithm's diversity increases, indicating exploration of new areas. More exactly, we observe that in the first iterations, the algorithm mainly performs an exploration of the search space, and therefore the diversity increases, while in the last iterations, the algorithm focuses on exploitation of the search space to converge towards the best solution.

To determine the computational efficiency and potential trade-offs with prediction accuracy we measured the execution time and the memory usage of the central node for FedWOA and FedAVG. Fig. 11 shows the execution time over a number of 10 iterations for both methods. FedWOA exhibits a higher execution time compared to FedAVG, due to its additional algorithmic complexity, which

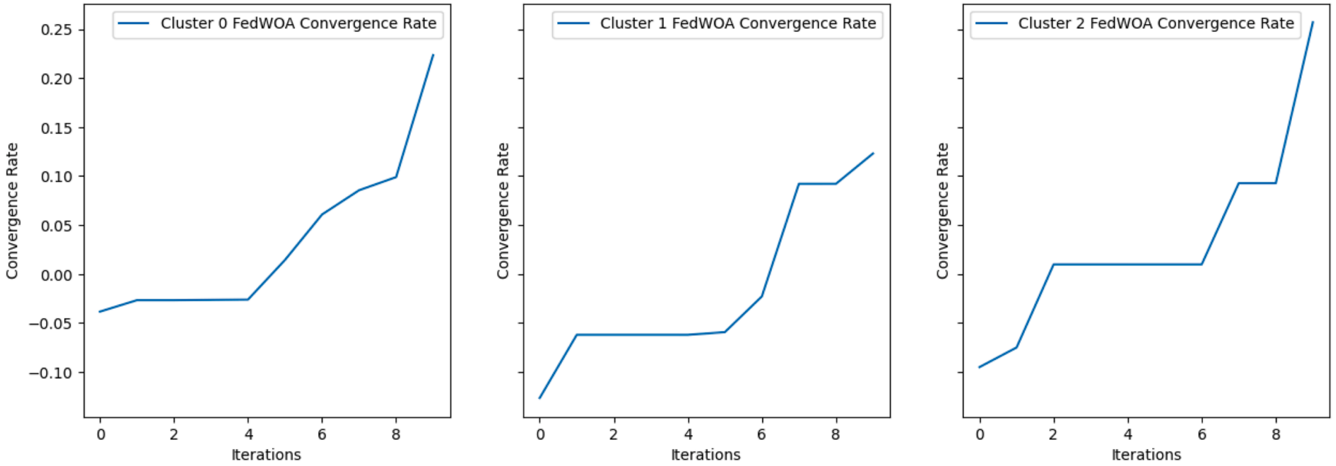


Fig. 9. Convergence rate of FedWOA for each of the three clusters.

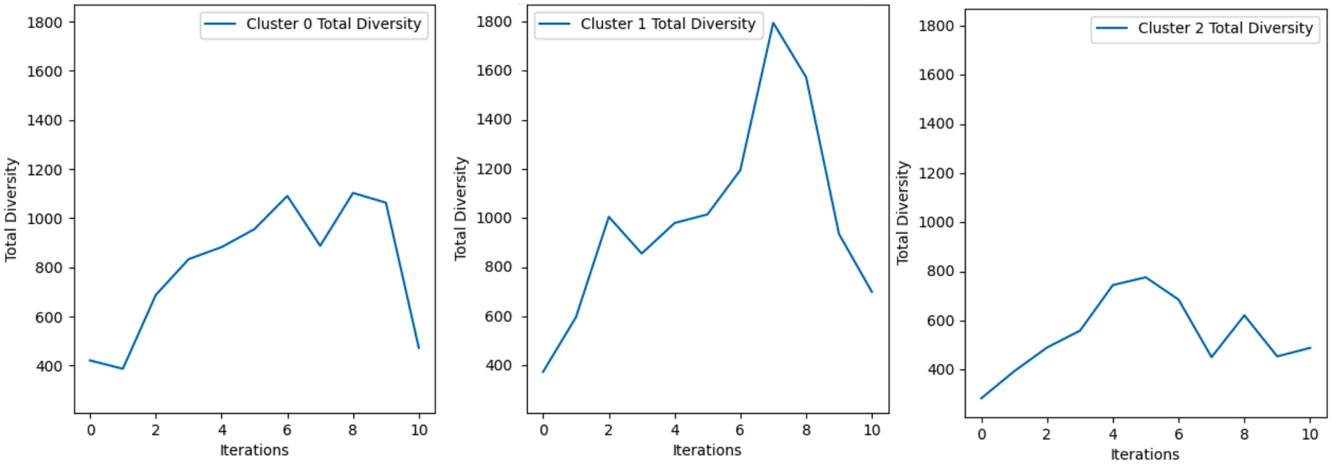


Fig. 10. Diversity of FedWOA for each of the three clusters.

