1290

UNIVERSIDADE Đ
COIMBRA

Nuno Alexandre Gonçalves Mendes

Federated Learning for the prediction of Net Energy Demand in Communities of Buildings

Dissertation under the Master of Science Degree in Electrical and Computer Engineering supervised by Professor Doctor Pedro Moura, and Professor Doctor Jérôme Mendes, and presented to the Department of Electrical and Computer Engineering, Faculty of Science and Technology, University of Coimbra.

September 2022

University of Coimbra

Faculty of Sciences and Technology

Department of Electrical and Computer Engineering

FEDERATED LEARNING FOR THE PREDICTION OF NET ENERGY DEMAND IN COMMUNITIES OF BUILDINGS

Nuno Alexandre Gonçalves Mendes

Dissertation under the Master of Science Degree in Electrical and Computer Engineering supervised by Professor Doctor Pedro Moura, and Professor Doctor Jérôme Mendes, and presented to the Department of Electrical and Computer Engineering, Faculty of Science and Technology, University of Coimbra.

September 2022

Acknowledgements

My biggest grateful thanks to Professor Doctor Pedro Moura, and Professor Doctor Jérôme Mendes. I would like to thank my supervisor, Professor Doctor Pedro Moura, for his teachings, encouragement and the opportunity to realize this dissertation. To my co-supervisor, Professor Doctor Jérôme Mendes, for his support, motivation and for the knowledge imparted.

I would like also to thank Professor Doctor Humberto Jorge for the help given to collect the first data from the Polo 2 campus. To researcher Rodrigo Salles, who always have been there to help me and to the ML@GridEdge team for all the very important support and help given.

This dissertation is the last step of my course, and to get here I always counted on my friends. They are part of my academic life and all their support, motivation given and the good times passed by made this last stage possible. To them, I am sincerely thankful.

For last, I will like to write my last acknowledgment in Portuguese: *Aos meus pais, irmã e irmão por toda a força, motivação, suporte, encorajamento e pela muita paciência que tiveram. Muito obrigado por tudo o que me deram nestes últimos anos que culminaram nesta dissertação. Sei que sem vocês muito dificilmente teria conseguido, estarei sempre muito grato a vós, são incríveis! Por último agradecer aos meus avós, e restante família por terem sido fantásticos durante esta caminhada.*

This research was supported by FCT through the project ML@GridEdge (UTAP-EXPL/CA/0065/2021).

Abstract

The future Transactive Energy (TE) communities rely on economic and control mechanisms for managing consumption and generation through enabling end-use energy trading. The optimization of such communities needs strategies where the prediction of the net energy demand is the main key to achieving better performance levels of the control systems. Such prediction usually relies on net-demand information, but each building can have additional private information, since the actual building automation systems have the capability of collecting large amounts of data, which has a critical role in the prediction systems.

Data generated through building automation systems are mainly considered private. In such a context, Federated Learning (FL), in recent years, has been used in many different areas where the main purpose is data protection. A FL strategy can be used for the prediction systems in TE communities, in order to use their private data and improve the forecast models. This strategy also enables a collaborative community where all participants can ensure benefits from the participation of other participants, without issues in terms of sharing private data.

A novel approach for predicting net-demand in TE communities based on FL is proposed in this dissertation. The developed framework allows the integration of third-party data providers, and the coordination by a server of two distinct forecast systems (one for generation and the other for demand). It ensures the forecast of net-demand in an indirect way by using the forecasted demand and generation, and collaborative learning among the buildings without sharing private data.

The proposed approach was tested using two different scenarios. The first scenario is in Portugal and uses data collected from several buildings on a campus of the University of Coimbra. This dataset was used to test the first stage of the developed framework that has only the forecast of demand and uses local data of generation to evaluate the net-demand. The second scenario uses a dataset provided by the National Renewable Energy Laboratory (NREL) for buildings in California, United States of America. Such a dataset has buildings with and without a photovoltaic (PV) system, and the buildings that have such a system have the data of the power demand separated from the data relative to the generation, as needed to test the developed framework.

The results present a high level of accuracy and adaptability to different situations, for instance, seasonal variations. The framework developed has a generalization associated that allows being introduced in different community types (a university campus and a residential community was tested), and also different types of buildings (with or without a PV system integrated).

Keywords: Federated Learning, Artificial Neural Networks, Net Energy Demand Forecast, Transactive Energy Community, Smart Grid.

Resumo

As futuras comunidades de energia transativa dependem de mecanismos de económicos e de controlo para a gestão do consumo e da geração através da comercialização de energia entre edificios. Para otimizar as comunidades de energia são cruciais os sistemas para a previsão do consumo de energia líquida. Tais sistemas de previsão por norma usam dados históricos da energia líquida, mas podem ser melhorados ao incluir também informações privadas dos edifícios.

Os sistemas de automação em edifícios permitem a recolha de dados em grandes quantidades que podem serem usados pelos sistemas de previsão, mas estes dados são maioritariamente classificados como privados. Nesse contexto, o *Federated Learning (FL)*, tem sido abordado em diversas áreas com o objetivo principal de proteger os dados privados dos utilizadores. Nas comunidades de energia pode ser aplicado um sistema de FL para implementação de sistemas de previsão, para aproveitar os dados privados associados a cada edifício para melhorar a capacidade de aprendizagem. O FL também permite que não se perca a colaboração entre os utilizadores finais, que podem beneficiar do treino ocorrido noutro utilizador sem terem que abdicar da privacidade.

Nesta dissertação é proposta uma nova abordagem para a previsão da energia líquida em comunidades de energia com base num sistema de FL. A estrutura implementada prevê a integração de entidades terceiras como fornecedores de dados, e dois sistemas de previsão (um para o consumo e outro para a geração), ambos geridos pelo mesmo servidor, que irão permitir que se faça a previsão do consumo líquido de energia elétrica.

Foram usados dois cenários para testar a estrutura proposta. O primeiro cenário é de um campus universitário em Portugal, o Pólo 2 da Universidade de Coimbra, e foram utilizados 6 edifícios para a recolha de dados. Este cenário serviu como primeiro teste à estrutura implementada, onde estava a apenas implementado o sistema de previsão do consumo, e utilizou-se os dados de geração para se determinar o consumo líquido de energia. O segundo cenário utilizou um conjunto de dados fornecidos pelo *National Renewable Energy Laboratory*, onde foi selecionado o estado da Califórnia, EUA. Este conjunto de dados possuí edifícios com e sem sistemas de geração fotovoltaica, sendo que as que possuem tais sistemas têm os dados da geração e do consumo separados. Esta separação de dados torna assim possível o teste completo da estrutura proposta.

Dos resultados obtidos foi possível observar que os sistemas de previsão atingiram um bom nível de precisão e de adaptabilidade, por exemplo a variações sazonais. A estrutura implementada permite ainda ser introduzida em diferentes tipos de comunidades, como mostrado com o teste que mostra a sua capacidade de generalização e adaptação. De notar ainda que no segundo cenário, ainda foi mostrado o uso da estrutura numa comunidade onde estavam integrados edifícios que poderiam possuir ou não um sistema de geração.

Palavras-chave: Aprendizagem Federada, Redes Neuronais Artificiais, Previsão do Consumo de Energia, Comunidade de Energia Transativa, Rede Elétrica Inteligente.

“As armas e os Barões assinalados”

— Luís de Camões

Contents

List of Acronyms

ANN Artificial Neural Network.

B2C Business-to-Consumer.

CMA Client Main Algorithm.

DAU Data Aggregator Unit.

DChE Department of Chemical Engineering.

DECE Department of Electrical and Computer Engineering.

DR Demand Response.

DSM Demand-Side Management.

DSO Distribution System Operator.

EU European Union.

FAE Federated Average Equation.

FL Federated Learning.

GEE Greenhouse Gases Emissions.

HFL Horizontal Federated Learning.

ID Identity.

LMTA Local Model Train Algorithm.

LSTM Long Short-Term Memory.

MAE Mean Absolute Error.

MAPE Mean Absolute Percentage Error.

MDMS Meter Data-Management System.

ML Machine Learning.

MSE Mean Squared Error.

NIST National Institute of Standards and Technology.

NREL National Renewable Energy Laboratory.

OHE One Hot Encoding.

PS Power System.

PSA Participant Selector Algorithm.

PV photovoltaic.

RMSE Root Mean Square Error.

RWG Random Weights Generator.

SG Smart Grid.

TE Transactive Energy.

VFL Vertical Federated Learning.

List of Nomenclatures

$\mathcal{W}^b_{r_{\{d,g\}}}$ Weights returned by the Client Main Algorithm (CMA) algorthim in the end of round r, for a building b, relatively to the system d or d (dimensionless).

$\Phi^b_{r_{\{d,g\}}}$ Scaler of the dataset $\mathcal{D}^b{}_{pvt_{\{d,g\}}}$ (dimensionless).

α Control variable used by Algorithm 2 (dimensionless).

$\mathcal{B}$ List of buildings Identity (ID)s that are connected to the server (dimensionless).

$\mathcal{B}_s$ Batch size parameter of the Artificial Neural Network (ANN) (dimensionless).

$\mathcal{B}_r$ List of clients IDs that are selected for a round, also known as participants (dimensionless).

$\mathcal{D}^b{}_{pvt_{\{d,g\}}}$ Dataset of private features in building b, relatively to a forecast system (dimensionless).

$\mathcal{D}_{cf}$ Dataset of common features (dimensionless).

$\mathcal{D}_{control}$ Control dataset in the server (dimensionless).

$\mathcal{E}$ Number of epochs in a ANN (dimensionless).

$\mathcal{K}$ Number of participants per round (dimensionless).

$\mathcal{L}^b_{r_{\{d,g\}}}$ Loss returned by the CMA algorthim in the end of round r, for a building b, relatively to the system d or d (dimensionless).

$\mathcal{L}_{min_{actual}}$ Minimal loss computed by a client local, previously saved on control dataset ($\mathcal{D}_{control}$) (dimensionless).

$\mathcal{L}_{min}$ Minimal loss computed by a client on the local train, in the actual round (dimensionless).

$\mathcal{N}_{ch}$ Number of cells in the hidden layers of the ANN (dimensionless)(dimensionless).

$\mathcal{N}_{ci}$ Number of cells in the input layer of the ANN (dimensionless)(dimensionless).

$\mathcal{N}_{fo}$ Number of future observations to be predicted (dimensionless).

$\mathcal{N}_{po}$ Number of past observations to make the forecast (dimensionless)(dimensionless).

$\mathcal{R}_d$ Maximum rounds allowed for demand forecast system d)(dimensionless).

$\mathcal{R}_g$ Maximum rounds allowed for generation forecast system g)(dimensionless).

$\mathcal{R}_n$ A round number pre-defined (dimensionless).

$\mathcal{R}_{\{d,g\}}$ Maximum rounds allowed for a global FL train in each system (dimensionless).

$\mathcal{T}_n$ A train number pre-defined (dimensionless).

$\mathcal{V}_{train}$ Boolean variable that validate a occurred training process by the ANN (dimensionless).

$\mathcal{W}_{r_{\{d,g\}}}^{\mathcal{B}_r}$ Variable that concatenates all weights derived from the two systems in all buildings (dimensionless).

$\mathcal{W}_{u_{\{d,g\}}}$ Computed weights from the Federated Average Equation (dimensionless).

$\mathcal{W}_{begin_{\{d,g\}}}$ Weights pre-saved on the server for begin a FL train (dimensionless).

d Forecast system of power demand (dimensionless).

g Forecast system of generation (dimensionless).

r_n Actual round number on FL train (dimensionless).

rst Boolean to reset or not the global train (dimensionless).

p_r Previous round participated on FL train by a client)(dimensionless).

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and Scope

Planet Earth is facing very drastic climatic changes, and the cooperation of all countries is needed to prevent drastic consequences. European Union and other countries signed an agreement, the Paris agreement with the main objective to limit global warming to below 2ºC compared to pre-industrial levels and to make efforts to limit the increase to 1.5°C [1].

According to [2], the Power System (PS) is responsible for about 29% of Greenhouse Gases Emissions (GEE). Therefore, one of the goals is the decarbonization of the PS and to ensure it, according to the same source, it is possible to reduce GEE by 99% related to the PS based on actual technologies and the knowledge about the future evolution related. The increasing penetration of renewable generation into the grid and into buildings has been leading to new challenges, namely intermittency, variability of the generation, and the mismatch between the profile of generation and demand in buildings.

In such a context, Smart Grid (SG) are the evolution of power systems with the integrated communications and technologies of Edge Computing bringing more intelligence and enabling micro-systems to operate independently of the main control station. EC is assent in 4Cs: communication, computation, caching, and control [3], allowing new approaches, as well as encouraging the appearance of new technologies, like energy storage systems, electrical vehicle charging, Demand Response (DR), and distributed generation.

Distributed Generation allows a building to generate energy to supply its own demand, passing from consumers of electricity to prosumers, consumers and producers, and consequently active agents of the electricity market [4]. However, typically, the profiles of PV generation and the electricity demand in buildings have a strong mismatch [5], and new technologies and methodologies are needed to ensure the required flexibility for the coordination between the available generation and demand. For instance, Figure 1.1, presents the mismatch between the demand and the generation registered on 1 January 2020, in a building at the University of Coimbra (Department of Electrical and Computer Engineering).

The flexibility technologies, such as battery energy storage [5] and DR [6], can be used to ensure the optimization of the energy prosumption at the building level, but also provide services to the electrical grid. However, to ensure such an objective a traditional approach with the Distribution System Operator (DSO) predicting the load at the substation level and adapting the grid resources is not enough. The prediction of the prosumption in each individual building is essential, together with the knowledge of the availability of flexibility resources.

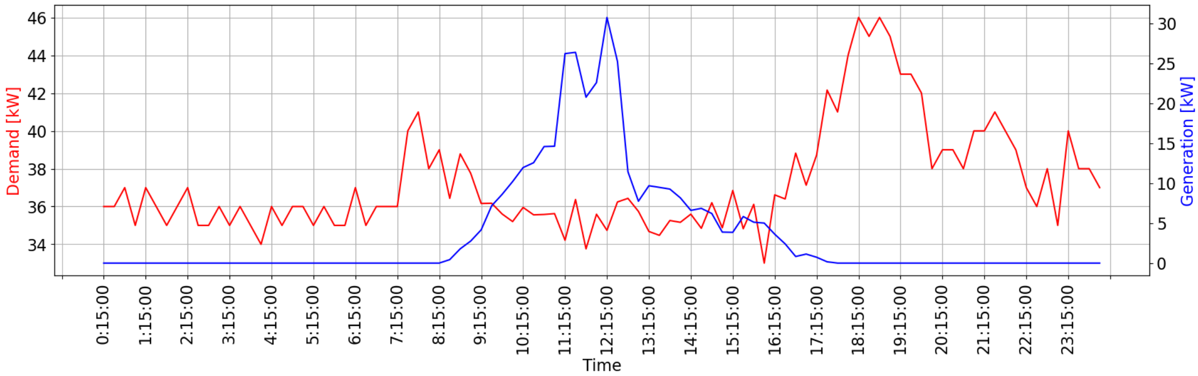


Figure 1.1: Example of demand and generation values recorded during a day.

These issues are more critical in future Transactive Energy (TE) systems that are defined as economic and control mechanisms for managing consumption and generation through enabling end-use energy trading [7]. In such a context, to improve the accuracy of prosumption prediction, collaboration between multi buildings is needed to create a multi-agent TE. However, that implies the sharing of data, which most of the participants do not want to share, since it can reveal sensitive commercial information and buildings strategy for participating in the TE market.

The most accurate solutions in the state of the art are based on aggregated loads at the substation level. However, for the management of communities using flexibility resources at the building level, the prediction of prosumption at every building level is required. Different works that ensure prediction at the building level are based on the net-metering data, losing the differentiated impact on demand and generation, do not take into account privacy issues, do not use private data (e.g. occupancy data), and do not make use of collaboration systems.

FL is a Machine Learning (ML) technique that allows training models in decentralized edge devices without sharing the data [8], reaching the possibility to use private data to get better results and creating collaborative systems without putting the problem of revealing sensitive data between the users. This approach has been mainly used in mobile and edge device applications. However, the use in other areas has been increasing, including communications [9], cloud computing [10], text recognition [11], smart manufacturing [12], health [13] or security [14]. FL for the forecast of net energy consumption in building communities makes it possible to achieve better prediction results in different energy grid levels.

1.2 Objectives

This work intends to develop a novel federated machine learning model for predicting the temporal energy needs of future connected communities. To increase the reliability of the prediction of net-demand two distinct forecast systems should be used, one for predicting demand, and the other dedicated to the prediction of the generation. The combination of these two predictions will return the prediction of the net-demand.

The proposed solution lends itself well to a multi-agent structure of connected energy communities where agents are willing to collaborate to improve their prediction accuracy. The framework to be developed is an adaptation of the FL model architecture to the architecture of the SG. To achieve this purpose, it is necessary to develop all the components related to the FL, in order to ensure the correct adaptation of each part of the framework to the context. Additionally, this framework should introduce two FL models working independently, one for each forecasting system, under the coordinator of the same server.

The results of the developed models will be crucial for the optimization of energy resources in smart cities. Such data sharing empowers effective management of energy resources (such as battery energy storage and electric vehicles) and ensures (economically and technically) viable transition to future decarbonized energy systems.

1.3 Main Contributions

The main contribution of this work includes introducing a federated learning framework to enhance the prediction accuracy of buildings' net demand, predicting the demand and generation independently. The proposed solution lends itself well to TE systems where buildings, aggregators, and DSO are decision-making entities. It is assumed that DSO shares the net-demand metering data with aggregators. The clients will collaborate to fine-tune their models by executing the training program using their data while keeping their data private. The aggregator will then update the model weights to improve the prediction of all buildings in the community.

The implemented approach is based on a Horizontal Federated Learning (HFL) model, but it introduces as a novelty the use of a third party to provide information in common to all buildings (e.g., weather data), the use of two HFL systems working in parallel under some coordination, and adaptability to the context of the formula used to calculate the average of the weights. While the proposed approach is presented in the context of net-demand prediction, it is envisioned that it can be extended to transactive energy communities' coordination problems.

The preliminary results of this work were accepted for publication in the following conference paper, presented in Appendix C:

- Mendes, N., Moura, P., Mendes, J., Salles, R., & Mohammadi, J. (2022) Federated Learning Enabled Prediction of Energy Consumption in Transactive Energy Communities. In 12^{th} IEEE PES Innovative Smart Grid Technologies Europe 2022 (ISGT-Europe 2022), Novi Sad (Serbia), $10 - 12^{th}$ October 2022.

The final results of this work are now being adapted to be submitted soon to the IEEE Transactions on Smart Grid.

1.4 Structure

The remainder of the dissertation is organized as follows. The Background and Literature Review are presented in Chapter 2. The Methodology is described in Chapter 3, and the Data and Scenarios are presented in Chapter 4. The achieved results are presented in Chapter 5. Finally, the main conclusions are highlighted in Chapter 6.

Chapter 2

Background and Literature Review

This chapter intends to introduce the main background of the work. First, a review of the Smart Grid (SG)s is presented and the following section provides the principal aspects related to Machine Learning (ML), and its importance in the SG context. The third section explains the origin of the FL, the principal aspects, and the advantages of this implementation on the SG, resolving some disadvantages of the ML in such a context. The last section presents, some literature review relatively to the three main topics (SG, ML, and Federated Learning (FL)).

2.1 Smart Grid

SG is the evolution of the traditional energy grid, relying on the use of the new digital technologies, such as the latest information and communication technologies. Such evolution has the objectives to save energy, increase the integration of renewable generation, reduce costs, and improving the reliability of the grid [15].

A SG conceptual model was introduced by National Institute of Standards and Technology (NIST). This model intends to focus on the main domains that are part of the SG system, as well as on their connections. As shown in Figure 2.1, it is the model adopted by European Union (EU), which is based on what was proposed by the agency NIST. This adapation was introduced in a report made in 2012 by CEN-CENELEC-ETSI Smart Grid Coordination Group [16]. The main difference between these models is the addition of a new domain, the Distributed Energy Resources, on the European Union (EU) version. Unlike the domain of Bulk Generation, Distributed Energy Resources includes small-scale generation (renewable generation) or storage, closer to the edge of the PS. The approach taken by the EU leads to the main goal of finding the balance between generation and demand as locally as possible (at building or neighborhood levels).

PS are in a restructuration with the objective of reducing the Greenhouse Gases Emissions associated with the generation, by integrating renewable generation, such as solar or wind power. However, such power plants are based on intermittent resources and are not dispatchable. Therefore, to keep the PS balance new options of flexibility are needed, such as energy storage or the control of demand.

Due to the restructuration of the Bulk Generation domain and the integration of Distributed Energy Resources, most parts of these types of power plants are designed as non-dispatchable, i.e., central with lower control capacity, one consequence of this is the need to control more the consumption instead of the generation to balance the system, so new strategies are needed to achieve this purpose. Alternatively, is the increment of the

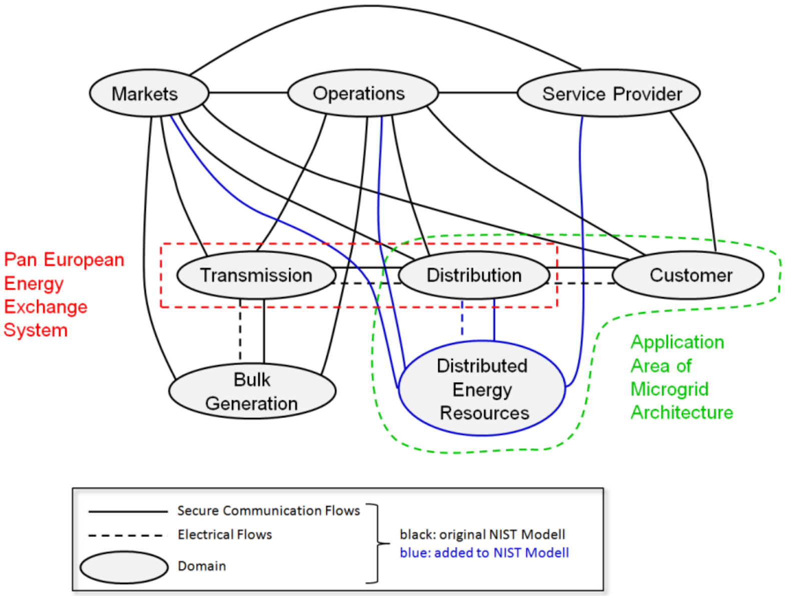


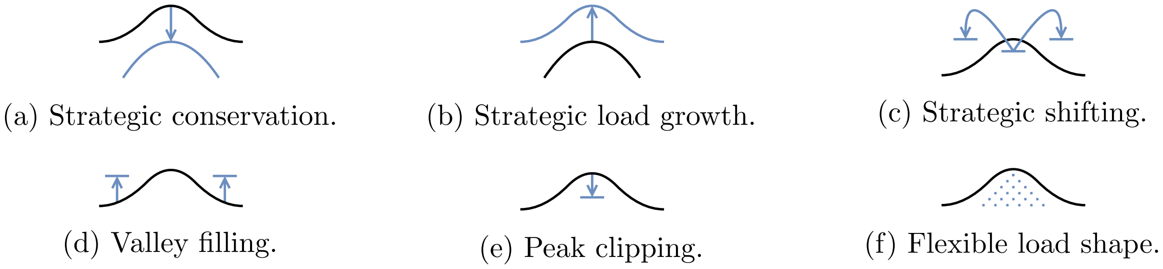
Figure 2.1: EU version of the SG conceptual model proposed by NIST agency [16].

“flexibility” to the power system and this can be done with the integrated energy storage, locally on the power plant or in a distributed way using, for instance, strategies related to electrical vehicles.

Demand-Side Management (DSM) are a set of measures between the service provider, market, and operations domains with the final customers, and can be defined as: “the planning, implementation, and monitoring of those utility activities designed to influence customer use of electricity in ways that will produce desired changes in the utility’s load shape, i.e., changes in the time pattern and magnitude of a utility’s load.” [17].

DSM can be split into two main areas [6]: energy efficiency and DR. Energy efficiency can be distinguished from DR through the time span of the impact. Energy efficiency can ensure medium and long-term impacts on the load diagram, having as goals not only the reliability and cost-effective operation of the grid, but also energy and cost savings, with the associated environmental and economic impacts.

Figure 2.2 presents the main objectives of the DSM related to the change of the load shape. The strategies presented from Figs. 2.2a to 2.2e ensure a change of the load diagram in the medium and long-term. If the objective is for short-term or real-time flexibility, another approach is needed, being implemented DR strategies, represented by Figure 2.2f. DR is defined as a strategy for end-use customers to reduce their use of electricity in response to power grid needs, economic signals from a competitive wholesale market, or special retail rates.



(a) Strategic conservation.

(b) Strategic load growth.

(c) Strategic shifting.