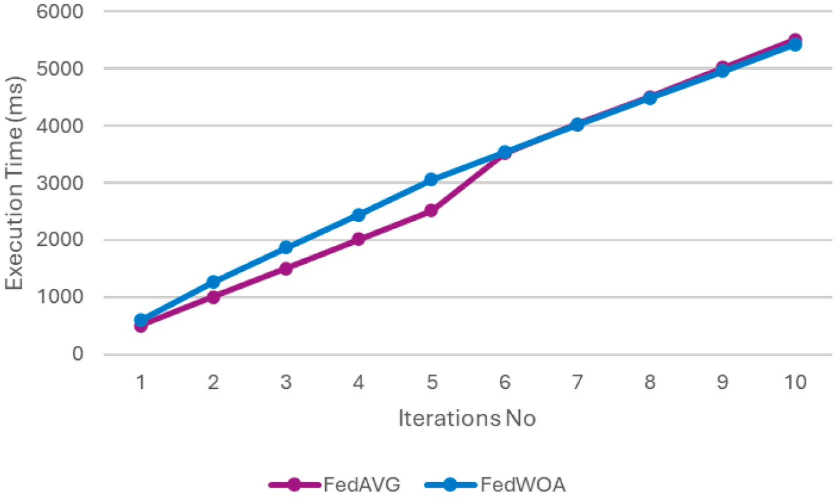


Fig. 11. Execution time for FedWOA vs FedAvg.

Table 8
Average memory usage and execution time for a local node.

Federated Algorithm	Memory Usage (MB)		Execution Time (s)	
	Train	Validation	Train	Validation
FedWOA	0.710957	2.921821	14.96	79.31818182
FedAVG	2.919885	2.930132	360	24

involves performing optimization operations for each cluster. This increase in processing time reflects the higher computational cost of FedWOA but results in improved model prediction accuracy. An analysis of the ratio between accuracy and execution time suggests that, although FedWOA involves higher computational complexity, it provides a good trade-off between accuracy and efficiency. Therefore, FedWOA is well-suited in scenarios where accuracy requirements are critical and can compensate for additional computational requirements. A solution to reduce this computational cost is process parallelization. In this case, each node would perform the training and loss calculation in parallel, which would distribute the workload and reduce the total execution time. This approach would also allow for a more efficient use of resources, reducing the negative impact on the computational cost of the algorithm.

We measured the peak memory usage in the central node at key computational steps, for both FedWOA and FedAVG. The memory usage is significantly higher for the FedWOA algorithm as the central node has to keep the optimization data (e.g. the population, computed loss, best model trained, etc.). In the case of the FedAVG algorithm, the central node has to store only the global model aggregated weights. The computed average peak memory usage in a federated iteration for the FedWOA was 0.14 MB, whilst for the FedAVG was 0.007MB. Finally, we also computed the time spent on average over 10 iterations by a local node on training and validation phases for both algorithms as well as the memory usage (see Table 8). The average memory usage and time spent on the training phase by a local node for a single iteration is lower in the case of FedWOA as the training nodes are selected in the exploration phase and there are several iterations over the federated process in which a node does not participate in training.

6. Conclusions

In this paper, we have proposed FedWOA, a FL model for predicting renewable energy production based on time series energy data from local prosumer nodes. The novelty of our approach is the usage of WOA to aggregate global prediction models from the weights of local LSTM neural network models. FedWOA identifies and aggregates the near-optimal vector of weights to construct the global shared model, communicates with the local nodes to improve prediction quality, and employs K-means clustering to group the prosumers with similar scale of energy data for addressing non-IID data issues. Compared to FedAVG, our solution provides better energy prediction accuracy with an average improvement of about 25 % for MSE and 16 % for MAE in the prediction validation phase, showing good convergence and reduced loss. Our findings reveal that using decentralized time series energy data sources, and collaborative global model optimization with WOA can lead to precise forecasts for small-scale energy prosumers. A drawback of the proposed method is the high execution time since at each iteration of the WOA algorithm, the loss function must be recomputed for everyone in the population. This process becomes computationally expensive, especially when the population size is large or when a cluster contains many local nodes. Another disadvantage of the proposed method is that the clustering quality has a direct impact on the FL performance. When the non-IID data on local nodes have very different distributions, the quality of the clustering can be affected. This can lead to the formation of heterogeneous clusters containing nodes with significantly different data distributions, or unbalanced clusters, and applying the WOA algorithm will result in sub-optimal local models that aggregated will negatively affect the overall FL performance. These drawbacks will be further tackled as future work for improving our FedWOA solution.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work has been conducted within the DEDALUS project, grant number 101103998, funded by the European Commission as part of the Horizon Europe Framework Programme.

Data availability

The data that has been used is confidential.