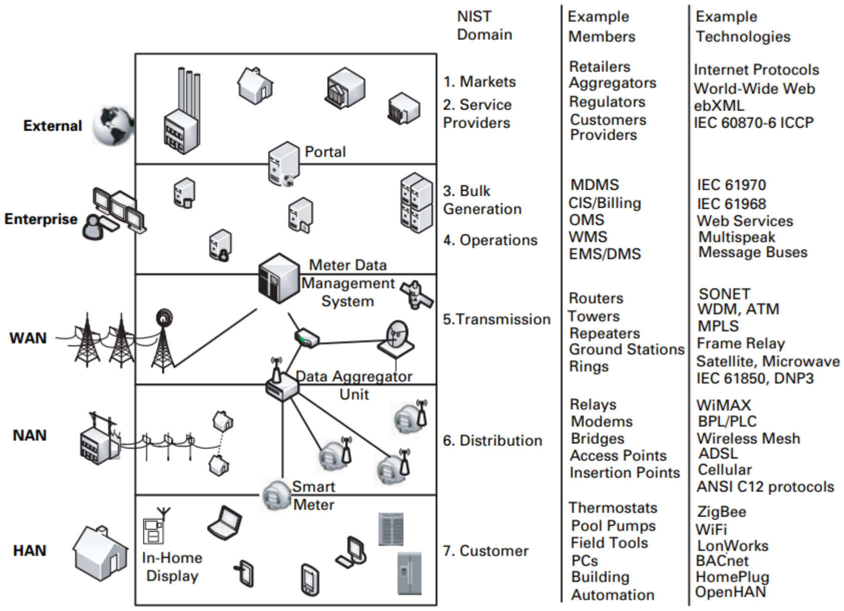
(d) Valley filling.

(e) Peak clipping.

(f) Flexible load shape.

Figure 2.2: Demand-Side Management objectives [17].

To perform these new services and improve the electrical grid, one major requirement is a reliable and secure telecommunications infrastructure, which needs to be well adapted to the electrical grid. NIST designed a conceptual model to the telecommunication infrastructure [18]. Such model is presented in Figure 2.3, and is noteworthy the connection made between a smart meter until the Meter Data-Management System (MDMS). MDMS has the function of supporting the applications of the power systems by providing storage, management, and processing of meter data received from the Data Aggregator Unit (DAU) [6]. These DAUs are collectors of data that receive, from multiple smart meters, the monitored data of electricity demand and/or generation (in the case of prosumers) in buildings. In [19], the information that is assessed by the MDMS is categorized into three categories: usage data (net-metering power energy), profile data (personal information about each building), and events (incidents that affect the metering). The same work presents the features of the MDMS, with two strongly aligned with the objectives of this dissertation: “Delivery of meter usage & weather data to the short term forecasting system”, and “The security for the customers to access to proprietary meter usage data”.



NIST Domain	Example Members	Example Technologies
1. Markets 2. Service Providers	Retailers Aggregators Regulators Customers Providers	Internet Protocols World-Wide Web ebXML IEC 60870-6 ICCP
3. Bulk Generation 4. Operations	MDMS CIS/Billing OMS WMS EMS/DMS	IEC 61970 IEC 61968 Web Services Multispeak Message Buses
5.Transmission	Routers Towers Repeaters Ground Stations Rings	SONET WDM, ATM MPLS Frame Relay Satellite, Microwave IEC 61850, DNP3
6. Distribution	Relays Modems Bridges Access Points Insertion Points	WiMAX BPL/PLC Wireless Mesh ADSL Cellular ANSI C12 protocols
7. Customer	Thermostats Pool Pumps Field Tools PCs Building Automation	ZigBee WiFi LonWorks BACnet HomePlug OpenHAN

Figure 2.3: Hierarchical overview of Smart Grid communication infrastructure and NIST Domains [6].

Concepts like Transactive Energy (TE) are being introduced in this renovation of PS. GridWise Architecture Council launched a report in July 2019 with a framework for TE systems [20] where TE is defined as: “A system of economic and control mechanisms that allows the dynamic balance of supply and demand across the entire electrical infrastructure using value as a key operational parameter”. This new paradigm is very important, since it will give more flexibility to the PS in order to integrate more smart control options to ensure the optimization of local energy communities. Additionally, such options can ensure a reduction of costs for the customers, through the use of price information to best manage their assets, namely by ensuring a reliable integration of renewable resources. With the use of modern technologies that are able to exchange and make use of information, it is possible to achieve higher efficiency in the management of the community of buildings. New options of advanced automation and optimization allow more effective use of intelligent management systems on the edge of the electrical grid.

TE will require new and wider data exchange, unlocking opportunities for new services, which will be crucial for the future operation of the PS. Control systems need to know the expected demand and availability of energy assets, in order to find the best control options. In the context of the PS this is made by predicting some crucial parameters like the electricity generation or demand in each building of the community. Therefore, prediction models will be the base of this type of system and to improve the implemented management strategies, it is crucial to improve such prediction models.

2.2 Machine Learning

Tomas M. Mitchell in [21] introduce the definition of learning from a computer program perspective: "A computer program is said to learn from experience $\mathcal{E}$ with respect to some class of tasks $\mathcal{T}$ and performance measure $\mathcal{P}$, if its performance at tasks in $\mathcal{T}$, is measured by $\mathcal{P}$, improves with experience $\mathcal{E}$.". Therefore, ML can be defined as a computer algorithm that learns with experience retained from data, being the result of this process a model out of the data [22].

The two most common problem types of machine learning tasks are classification and regression [22]. These are usually supervised learning processes, i.e., the model is trained knowing the correct outputs and the objective is to minimize the difference between the real results and the results provided by the model. The classification models are used to determine a specific class, in contrast, the regression models are made to estimate a value. An example of a classification task is the identification of an object that is represented in a particular image. Relatively to a regression, an example can be the prediction of future insurance prices [23].

An ANN is a method to provide a robust approach to approximating real-valued target functions and, according to [24], ANN is the principal ML method used for the prediction of energy in buildings. ANN can be defined as a set of interconnected model neurons or units imitating the logic of a brain [25]. Like a human brain, an ANN is constituted by multiple neurons, which are separated into three principal layers: input, hidden, and output. The training is made by adjusting the connection between neurons, layer by layer. This adjustment is made by changing the values of the weights. Figure 2.4 presents a representation of an artificial neural network architecture, where the circles represent the neurons.

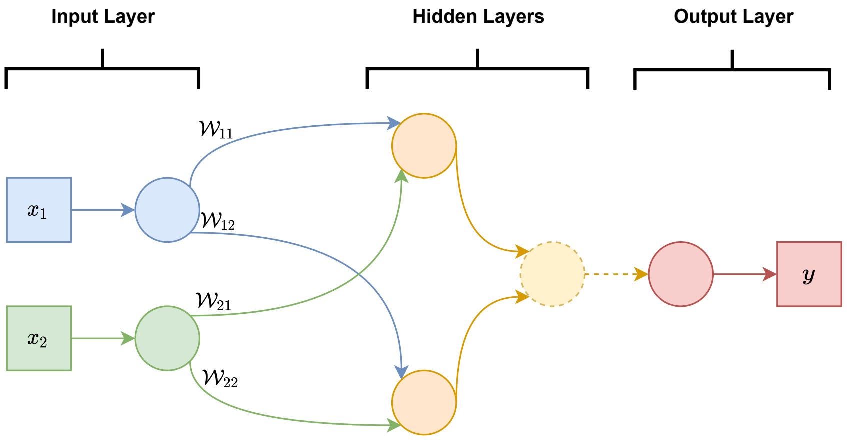
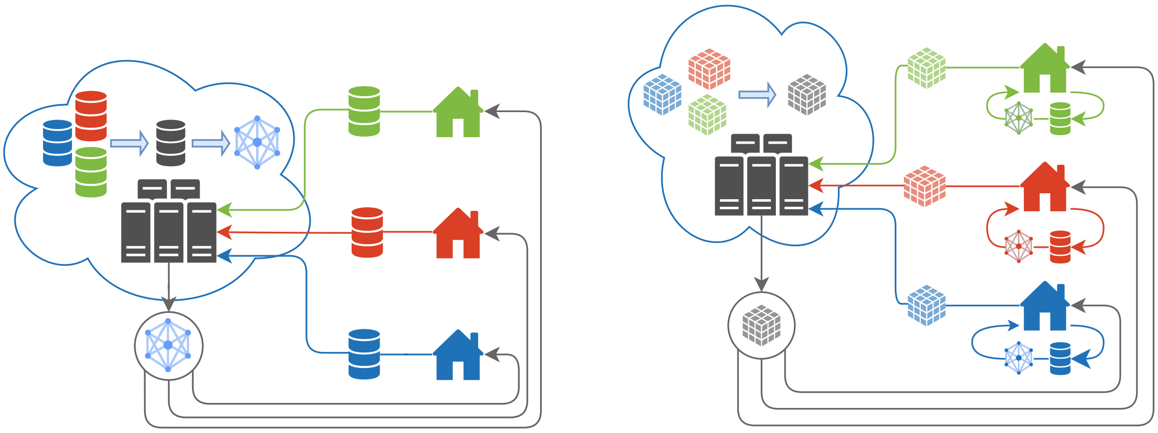


Figure 2.4: Architecture of an artificial neural network. Adapted from [25].

2.3 Federated Learning

FL was introduced by Google [26, 27, 28], and is a concept that praises the protection of private data by preventing data leakages. It can be defined as a distributed collaborative ML approach that allows for data training by coordinating multiple devices with a central server without sharing actual datasets. Therefore, the main objective is to train a model at each location where a data source is available, and then let the sites communicate their respective models weights to reach a consensus for a global model. Figures. 2.5 shows a comparison between a train centralized with a federated learning approach in edge devices on buildings. In Figure 2.5a the data are shared with the main server, and this one train a model that will be sent back to the devices. Figure 2.5b edge devices do not share data and train their local models, in the end, the weights are shared. With these ones, the server computes the global weights, which will be sent back to the devices to update their local models.



(a) Centralize machine learning train.

(b) Federated learning train.

Figure 2.5: Comparison between a train centralised with a federated learning approach in edge devices.

One of the first applications of this new approach was on Gboard systems, which is Google's keyboard, for auto-completion of words. This application is very important, taking into consideration that this type of model trains with private data collected from personal devices (like smartphones or tablets). With these strategies, it is possible to avoid sharing the data by training the models locally and not dismissing the collaboration between the devices for achieving better models for predictions.

The structure of FL is explained by [29], exposing the different options in terms of approaches, schemes, and categories. A FL system can be designed using two different approaches, a Business-to-Business or Business-to-Consumer (B2C). The Gboard system is an example of B2C where the business was represented by Google and had the objective to provide a better service for their users. The consumers can keep the privacy of their data, as well as increase the performance, since the communication will be faster due to the lower need for information to be broadcasted to the server. Behind these advantages, this structure does not lose the capacity for collaboration between different final users, allowing the learning process between the users to improve the prediction models and consequently increase their performance.

The model architecture of FL can be defined with the integration of a central coordinator or without it, i.e., in a client-server model or a peer-to-peer model. The peer-to-peer system is more secure, but on the other side needs more computation capacity, due to the high levels of encryption. In such a case, the client-server has the possibility of a higher level of control, but needs a connected third party.

FL can be split into three different categories: Vertical Federated Learning (VFL), Horizontal Federated Learning (HFL), and Federated Transfer Learning. The different categories are related to the constitution of the datasets to train, more specifically with the features available in each client and how they are related to each other. If all clients have overlapping data features on a sample, it is categorized as HFL. If the sample is composed of features that are divided between the clients it is considered Vertical Federated Learning (VFL). Federated Transfer Learning is when it is not possible to apply none of the above categories, i.e., the features in the clients do not match and together do not form a sample.

Figure 2.6 portrays the example of the Gboard system where it applies a HFL approach, so the training will be considering the same features in the clients. These cases can be considered as B2C since the targets are different users and they are an active part of a system that a business implements, with the objective of guaranteeing the privacy of their own data and, collaboratively, helping on the improvement of the system.

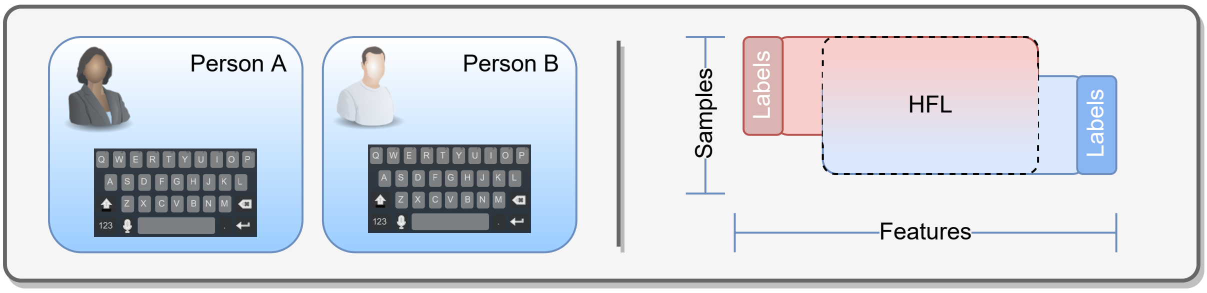


Figure 2.6: Example of Horizontal Federated Learning.

Figure 2.7 presents an example of a possible situation where a VFL approach is used. In such an example an insurance company wants to build a ML model to help in then the evaluation of a client, Person A. To improve the model, a hospital will be part of the system, but an exchange of data between the entities cannot be implemented. In this situation are two entities having mostly different features, but with a common objective (the target). So the aggregation of these features will allow form samples to train a machine learning model. Therefore, a VFL approach is ideal in this scenario. This type of approach is also a Business-to-Business design.

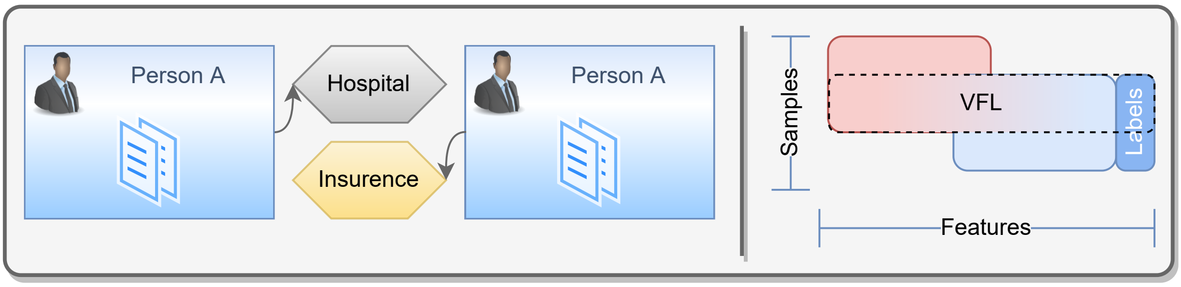


Figure 2.7: Example of Vertical Federated Learning.

Wearable healthcare devices, like for example smart watches, are increasing as an essential accessory of daily life. Therefore, a high quantity of data is being collected and predictive models can be produced for example to prevent some critical events like heart attacks. However, this is private data, and people use multiple different devices generating multiple different features. Figure 2.8 presents the example where a Federated Transfer Learning approach can be used, that is in scenarios with a lower overlap in terms of data samples and in terms of features. In this approach, the model will learn in each device and transfer their learning to others, where other features are applied to the same model. The common features between the devices can be used to tune in the model.

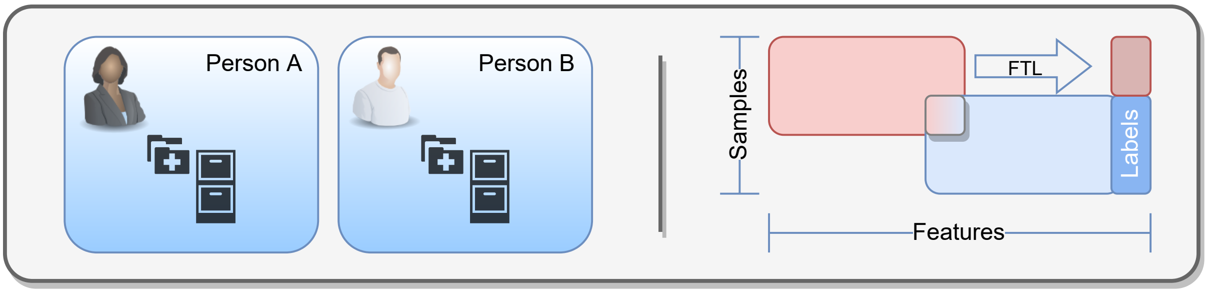


Figure 2.8: Example of Federated Transfer Learning.

2.4 Related Works

Energy prediction models have a critical role in energy policy and energy management in buildings. For demand prediction, the prevailing techniques used in large-scale building applications, include white-box, black-box, and grey-box based methods [30]. White-box approaches are physical techniques that can describe the thermal behavior of a building by mapping the inputs-outputs by first-principle equations. With the study of heat transfer, it is possible to implement better solutions with the objective of increasing energy efficiency. Recently, the booming development of deep learning techniques brings more exactitude to the black-box approaches, bringing promising alternatives to conventional data-driven approaches [31]. Black-box approaches the mapping of the inputs-outputs is derived from the data (data-driven modeling), not requiring transfer equations, or thermal and geometrical parameters. In agreement with what was exposed in Section 2.2, the reference [32] highlighted that statistical approaches, in concrete ANNs, are the most used ones for this type of forecast. Grey-box is a hybrid approach that embraces withe-box and black-box models, in a conventional way. This methodology makes use of the physical interpretation and uses statistical methods to determine the right parameters to use [33]. Figure 2.9 presents a diagram that exposes the difference between the white and black models type, as well as represents the grey-box model approach.

2.4.1 White-Box Models

J.A. White et. al in 1996 [34] introduces a new method to predict the monthly building energy demand using average monthly temperatures. Until the date of this publication, the methods used for load calculation are, for example, hourly simulation models like the DOE2 or ACESS, or for calculating cooling energy requirements the Equivalent Full Load Hours method is referenced [34]. In comparison with these methodologies, the proposed

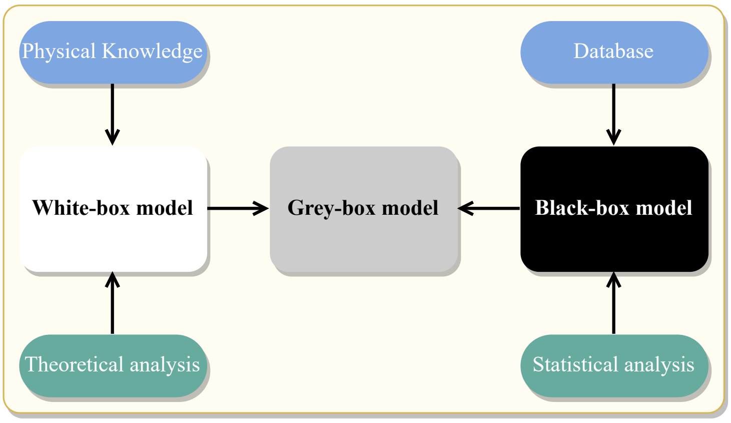


Figure 2.9: Representation of the white, black and grey models approaches. Adapted from [33].

method reveals is higher accuracy and can be applied to multiple types of buildings. In [35] a method for the prediction of daily demand profile for domestic buildings is presented. Such an approach used cluster analysis of the factors that made vary the load profile: daily energy-consumption for each appliance, ownership of each appliance, and occupied period. For the heating load profile is used: climate features, thermal characteristics, orientation, etc. The authors highlighted the fact that this method will have the capacity to adapt to different seasonalities, but need an adaptation considering the average daily-consumption of that seasons. Therefore, the method was only implemented for the winter months and will not work so well in summer conditions.

In [36], the thermo-aeraulic behavior of buildings with high energy efficiency is studied. The simulation platform SimSpark was used with the aim of selecting the best conditions in terms of energy efficiency. The conclusion was that counter-flow ventilation is indispensable to obtaining high energy efficiency. Ivan Korolija et al. in [37] present parameterized archetypal simulation models for the UK office building stock with the objective of predicting building energy consumption from heating and cooling demands.

In [38] a methodology that can be used in the design process to quickly assess the potential impact of daylighting in order to minimize the energy use associated with electrical lighting is presented. This method considers the physical characteristic of a building allowing to determine the percent savings in annual use of artificial lighting. This approach is done by using daylighting controls in office buildings.

The above five examples allow concluding that some of the disadvantages of this type of model can be the absence of a direct prediction of the power demand and having low adaptability to different contexts. These factors will increase the errors associated with the prediction. Another disadvantage is the need of knowing the characteristics of each building to study the thermal behavior, which is a strong barrier when different buildings need to be considered.

2.4.2 Grey-Box Models

Grey-Box models use the statistical model to optimize the parameters of the physical equations. For example, Che-Chiang Hsu and Chia-Yon Chen in [39] improve a grey-box model to predict the power demand by adding a ANN. This type of approach is not exclusive to predicting the power consumption, and can also be used to predict PV output power. An example is [40], where a grey-box neural network was introduced, with the prediction being done over weather data to improve the parameters of the physical model. The two presented examples show how this approach can predict demand and generation. However, the disadvantages are the complexity of this framework and the difficulty of collaboration between buildings.

2.4.3 Black-Box Models

Artificial Neural Networks

The use of ANNs in the context of PSs has been showing a strong increase [41]. Since the '90s this type of model is being used to forecast power load. Recently, the growth are being exponential, this is mainly due to the efforts made to improve ANNs techniques. This development brings new techniques, like deep learning, also with notable growth in the last years. Figure 2.10 presents the number of journal papers over the last three decades about load forecast, where it is possible to visualize the growth of the use of this technique in the energy power field.

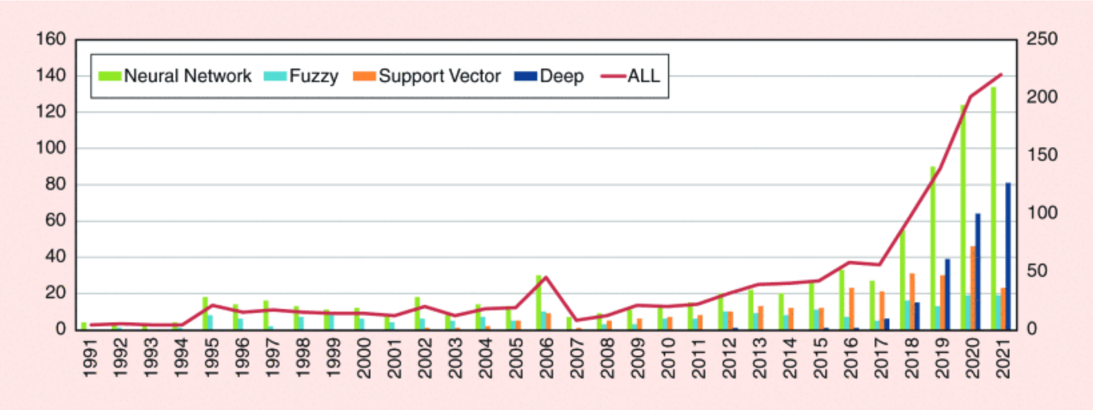


Figure 2.10: The number of journal papers with "load forecast" and "AI/ML techniques", in the title and in the abstract [41].

For example, a novel approach hybrid convolutional neural network with a long short-term memory autoencoder model was introduced by Zulfiqar Ahmad Khan et al. in [42] with the objective of predicting the electricity consumption in residential and commercial buildings. In [43], an ANN was used to predict the electricity consumption in Europe and Siberia. The results showed that ANN are adaptable to different regions, without changing the architecture itself just changing the reference data to train it.

For the prediction of power demand, it is also very important to set the considered time horizon. In [44] a novel framework was developed for the prediction of the total electricity consumption in commercial buildings with a day-ahead time horizon and with a 15 minute temporal resolution. For long-term forecasting, A.Azadeh et al. in [45] used an ANN approach to predict the annual electricity consumption in high energy consumption industrial sectors. Such a prediction has additional difficulties associated with the nonlinear variations of demand (strong variations), and due to the heterogeneity of industrial sectors. The paper shows that ANN fits well in such a situation, with a mean absolute percentage error of 0.0099.

The evolution of the PS brings more emphasis to distributed energy resources, and mainly to distributed generation based on renewable energy, and its integration needs new management options. Distributed generation is mainly ensured by PV generation in buildings. However, the generation profile usually does not match the demand profile. In order to improve the matching, it is critical to also predict the generation. To ensure such prediction, numerical weather data is widely used. Figure 2.11 represents the traditional approach to forecasting the power load, where the weather data has a crucial role.