References

[1] El Mestari SZ, Lenzini G, Demirci H. Preserving data privacy in machine learning systems. Comput Secur 2024;137:103605. ISSN 0167-4048.
[2] Divya SV, Venkadesh P, Shiny KV, Nels SN. In: Prasanth A. Classifying and Predicting Covid-19 during pregnancy using RNN and ML Techniques, 2024 International Conference on Social and Sustainable Innovations in Technology and Engineering (SASI-ITE), Tadepalligudem, India; 2024. p. 54–9.
[3] Editor(s) Balasubramaniam S, Sumina S, Satheesh Kumar K, Prasanth A. Chapter 7 - machine learning based models for implementing digital twins in healthcare industry. In: Kumar Dhanaraj Rajesh, Vashishtha Srishti, Sathyamoorthy Malathy, Balusamy Balamurugan, Cengiz Korhan, editors. Metaverse technologies in healthcare. Academic Press; 2024. p. 135–62. PagesISBN 9780443135651.
[4] Toderean L, Chifu VR, Cioara T, Anghel I, Pop CB. Cooperative games over blockchain and smart contracts for self-sufficient energy communities. IEEE Access 2023;11:73982–99.
[5] Mehdinejad M, Shayanfar H, Mohammadi-Ivatloo B. Peer-to-peer decentralized energy trading framework for retailers and prosumers. Appl Energy 2022;308:118310. ISSN 0306-2619.
[6] Pop C, Cioara Antal M, Anghel I, Salomie I, Bertoncini M. Blockchain based decentralized management of demand response programs in smart energy grids. Sensors 2018;18:162.
[7] Antal M, Toderean L, Cioara T. Anghel I. Hybrid deep neural network model for multi-step energy prediction of prosumers. Appl Sci 2022;12:5346.
[8] Li X, Wang Z, Yang C, Bozkurt A. An advanced framework for net electricity consumption prediction: incorporating novel machine learning models and optimization algorithms. Energy 2024;296:131259.
[9] Adewole KS, Torra V. DFTMicroagg: a dual-level anonymization algorithm for smart grid data. Int J Inf Secur 2022;21:1299–321.
[10] Mu T, Lai Y, Feng G, Lyu H, Yang H, Deng J. A user-friendly attribute-based data access control scheme for smart grids. Alexand Eng J 2023;67:209–17. PagesISSN 1110-0168.
[11] Mitrea D, Cioara T, Anghel I. Privacy-preserving computation for peer-to-peer energy trading on a public blockchain. Sensors 2023;23:4640.
[12] Goldsteen A, Ezov G, Shmelkin R, Moffie M, Farkash A. Data minimization for GDPR compliance in machine learning models. AI Ethics 2022;2:477–91.
[13] Antal M, Mihailescu V, Cioara T, Anghel I. Blockchain-based distributed federated learning in Smart grid. Mathematics 2022;10:4499.
[14] Rajesh M, Ramachandran S, Vengatesan K, Dhanabalan SS, Nataraj SK. Federated learning for personalized recommendation in securing power traces in smart grid systems. IEEE Trans Consum Electron 2024;70(1):88–95. Feb.
[15] Wang Y, Zobiri F, Mustafa M, Nightingale J, Deconinck G. Consumption prediction with privacy concern: application and evaluation of Federated Learning, Sustainable Energy. Grids Networks 2024;38:101248. ISSN 2352-4677.
[16] Li T, Sahu AK, Talwalkar A, Smith V. Federated Learning: challenges, methods, and Future directions. IEEE Signal Process Mag 2020;37(3):50–60. May.
[17] Khan LU, Saad W, Han Z, Hossain E, Hong CS. Federated learning for Internet of Things: recent advances, taxonomy, and open challenges. IEEE Comm Surveys Tutorials 2021;23(3):1759–99.
[18] Qi P, Chiaro D, Guzzo A, Ianni M, Fortino G, Picciali F. Model aggregation techniques in federated learning: a comprehensive survey. Future Generat Comput Syst 2024;150:272–93.
[19] Pop CB, Cioara T, Anghel I, Antal M, Chifu V, Antal C, Salomie I. Review of bio-inspired optimization applications in renewable-powered smart grids: emerging population-based metaheuristics. Energy Reports 2022;8:11769–98. ISSN 2352-4847.
[20] Mirjalili S, Lewis A. The whale optimization algorithm. Adv Eng Software 2016;95:51–67. ISSN 0965-9978.
[21] Arcas GI, Cioara T, Anghel I. A whale optimization for cloud–Edge-offloading decision-making for smart grid services. Biomimetics 2024;9:302.
[22] 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;112:59–67.
[23] Fang L, Liu X, Su X, Ye J, Dobson S, Hui P, Tarkoma S. Bayesian inference federated learning for heart rate prediction. In: Proceedings of the International Conference on Wireless Mobile Communication and Healthcare, Virtual Event, 19 November 2020; Springer: Cham, Switzerland; 2020. p. 116–30. 105.
[24] Brophy E, De Vos M, Boylan G, Ward T. Estimation of continuous blood pressure from ppg via a federated learning approach. Sensors 2021;21:6311.
[25] Lo J, Timothy TY, Ma D, Zang P, Owen JP, Zhang Q, Wang RK, Beg MF, Lee AY, Jia Y, Sarunic MV. Federated learning for microvasculature segmentation and diabetic retinopathy classification of OCT data. Ophthalmol Sci 2021:1.
[26] Chen Y, Qin X, Wang J, Yu C, Gao W. FedHealth: A federated transfer learning framework for wearable healthcare. IEEE Intell Syst 2020;35:83–93.
[27] Dou Q, So TY, Jiang M, Liu Q, Vardhanabhuti V, Kaissis G, Li Z, Si W, Lee HHC, Yu K, Feng Z, Dong L, Burian E, Jungmann F, Braren R, Makowski M, Kainz B, Rueckert D, Glocker B, Yu S, Heng PA. Federated deep learning for detecting COVID-19 lung abnormalities in CT: A privacy-preserving multinational validation study. NPJ Digit Med 2021;4:60.
[28] Mohammed MA, Lakhan A, Abdulkareem KH, Zebari DA, Nedoma J, Martinek R, Kadry S, Garcia-Zapirain B. Homomorphic federated learning schemes enabled pedestrian and vehicle detection system. Int Things 2023;23:100903. ISSN 2542-6605.
[29] Xie K, Zhang Z, Li B, Kang J, Niyato D. Efficient federated learning with spike neural networks for traffic sign recognition. IEEE Trans Veh Technol 2022;71. No. 9, September.
[30] Qi Y, Hossain MS, Nie J, Li X. Privacy-preserving blockchain-based federated learning for traffic flow prediction. Futur Gener Comput Syst 2021;117:328–37.
[31] Wu Q, He K, Chen X. Personalized federated learning for intelligent IoT applications: A cloud-edge based framework. IEEE Open J Comput Soc 2020;1:35–44.
[32] Lazzarini R, Tianfield H, Charissis V. Federated Learning for IoT intrusion detection. AI 2023;4:509–30.
[33] Fu X, Peng R, Yuan W, Ding T, Zhang Z, Yu P, Kadoch M. Federated learning-based resource management with Blockchain Trust assurance in smart IoT. Electronics 2023;12:1034.
[34] Yang HH, Liu Z, Quek TQS. Poor HV. Scheduling policies for federated learning in wireless networks. IEEE Trans Commun 2020;68:317–33.