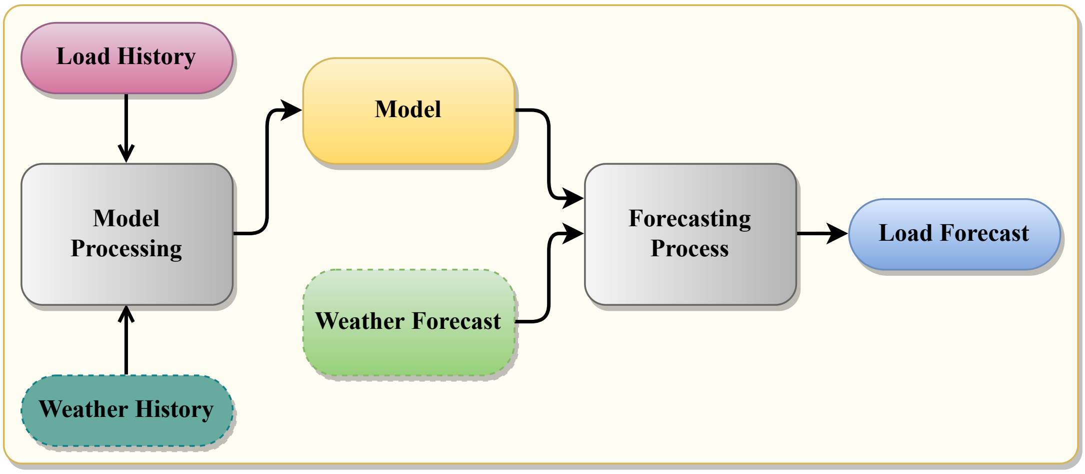


Figure 2.11: Common load forecast. Adapted from [41].

In [46] a neural network-based numerical weather prediction model was developed to be used in a framework with the objective to forecast solar power generation in a residential microgrid. Deep learning models are also being applied to this purpose, for example, [47] presents multiple models of this typology intending to forecasting the solar energy output.

In [48] a direct prediction of the net demand power for a residential building with a resolution of 15 minutes is done. Another example is [49] where a model was developed to predict the net load power, in this case, a forecast for a day ahead with a resolution of one hour was proposed. Such work also highlights the importance of adding weather features to increase the accuracy of the models.

S. Ehsan Razavi et al. in [50] studied the effects on the forecast when added PV generation to the system, at a residential level and at an aggregated level. Figure 2.12 presents the forecast error distribution. It can be concluded that the forecast at a residential level is less accurate, for a higher percentage of the time. It is also possible to conclude that increasing the PV generation increases the error in the hours with the highest generation level from the PV system.

Forecast at an aggregate level appears to be less challenging. This can be due to the biggest amount of data to train the prediction models. Another reason is the aggregation of multiple renewable power, which will generate an effect of cancellation over the associated intermittency to these types of power plants. The cancellation effect stabilizes the profile of the power generation, being more easier to achieve more accurate models.

A solution to approximate the forecast errors of the residential level to what is verified at an aggregate level can be forecasting independently the PV generation and the power consumption. This is based on the difference in terms of features used to ensure the prediction of power demand and power generation. Another solution can be taking advantage of the collaborative opportunities between buildings with similar characteristics, each one with private data resources that can be important to the other buildings. These

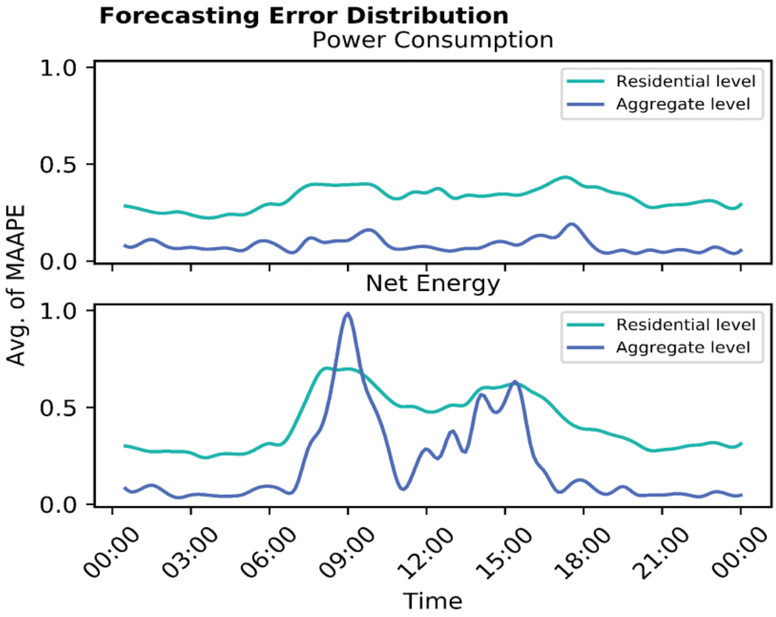


Figure 2.12: Forecast error on a residential level and an aggregated level. Comparing the forecast of power consumption with the forecast of net energy [50].

data silos have private information, like sub-metering data, occupancy data or private routines, but most buildings do not intend to publicly share such data making this approach unfeasible.

Federated Learning

The use of private data in collaborative machine learning problems on decentralized data where privacy is paramount has been ensured using FL [8]. Such approaches have been used primarily in mobile and edge device applications, but the use of FL in power systems is limited.

In the context of the SG, data silos issues are a concern mainly due to the privacy of the data. On another side, the aggregation of data from different buildings will increase the number of patterns, which will improve the learning process of the ML models. These models will be more prepared for different situations, getting more adaptability, being this the main reason for collaboration between buildings. These two situations, with an apparent contradiction, can be guaranteed with an FL approach. Some works have been using FL in power systems. For example, Junyang Li et el. in [51] used FL, in a client-server architecture, to predict the energy consumption, and with such a method it was possible to resolve the data silos problem. This type of structure does not avoid the typical issues of the ANN, like for example seasonality variations. In [52], FL was applied to predict electrical load, but using a short period of time to avoid the weather's fluctuations and seasonality. Christopher Briggs et al. present in [53] a comparison between FL, using a Long Short-Term Memory (LSTM) network, with two strategies: a non-private centralized training approach and a fully private localized learning approach for predicting residential energy demand. One of the conclusions was that FL models can outperform centralized learning, but perform worse than localized learning.

Recently, Xinxin Zhou et al. in [54] presented a FL model for household load forecasting. In such work, a non-intrusive load monitoring method based on the CNN-LSTM hybrid model is used. This local structure forecasts the consumption per appliance, and the sum of these values will generate the predicted power load.

FL has been apply on the edge of the power systems for different objectives, for example in [55] FL is used to identify the socio-demographic characteristics of electricity consumers, in [56] FL is used for the prediction of energy demand in electrical vehicles networks, in [57] a FL approach using clustered aggregation is implemented for the prediction of electricity demand, and [58] presents two case studies for predict power consumption, using HFL and VFL.

Such works presented that FL can be employed in the power system to resolve the data silos problem and the concerns relative to the privacy of the data, and can be used to achieve different goals. However, FL has not yet been used to enhance the prediction of net-demand in communities of buildings, predicting in an independent way the power generation and demand.

Chapter 3

Methodology

3.1 Framework

This dissertation considers a Transactive Energy (TE) environment that includes buildings, servers, Distribution System Operator (DSO) and third-party data providers, as presented in Figure 3.1. This structure is a representation of a local power system, respecting the fundamentals of the National Institute of Standards and Technology (NIST) model. In this setup, the DSO oversees the power flow between each building and the grid and has, therefore, access to all buildings' historical generation and demand metering data.

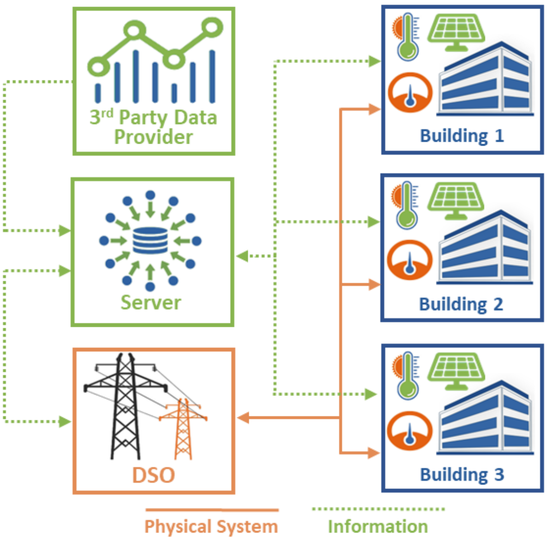


Figure 3.1: Power and data flow in the community.

Each building can also have additional information, such as separated generation and demand profiles, sub-metering, and occupancy data. However, in most situations, due to privacy concerns, the buildings do not intend to share their data with the server. Such information is relevant not only for the building owning the data, but also to other buildings since there is a high correlation between the photovoltaic (PV) generation in buildings in the same community and the demand correlation between buildings with the same user can also be high. This data can be used to increase the reliability of forecast systems, being such systems a critical part of a future TE environment. Other relevant data in common for all buildings, such as weather data, can be obtained from third-party data providers. This data can be very helpful, for example, to the forecast of generation, because of the relation the PV production has with the radiation values.

Figure 3.2 presents the Federated Learning (FL) implementation in this dissertation, respecting the NIST model and aligned with the previous structure (Figure 3.1). In this chosen approach the weights from the Artificial Neural Network (ANN), implemented in each building, will be sent to a server, and net-metering data sent to the DSO. The net-metering data in a TE community can be aggregated in a Data Aggregator Unit (DAU), to increase his privacy [59], for example, and then sent to the Meter Data-Management System (MDMS) where will be processed.

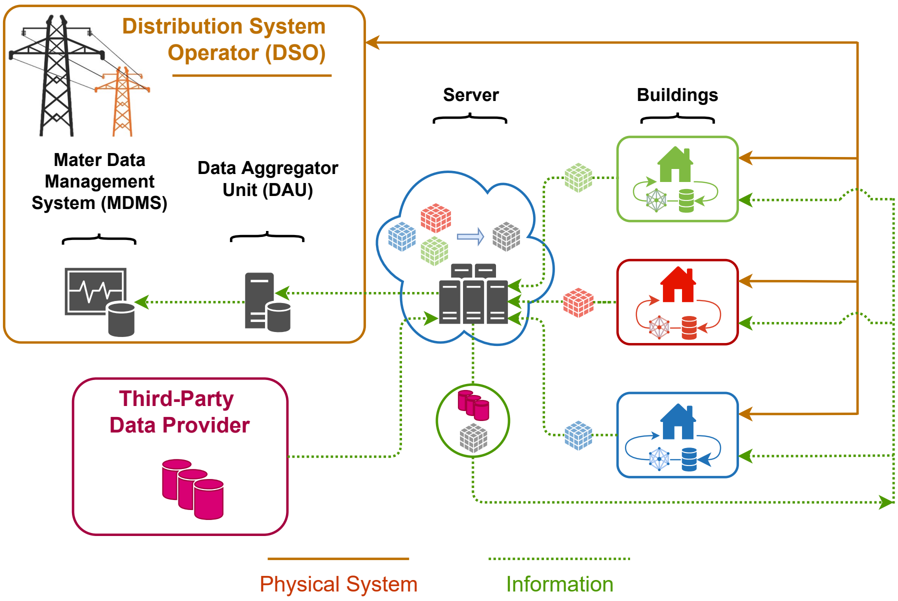

Distribution System Operator (DSO)
Mater Data Management System (MDMS)
Data Aggregator Unit (DAU)
Server
Buildings
Third-Party Data Provider
Physical System
Information

Figure 3.2: Framework implemented. This structure use a client-server architecture, with a Horizontal Federated Learning model.

In the same community in a given location, there is almost a perfect correlation between the PV generation variation in the several buildings (the only factors affecting it can be slightly different slops and orientation of panels or an eventual shading in some buildings). In the case of demand, even considering the same type of buildings, the correlation is not so high, since it is affected by factors with a different impact on the several buildings, such as temperature, schedule of services and number of users. Considering directly the net-demand would attenuate such correlations. For instance, on a sunny day with high temperatures, the PV generation and demand will be high, with the generation increase attenuating the demand increase, but with a differentiated impact in different buildings. Therefore, a separate prediction for generation and demand increases the reliability of the net-demand prediction.

This dissertation proposes a FL scheme for the presented setup. This strategy will enable the non-sharing of personal data by training locally the forecast models of power demand, and generation. This approach also provides collaboration inside of the TE community. In this setup, the buildings instead of sharing their personal information will only need to share the information relative to the forecast models.

In order to integrate the proposed system into the setup present in Figure 3.1, it is necessary to have a setup to ensure a global training with specific characteristics, namely: all the buildings will have the same features (for each forecast system), the structure of the model will be the same for all buildings, and is necessary to have a coordinator, the server. To ensure such characteristics, a FL approach was selected, with a client-server structure designed to a Business-to-Consumer (B2C) model. The FL approach will increase the protection of the private data of the buildings, while the use of a client-server structure has the objective to ensure a higher level of control, and have an independent entity connected to the DSO. The B2C model was chosen because in this case, the objective is the improvement of a model in multiple buildings with high overlap in terms of features, therefore fitting in the Horizontal Federated Learning (HFL) category.

The selected architecture fits on global structure presented in Figure 3.1, where the server will be responsible for the system coordination. The clients will be the buildings, where each building has its own system for training its personal forecasting(s) model(s). To protect the location information of the buildings these are identified by an ID, the set of IDs is represented by $\mathcal{B}$. Additionally, a third party was added to the system, which will be responsible for proving weather data. This approach intends to reduce the number of entities connected to the buildings, increasing their protection.

3.2 Server

The server is the coordinator of the two systems, one to forecast power demand and a second to forecast generation, being also responsible for interacting with the participants and controlling their learning process. The server is constituted by multiple algorithms that are used by both systems, but each one is used in an independent way, without any interaction between the systems.

Figure 3.3 represents the proposed structure for the server, with five variables as beginning inputs. These variables are the number of participants per round $\mathcal{K}$, the maximum number of rounds allowed per system: $\mathcal{R}_d$ (demand forecast) and $\mathcal{R}_g$ (generation forecast), a dataset provided from a third-party $\mathcal{D}_{cf}$ (represented as TP), and, the fifth variable, rst, is always pre-set as true and allows to decide how it is the beginning of the training. Figure 3.3, also includes other inputs, represented as 4 and 5, which are the weights returned by the participants at the end of a round and the minimum value of the loss $\mathcal{L}_{min}$, respectively.

The proposed framework adds a connection between the server and a third-party, represented in Figure 3.3 with TP. This third-party will provide a dataset with common features for all clients, $\mathcal{D}_{cf}$. The server will supervise the data and have the responsibility to share it with its clients. This third-party entity has data that is not classified as private and is of significant importance to achieving higher scores on the resultant forecast models. This approach also increases the security for the clients, since they have fewer connections to auxiliary parties.

Algorithm 1 presents the defined structure at the server level, where the systems are identified by g for the generation forecast and by d when comes to the power demand forecast. To simplify the pseudo-code, the independent cycles for power demand and generation forecasts are written in a unique cycle, where is use $\{r_g, r_d\}$ to differentiate the independent variables. This algorithm follows the architecture presented in Figure 3.3. After establishing the initial parameters, the algorithm advances for two loops $r_d = 0, 1, \ldots, \mathcal{R}_d - 1$ and $r_g = 0, 1, \ldots, \mathcal{R}_g - 1$ (line 5). For each round, $r_{\{g,d\}}$, a random set of buildings, $\mathcal{B}_r$, is selected by the Participant Selector Algorithm (PSA) (Algorithm 3) (line 6).

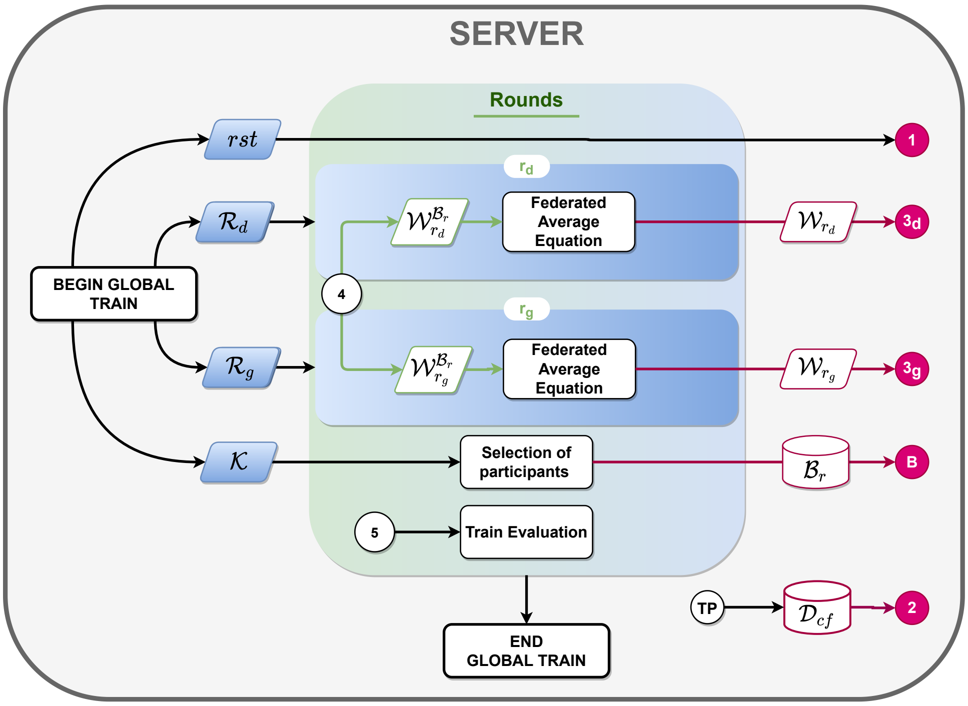


Figure 3.3: Proposed framework implementation for the server.

A inner loop iterate over the selected participants $\mathcal{B}_r$ ($b = b_0, b_1, \cdots, b_{\mathcal{K}-1}$) (line 7), activating the local trains with the CMA (Algorithm 4) (line 9 or 11). The CMA algorithm returns the last local weights $\mathcal{W}^b_{r_{\{d,g\}}}$, as well as the minimum loss $\mathcal{L}^b_{r_{\{d,g\}}}$. The returned local weights ($\mathcal{W}^b_{r_{\{d,g\}}}$, for $b = b_0, \cdots, b_{\mathcal{K}-1}$) are concatenated in $\mathcal{W}^{\mathcal{B}_r}_{r_{\{d,g\}}}$.

Variable rst allows requesting different things to the CMA algorithm, by defining different inputs. Therefore, according to the rst value, the following steps are implemented:

- $rst = True$: The first $\mathcal{K}$ participants will receive from the server the dataset of the common features $\mathcal{D}_{cf}$ shared by the third-party, and the beginning weights $\mathcal{W}_{begin_{\{d,g\}}}$ when rst is defined as $True$, and is the first round ($r = 0$) (line 9). If the condition is not verified, the CMA will receive the weights updated by the Federated Average Equation (FAE) (line 11), and the dataset of common features $\mathcal{D}_{cf}$.

- $rst = False$: Before starting the FL train, a function (λ) is activated, receiving as input a previous test number $\mathcal{T}_n$, and the number of the round from which it is intended to continue $\mathcal{R}_n$. This function will collect the weights from the train and defined round ($\mathcal{W}^{\mathcal{B}_r}_{r_{\{d,g\}}}$), from the collected weights, being used the FAE to compute the updated weights ($\mathcal{W}_{u_{\{d,g\}}}$).

When $rst = False$, the objective is to continue a previous training, enabling the use of the weights from a previous round in a specific global train. This approach can be useful in a scenario like, for example, the addition of a new building to the community or in debugging situations. Therefore, this continuous train will receive more developed models for the two systems of forecasting (d and g).

At the end of the loop over the participants, the updated weights ($\mathcal{W}_{u_{\{d,g\}}}$) are calculated, by the FAE which will receive the concatenated weights for all participants ($\mathcal{W}^{\mathcal{B}_r}_{r_{\{d,g\}}}$).

Algorithm 1 Server Main Algorithm

Variables:

- **Inputs:** $rst,\ \mathcal{R}_{\{d,g\}}=\{\mathcal{R}_d,\mathcal{R}_g\},\ \mathcal{K},\ \mathcal{D}_{cf},\ \mathcal{T}_n,\ \mathcal{R}_n.$
- **Train Inputs:** $\mathcal{W}^b_{r_{\{d,g\}}},\ \mathcal{L}^b_{r_{\{d,g\}}}.$
- **Outputs:** $\mathcal{W}_{u_{\{d,g\}}}=\{\mathcal{W}_{u_d},\mathcal{W}_{u_g}\},\ \mathcal{D}_{cf}.$
- **Local:** $\mathcal{W}^{\mathcal{B}_r}_{r_{\{d,g\}}},\ \mathcal{W}_{begin_{\{d,g\}}},\ \mathcal{B}_r,\ \mathcal{B}.$

Pseudo-code

```
if $rst$ = False then:
    $\mathcal{W}^{\mathcal{B}_r}_{r_{\{d,g\}}} \leftarrow \lambda(\mathcal{T}_n, \mathcal{R}_n)$
    $\mathcal{W}_{u_{\{d,g\}}} \leftarrow \mathrm{FAE}(\mathcal{W}^{\mathcal{B}_r}_{r_{\{d,g\}}})$   (Equation (3.2))
end if
for $r_{\{d,g\}} = 0, 1, \cdots, \mathcal{R}_{\{d,g\}} - 1$ do
    $\mathcal{B}_r \leftarrow \mathrm{PSA}(\mathcal{B}, \mathcal{K}, \mathcal{D}_{control})$   (Algorithm 3)
    for $b = b_0, b_1, \cdots, b_{\mathcal{K}}$ do
        if $rst = True \wedge r = 0$ then
            $\mathcal{W}^b_{r_{\{d,g\}}},\ \mathcal{L}_{min_{r_{\{d,g\}}}} \leftarrow \mathrm{CMA}(\mathcal{D}_{cf}, \mathcal{W}_{begin_{\{d,g\}}})$   (Algorithm 4)
        else
            $\mathcal{W}^b_{r_{\{d,g\}}},\ \mathcal{L}_{min_{r_{\{d,g\}}}} \leftarrow \mathrm{CMA}(\mathcal{D}_{cf}, \mathcal{W}_{u\{d,g\}})$   (Algorithm 4)
        end if
        $\mathcal{W}^{\mathcal{B}_r}{}_{r_{\{d,g\}}} \frown \mathcal{W}^b_{r_{\{d,g\}}}$
    end for
    $\mathcal{W}_{u_{\{d,g\}}} \leftarrow \mathrm{FAE}(\mathcal{W}^{\mathcal{B}_r}_{r_{\{d,g\}}})$   (Equation (3.2))
end for
```

3.2.1 Control and Evaluation of the FL Training

The server has control over the local training, having a dataset where the following parameters are registered, for each client: building ID, system (power demand or generation), alpha value (α), round when α achieve the value of zero, minimal loss, round of the minimal loss, a datapoint for a occurred local error, and the number of the total rounds.

To control the learning process an algorithm based on an early stopping strategy [23] was developed. The idea behind is "Instead of running the optimization algorithm until a (local) minimum of validation error is reached, it runs until the error on the validation set has not improved for some amount of time.". Algorithm 2 adapt this strategy to the developed FL framework, for each participant in each round. At the end of each local training, the respective participants analyze their own training and the minimal loss achieved ($\mathcal{L}_{min}$) is shared with the server. The $\mathcal{L}_{min}$ value received is compared with the previously recorded one ($\mathcal{L}_{min_{actual}}$), being recorded when the value is lower. When the minimal loss shared by the client is higher than the recorded, the values are not saved and the α values decreases one unit.

Algorithm 2 Early Stopping Algorithm for Federated Learning.

Variables:

- **Input:** $\mathcal{D}_{control}$, $\mathcal{L}_{min}$, r_n.
- **Local:** $\mathcal{L}_{min_{actual}}$, α.

Pseudo-code:

```
$\mathcal{L}_{min_{actual}} \leftarrow \mathcal{D}_{control}$
if $\mathcal{L}_{min} < \mathcal{L}_{min_{actual}}$ then
    update $D_{control} \leftarrow \mathcal{L}_{min}$, $r_n$
else
    $\alpha \leftarrow \mathcal{D}_{control}$
    $\alpha = \alpha$ - 1
    update $D_{control} \leftarrow \alpha$
end if
```

Observation:

- In Algorithm 3, line 2, a function ψ verifies the control dataset. From this verification process, if $\alpha = 0$ the client will not be selected. This complements the early stopping algorithm proposed, not allowing the client training anymore.

This global analysis checks also if the train occurs without any problems by checking the existence of not-a-number values in the loss set. This information is shared with the server, and the server will record the same information on the control dataset ($\mathcal{D}_{control}$), on a variable that records if an error occurred. If the server receives the information of a registered error, the number of total rounds of that client is not updated, the respective variable is updated to one (in the beginning, all clients have this parameter as zero), and the Algorithm 2 are not been activate for the correspondent participant. When any error occurs, the number of total rounds is updated, and the Algorithm 2 is triggered.

Locally, the objective is to achieve the minimal value for the validation loss. Firstly, during the training, the early stopping strategy over the values of the validation loss is implemented. When the training is concluded the algorithm advances to verify if any error occurred. By checking the values of the validation loss the existence of non-a-number values is verified, being its existence an indication of an occurred error in the training process. With the global and local analysis, the validation loss and the loss values are assessed. These strategies allow a complete analysis over the ANN train in each client.

3.2.2 Participant Selector Algorithm

The Participant Selector Algorithm (PSA), introduced in Algorithm 3, has the objective of selecting $\mathcal{K}$ clients per round, maintaining the number of participating rounds for each building balanced with a randomness characteristic associated. The PSA gets as input the building IDs ($\mathcal{B}$), the control dataset of the server ($\mathcal{D}_{control}$), and the number of participants per round ($\mathcal{K}$). Being the final set, $\mathcal{B}_r$, composed by different $\mathcal{K}$ building IDs, and mandatory with $\mathcal{K}$ buildings selected.