[35] Samarakoon S, Bennis M, Saad W, Debbah M. Distributed federated learning for ultra-reliable low-latency vehicular communications. IEEE Trans Commun 2020;68:1146–59.

[36] Fekri MN, Grolinger K, Mir S. Distributed load forecasting using smart meter data: federated learning with recurrent neural networks. Int J Electr Power Energy Syst 2022;137.

[37] Moradzadeh A, Moayyed H, Mohammadi-Ivatloo B, Aguiar AP, Anvari-Moghaddam A. A secure federated deep learning-based approach for heating load demand forecasting in building environment. IEEE Access 2021;10:5037–50.

[38] Savi M, Olivadese F. Short-term energy consumption forecasting at the edge: a federated learning approach. IEEE Access 2021;9. 95 949–95 969.

[39] Fernandez JD, Menci JP, Lee CM, Rieger A, Fridgen G. Privacy preserving federated learning for residential short-term load forecasting. Appl Energy 2022;326. 119 915.

[40] Zhang G, Zhu S, Bai X. Federated learning-based multi-energy load forecasting method using cnn-attention-lstm model. Sustainability 2022;14(19). 12 843.

[41] Shi Y, Xu X. Deep federated adaptation: an adaptative residential load forecasting approach with federated learning. Sensors 2022;22:3264.

[42] Gholizadeh N, Musilek P. Federated learning with hyperparameter-based clustering for electrical load forecasting. Internet Things 2022;17:100470.

[43] Wang Y, Gao N, Hug G. Personalized federated learning for individual consumer load forecasting. CSEE J Power Energy Syst 2022;9(1).

[44] Liu Y, Dong Z, Liu B, Xu Y, Ding Z. FedForecast: A federated learning framework for short-term probabilistic individual load forecasting in smart grid. Int J Electr Power Energy Syst 2023;152:109172. ISSN 0142-0615.

[45] Venkataramanan V, Kaza S, Annaswamy AM. DER forecast using privacy-preserving federated learning. IEEE Int Things J 2023;10(3):2046–55. 1 Feb.1.

[46] Wang Y, Guo Q. Privacy-preserving and adaptive federated deep learning for multiparty wind power forecasting. IEEE Trans Ind Appl 2024.

[47] Li J, Zhang C, Zhao Y, Qiu W, Chen Q, Zhang X. Federated learning-based short-term building energy consumption prediction method for solving the data silos problem. Build Simul 2022;15:1145–59.

[48] de Moraes Sarmento EM, Ribeiro IF, Marciano PRN, Neris YG, de Oliveira Rocha HR, Mota VFS, da Silva, Villaça R. Forecasting energy power consumption using federated learning in edge computing devices. Internet Things 2024;25:101050. ISSN 2542-6605.

[49] Michalakopoulos V, Sarantinopoulos E, Sarmas E, Marinakis V. Empowering federated learning techniques for privacy-preserving PV forecasting. Energy Reports 2024;12:2244–56. PagesISSN 2352-4847.

[50] Dogra A, Anand A, Bedi J. Consumers profiling based federated learning approach for energy load forecasting. Sustain Cities and Society 2023;98:104815. ISSN 2210-6707.

[51] Vaiyapuri T, Algamdi S, John R, Sbai Z, Al-Helal M, Alkhayyat A, Gupta D. Metaheuristics with federated learning enabled intrusion detection system in Internet of Things environment. Expert Systems Wiley 2022.

[52] Park S, Suh Y, FedPSO Lee J. Federated learning using particle swarm optimization to reduce communication costs. Sensors 2021;21:600.

[53] Alohali MA, Aljebreen M, Nemri N, Allafi R, Duhayyim MA, Alsaid MI, Alneil AA, Osman AE. Anomaly detection in pedestrian walkways for intelligent transportation system using federated Learning and Harris Hawks optimizer on remote sensing images. Remote Sens 2023;15:3092.