To proceed to this selection the control dataset ($\mathcal{D}_{control}$) is used, first by selecting the intended system (demand or generation), and after by selecting the participants that have no register of an occurred error and with α value higher than zero (line 2). From the resultant set of buildings, for each system, the number of clients selected is checked and if it is lower than $\mathcal{K}$, the server does not select any client to train and, consequently, the server will stop the FL train for that system (line 4). Between lines 5 and 7, the following is done: When the selected clients are equal to or higher than $\mathcal{K}$, the lowest number of total rounds done by a client is recorded, being selected all clients with the exact same total rounds. With this resultant set of buildings, the number of buildings is again verified, to check if this set is higher than $\mathcal{K}$, being the final set of buildings selected through a random function. When the selected clients are lower than $\mathcal{K}$, such value is incremented to the value of minimal rounds, and the process is repeated until achieving $\mathcal{K}$ clients.

3.2.3 Federated Average Equation

The Federated Average Equation (FAE) was based on the "FedAVG Algorithm" proposed by H. Brendan McMahan et al. in [28]. On the "FedAVG Algorithm", the coordinator takes the weighted average of the received models' weights, by using equation (3.1).

$$\text{Weighted average} = \mathcal{W}_u \leftarrow \sum_{b=0}^{\mathcal{B}_r} \frac{n_b}{n} \mathcal{W}_r^b \ . \tag{3.1}$$

This equation assumes a global dataset $\mathcal{D}$, where the data is partitioned by clients in each round. $\mathcal{D}_b$ is the set of indexes of data points on client b, being $n = |\mathcal{D}|$ and $n_b = |\mathcal{D}_b|$. Accordingly to equation 3.1, it is possible to infer that this approach values a client with a larger dataset.

In a Transactive Energy (TE) community, different buildings will have different profiles of consumption. To enable the forecast models to achieve more accurate predictions in distinct profiles, all participants need to contribute with their "learning" at the same rate. Therefore, in this context, it is important to enhance the models to different target

Algorithm 3 Participant Selector Algorithm (PSA).

Variables:

- **Inputs:** $\mathcal{B}$, $\mathcal{K}$, $\mathcal{D}_{control}$.
- **Outputs:** $\mathcal{B}_r$.

Pseudo-code:

```
for $s \in d, g$ do
    $\mathcal{B}_s \leftarrow \psi\ (\mathcal{D}_{control},\ s)$
    if $n(\mathcal{B}_s) < \mathcal{K}$ then
        return $\mathcal{B}_s = \{\}$
    else
        while $n(\mathcal{B}_s) \neq \mathcal{K}$ do
            $\mathcal{B}_s \leftarrow \gamma(\mathcal{B})$
        end while
    end if
end for
return $\mathcal{B}_s$
```

profiles, and for this reason, a client with more samples should not have a higher impact when compared with the other ones. Therefore, equation (3.1) needs to be adapted, being proposed Equation (3.2). Equation (3.2) does not weight the average by the size of the dataset, using instead the number of clients per round without any error ($\mathcal{N}_p$). With this approach, all clients without any error in the training process take equal importance regardless of the number of local samples. The models' weights of the clients that register an error on the local train, will not be used in this equation, avoiding error propagation through the FL train.

$$\text{FAE}(\mathcal{W}^{\mathcal{B}_r}_{r_{\{d,g\}}}) = \mathcal{W}_u \leftarrow \sum_{b=0}^{\mathcal{B}_r} \frac{1}{\mathcal{N}_p} \mathcal{W}^b_r \ . \tag{3.2}$$

3.3 Client

Clients are all the buildings connected to the server, being the PSA algorithm used by the server to select $\mathcal{K}$ buildings that will be designated as participants. The architecture of the FL assembly in each participant is represented in Figure 3.4. The training process is presented in Algorithm 4.

Figure 3.4 has two independent but similar parts, one for the demand forecast system and a second for the generation forecast system. Circles numbered with 1, 2, 3_d, and 3_g are the input variables. Input variable 1 represents the rst variable. Variable 2 is the $\mathcal{D}_{cf}$ sent by the server, and, if $rst= False$, the variables 3_g and 3_d will provide the values of the weights for the forecast model of generation and power demand, respectively. The circle with the number 4 in Figure 3.4 represents the output variable $\mathcal{W}^b_{r_{\{d,g\}}}$. When the variable rst was defined as $False$, the algorithm inside of the block that says “Weights Recovery” is computed, and the resultant variables are sent back to the server by circle number as 4. The block named “’Loss Evaluation’ is where the values of the loss received from the train are checked. If not-a-number values are found, the variable $\mathcal{V}_{train}$ is set as $False$, and $\mathcal{L}_{min}$set as an empty variably. Otherwise, $\mathcal{V}_{train}$ is set as $True$, and $\mathcal{L}_{min}$ is defined. Circles numbered with 5 and 6, are the remaining output variables of $\mathcal{V}_{train}$ and $\mathcal{L}_{min}$, respectively.

Figure 3.4 also represents the final part of the FL train in each client when the global train has ended. At that point, each client has a forecast model for demand and another for the forecast of generation. The prediction of net-demand, $\mathcal{P}_{nd}$, will then be computed through the values of the predicted demand, $\mathcal{P}_d$, and the predicted generation, $\mathcal{P}_g$. Therefore, the forecast of the net-demand will be obtained by Equation (3.3).

$$\mathcal{P}_{nd} = \mathcal{P}_d - \mathcal{P}_g \ . \tag{3.3}$$

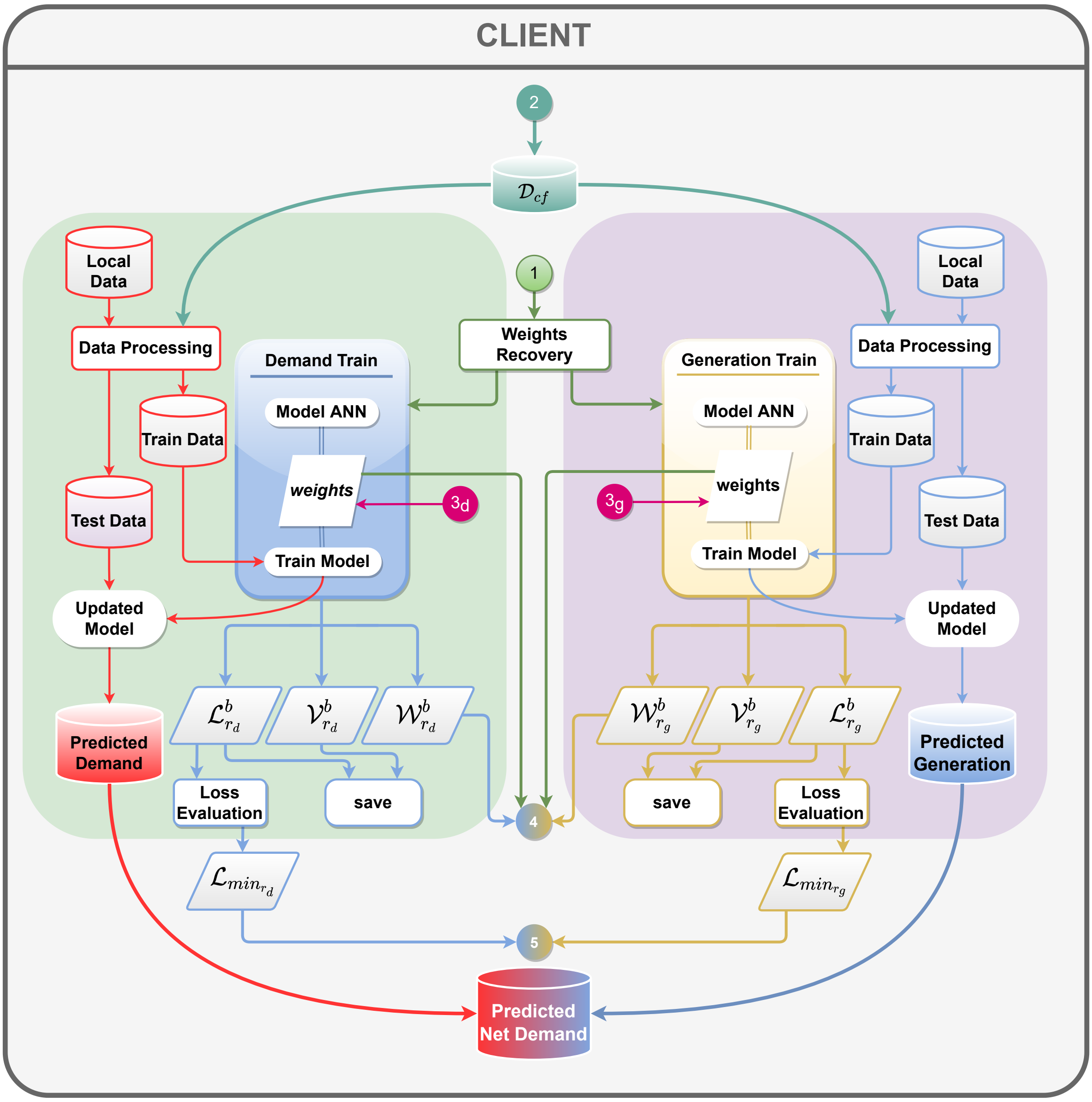


Figure 3.4: Proposed framework implementation for each client.

The Algorithm 4 presents the implemented code for a local train. In each participant, data processing is done: starting by concatenating the local data, $\mathcal{D}^{b}{}_{pvt_{\{d,g\}}}$, with the data provided from the server $\mathcal{D}_{cf}$ (line 1). The second stage is standardizing this resultant dataset (function ϕ, lines 2-8). To ensure it, the availability of the scaler is checked, $\Phi^{b}{}_{\{d,g\}}$. If this is true the scaler is read. Otherwise, it is necessary to compute it by calculating the mean and the standard deviation of the data [60]. Then, the scaler is saved for the next rounds. Standard deviation, in a generic way, for the set of N values $x_1, x_2, \cdots, x_N$, is calculated in three steps: first, the mean can be computed by Equation (3.4) and the standard deviation is the square root of the variance (Equation (3.5)), as given by Equation (3.6).

$$\text{Mean} = \overline{x} = \frac{\sum_i^N x_i}{N}, \tag{3.4}$$

$$\text{Variance} = s^2 = \frac{\sum_i^N (x_i - \overline{x})^2}{N-1}, \tag{3.5}$$

$$\text{Standard Deviation} = s = \sqrt{s^2}. \tag{3.6}$$

The standardized datasets, $\mathcal{D}_{res-SD_{\{d,g\}}}$, will be split into a training dataset and a test dataset (line 9). The training data will be used to train the local forecast model, and the testing data to verify it. The next step is to split the data into features $\mathcal{X}$ and targets $\mathcal{Y}$. In line 10, a function ζ is used to make the adjustments to the data in order to be used by the ANN. This procedure will depend on the number of observations that are defined to make the prediction $\mathcal{N}_{po}$ and the range of that prediction $\mathcal{N}_{fo}$. In line 11, the $\mathcal{W}$ is defined as being equal to $\mathcal{W}_{u_{\{d,g\}}}$ or $\mathcal{W}_{begin_{\{d,g\}}}$, depending on their existence. The train of the ANN is done in line 12, where will be returned the final weights $\mathcal{W}^{b}_{r_{\{d,g\}}}$, the loss $\mathcal{L}^{b}_{r_{\{d,g\}}}$, and the validation loss $\mathcal{V}^{b}_{r_{\{d,g\}}}$. During the training, the process of early stopping is implemented and an extra function that saves the best model according to the validation loss values is also implemented. To finish the CMA, the loss and the validation loss are saved (line 14), and line 16 returns the weights, the minimum value of the loss $\mathcal{L}_{min}$ (achieved in line 15), and a variable that validates or not the train. This variable is defined in line 13, through function ξ that verifies if not-a-number values exist in the loss registered during the train.

Algorithm 4 Client Main Algorithm (CMA)

Variables:

- **Input:** $\mathcal{D}_{cf}$, $\mathcal{W}_{u_{\{d,g\}}}$, $\mathcal{W}_{begin_{\{d,g\}}}$.
- **Output:** $\mathcal{W}^b_{r_{\{d,g\}}}$, $\mathcal{L}^b_{r_{\{d,g\}}}$.
- **Local:** ${\mathcal{D}^b}_{pvt_{\{d,g\}}}$, $\Phi^b_{r_{\{d,g\}}}$ $\mathcal{N}_{fo}$, $\mathcal{N}_{po}$.