[54] Neto HNC, Dusparic I, Mattos DMF, FedSA Fernande C. Accelerating intrusion detection in collaborative environments with federated simulated annealing. In: 2022 IEEE 8th International Conference on Network Softwarization (NetSoft), Milan, Italy; 2022. p. 420–8.

[55] Połap D, Woźniak M. Meta-heuristic as manager in federated learning approaches for image processing purposes. Appl Soft Comput 2021;113:107872. Part AISSN 1568-4946.

[56] Pu J, Fu X, Dong H, Zhang P, Liu L. Dynamic adaptive federated learning on local long-tailed data. IEEE transactions on services computing, vol; 2024. p. 1–14. PrePrints 5555.

[57] Mollanejad A, Navin AH, Ghanbari S. Fairness-aware loss history based federated learning heuristic algorithm. Knowl-Based Syst 2024;288:111467. ISSN 0950-7051.

[58] Jadav NK, Tanwar S. Whale optimization-orchestrated Federated Learning-based resource allocation scheme for D2D communication. Ad Hoc Networks 2024; 163:103565. ISSN 1570-8705.

[59] Antal C, Cioara T, Antal M, Mihailescu M, Mitrea D, Anghel I, Salomie I, Raveduto G, Bertoncini M, Croce V, Bragatto T, Carere F, Bellesini F. Blockchain based decentralized local energy flexibility market. Energy Reports 2021;7:5269–88. PagesISSN 2352-4847.

[60] Ahsan MM, Mahmud MAP, Saha PK, Gupta KD, Siddique Z. Effect of data scaling methods on machine learning algorithms and model performance. Technologies 2021;9:52.

[61] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Köpf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J, Chintala S. PyTorch: an imperative style, high-performance deep learning library. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems. Red Hook, NY, USA: Curran Associates Inc.; 2019. p. 8026–37.

[62] He J, Guangming L. Average convergence rate of evolutionary algorithms. IEEE Trans Evol Comput 2016;20(2):316–21.

[63] Olorunda O, Engelbrecht AP. Measuring exploration/exploitation in particle swarms using swarm diversity. In: Proc. IEEE Congr. Evol. Comput.; 2008. p. 1128–34. Jun.

Viorica Chifu received her Ph.D. degree in Computer Sciences in 2010 with the thesis “Autonomic Web Service Composition”. She is an associate professor of Computer Science at the Technical University of Cluj-Napoca and works in smart grid decentralization and bio-inspired optimization. Viorica is a reviewer for high-impact journals and a PC member in international conferences.

Tudor Cioara received a Ph.D. degree in Computer Science in 2012 and a habilitation degree in 2019. He is currently a full professor at the Technical University of Cluj-Napoca and the leader of the Distributed Systems Research Laboratory coordinating several European projects on smart grids, blockchain, and energy communities. His current research interests focus on decentralized control and machine learning.

Cristian Anitiei received a bachelor’s degree in computer sciences in 2023 from Technical University of Cluj-Napoca with a thesis in the domain of federated learning research. He is currently a master’s student at the Artificial Intelligence track. His research interests include optimization using bio inspired metaheuristics, machine learning and distributes systems.

Cristina Bianca Pop received a Ph.D. in Computer Science in 2013 and she is associate professor at the Technical University of Cluj-Napoca. Her research interests are related to Bio-inspired Computing, Ambient Assisted Living, Machine Learning, Big Data Analytics, and Energy Systems. She has reviewed papers for high-impact journals (Applied Sciences, Sensors, Electronics) and was a guest editor of special issues.

Ionut Anghel is a full professor of Computer Science at the Technical University of Cluj-Napoca. He is a reviewer for high-impact journals (Future Generation Computer Systems, Applied Energy, Computers & Electrical Engineering, Energies, Energy Efficiency, etc.) and a PC member in international IEEE conferences (ICCP, IEEE, CSE, ENBIS, etc.). Ionut is also a co-editor for several journals (e.g. Sensors, Sustainability).

Liana Toderean received the master’s degree in computer science from the Technical University of Cluj-Napoca, in 2022, where she is currently pursuing the Ph.D. degree, with a focus on targeting technological enablers for decentralized demand response considering small-scale prosumers and energy communities. She is also a Teaching Assistant with the Technical University of Cluj-Napoca.