Pseudo-code:

```
$\mathcal{D}_{res_{\{d,g\}}} \leftarrow \mathcal{D}_{cf} \frown {\mathcal{D}^b}_{pvt_{\{d,g\}}}$
if $\exists\ {\Phi^b}_{\{d,g\}}$ then
    read ${\Phi^b}_{\{d,g\}}$
else
    ${\Phi^b}_{\{d,g\}} \leftarrow \phi(\mathcal{D}_{res_{\{d,g\}}})$
    save ${\Phi^b}_{\{d,g\}}$
end if
$\mathcal{D}_{res-SD_{\{d,g\}}} \leftarrow {\Phi^b}_{\{d,g\}}(\mathcal{D}_p)$
$\{\mathcal{D}_{train_{\{d,g\}}}, \mathcal{D}_{test_{\{d,g\}}}\} \leftarrow \mathcal{D}_{res-SD_{\{d,g\}}}$
$\mathcal{X}_{\{d,g\}}\ , \mathcal{Y}_{\{d,g\}} \leftarrow \zeta(\mathcal{D}_{train_{\{d,g\}}}, \mathcal{N}_{po}, \mathcal{N}_{fo})$
$\mathcal{W} \leftarrow \text{available}(\mathcal{W}_{u_{\{d,g\}}}, \mathcal{W}_{begin_{\{d,g\}}})$
$\mathcal{W}^b_{r_{\{d,g\}}}, \mathcal{L}^b_{r_{\{d,g\}}}, \mathcal{V}^b_{r_{\{d,g\}}} \leftarrow \text{ANN}(\mathcal{X}_{\{d,g\}}\ , \mathcal{Y}_{\{d,g\}}, \mathcal{W}, \mathcal{N}_l, \mathcal{E})$
$\mathcal{V}_{train_{r_{\{d,g\}}}} \leftarrow \xi\ (\mathcal{L}^b_{r_{\{d,g\}}})$
save $\mathcal{L}^b_{r_{\{d,g\}}}, \mathcal{V}^b_{r_{\{d,g\}}}$
$\mathcal{L}^b_{r_{\{d,g\}}} \leftarrow \mathbf{min}(\mathcal{L}^b_{r_{\{d,g\}}})$
return $\mathcal{W}^b_{r_{\{d,g\}}}, \mathcal{L}_{min_{r_{\{d,g\}}}}$
```

Chapter 4

Data and Scenarios

This chapter presents the datasets and the scenarios used for the validation of the proposed framework presented in Chapter 3. The validation was divided into two phases. The first phase was used to validate the forecast power demand framework, and the second phase tested the whole framework with more complex tests. Two distinct scenarios were used, one for each phase, to test the proposed framework. First, Scenario A is referent to the Polo 2 campus at the University of Coimbra, and the main goal of this scenario was to ensure an initial test of the part of the framework relative to the power demand forecast and use the Equation (3.3) to calculate the net-demand. Scenario B uses a dataset with synthetic data from NREL [61], where the main goal is to test the completed developed framework. NREL provide a collection of datasets, with synthetic data that simulates the energy consumption and generation profiles in the United States residential and commercial buildings stock.

4.1 Scenario A

Scenario A was a preliminary test, testing only the system of demand forecast. Figure 4.1 presents the used diagram in the initial phase relatively to the client. The server was designated “Aggregator” because his main purpose was aggregate the weights and do the computation of the average of the weights (Equation (3.2)). At this point of development, the function of train evaluation was not implemented, and the selection of the participants had a more simple version, only randomly selecting without any associated conditions.

The used data belongs to the Polo 2 campus at the University of Coimbra. In particular, six buildings from the University of Coimbra were selected: the Department of Civil Engineering, Department of Chemical Engineering (DChE), Department of Electrical and Computer Engineering (DECE), Department of Earth Sciences, Department of Informatics Engineering, and Department of Mechanical Engineering.

In this scenario, the dataset of the buildings only had data of net-demand, with the exception of the DECE which also had generation data. To use this dataset to predict the net-demand, it was decided that all buildings will have a generation profile based on the data from DECE, corrected by the ratio between the installed PV power in the two buildings. This decision was made due to the physical proximity of the buildings, leading to a high correlation in the available solar radiation. Before the implementation of the test, according to Equation (3.3), the data was updated by adding the generation data to obtain the demand in each building. After obtaining the demand forecast, the generation

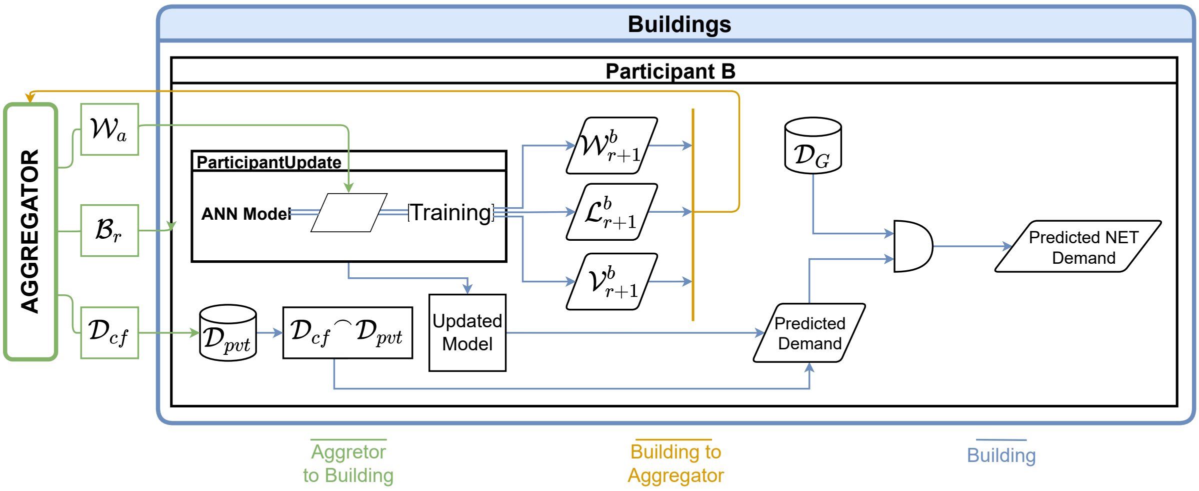


Figure 4.1: Initial phase of the client framework.

data was subtracted to obtain the forecasted net-demand to be compared with the initial data, the net-demand in each building.

The private dataset for all buildings needs to match in terms of features. The common feature set includes: (i) each building's net-demand, and (ii) on-site PV generation data (which was based on the generation of the DECE building). This information allows for calculating electricity demand. However, since the buildings belong to a University, there is a clear seasonality of activities and therefore other included features were the academic calendar (classes, exams and vacations) and the days of the week (weekends and weekdays).

The buildings do not directly collect any weather data, therefore requiring the participation of a third party to provide such data. Additionally, due to the physical proximity of the buildings, only one weather dataset is needed. The "Wunder Ground" website [62] served as a third-party temperature, humidity, and solar radiation data provider. This part was implemented in the framework using a technique called *Web scrapping*, which makes the download of the intended data directly from the website.

The net-demand and PV generation data have a periodicity of 15 min, and therefore one year corresponds to a total of 35040 samples per building. The academic calendar and days of the week are categorical data. In order to convert this data into numeric values, a process who is known as One Hot Encoding (OHE) [23] was implemented. This process consists of splitting the categories associated with variables and each sample. Put differently, correspondent sub-categories are identified with one while others are assigned zero. For example, in the simulations, three sub-categories were considered for the academic calendars: classes, exams, and vacations. Weekends and weekdays are separated as well.

The weather dataset presents a variable sampling frequency and, consequently, such data needed to be processed to match the private data on the buildings (net demand and PV generation). The weather data is a temporal time series and was processed day by day to not get a high loss in the temporal pattern. For each day, and for each feature, the existence of non-values was checked to eliminate the correspondent sample. Therefore, considering the frequency of the net demand and PV generation, one single day needs to have at least 96 samples and at least 4 samples per hour and if one of these conditions is not verified the entire day is eliminated. The result of this process was a loss of 10.41% of data, corresponding to the elimination of 38 days, with 3648 sample losses. It should be noted that such days also need to be eliminated from the private's datasets.

The proposed FL framework was validated by splitting the dataset of each building into two parts: 74% dedicated to training (used to train the local models) and the rest to testing (utilized to test the obtained models after the global training). The months of January, August, and October were selected for testing the models in different seasonal conditions. The remaining months were allocated to the training dataset. The model uses 25% of the training dataset for validation in the local training. Figure 4.2 illustrates the division realized over the dataset in each building.

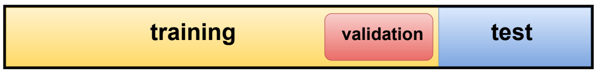


Figure 4.2: Dataset division: training dataset is represented in yellow, the validation dataset in red, and the test dataset in blue.

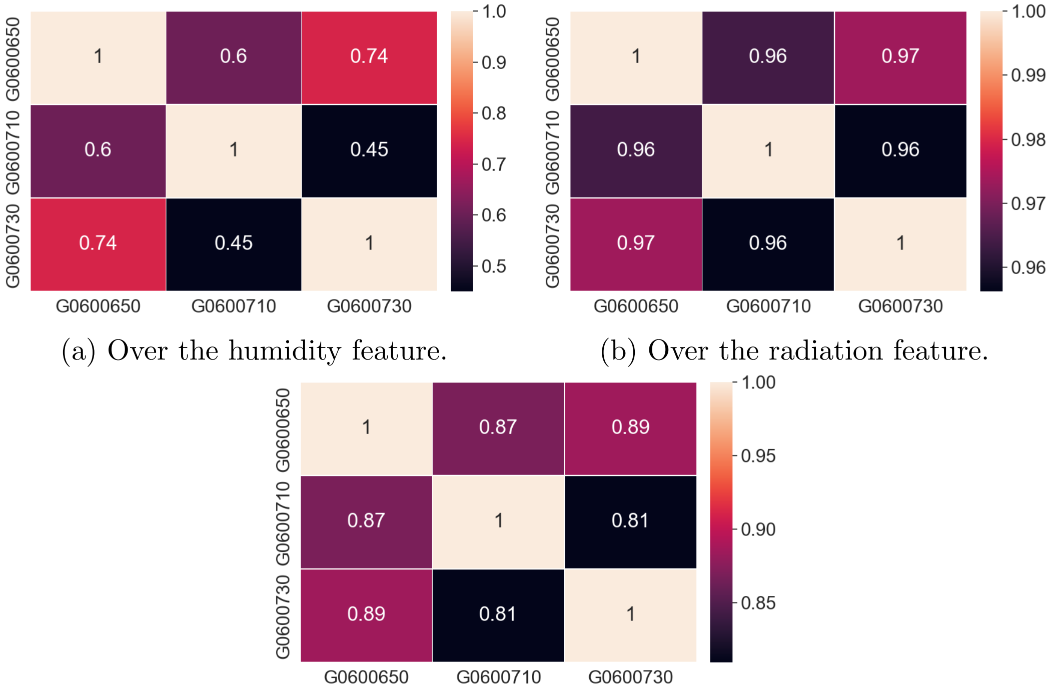
4.2 Scenario B

NREL provides a database directed for energy [61], composed of 2336 different territorial regions and 3107 distinct weather zones. Where a territorial region can have multiple climate zones, and a climate zone can belong to different territorial regions. To select from the available datasets, the buildings and regions that better fit the requirements to test the framework developed were selected based on criteria such as the number of buildings with PV or the climate region to which each building belongs. The commercial buildings were discarded, since the buildings used in the dataset do not have PV systems. Such buildings can only be used to test the demand forecast. Therefore, this type of building was excluded, being decided to use only residential buildings.

4.2.1 Analysis of Buildings

To implement the analysis of the dataset the building characteristics were provided on a file called *metadata*. In such a file, a feature that identifies if the building has or not a PV system is provided. Using such information a list of the fifty zones with the highest number of buildings with PV was done. The objective was to identify zones with a high share of buildings with PV, in order to create a community of buildings with a high penetration of PV. It was concluded that all regions with a high share of PV are part of the California state, and possess three climate different areas. These areas are identified with the following codes: G0600650, G0600710, and G0600730. The climate datasets are a very important characteristic in the pattern associated with the profile of PV generation, and also relevant for the power demand load profile.

To verify if these climate areas have similar patterns, allowing to be approached by the same forecast system, the degree of correlation between the features of the three different datasets was calculated using the Pearson correlation coefficient [63]. The analyzed features were the temperature, humidity, and solar radiation, since these are the ones to be used by the forecast system. The matrices of the Person correlation are presented in Fig. 4.3. It was possible to conclude that temperature and solar radiation have a very good correlation between the three different climate areas. Relatively to the humidity, the same is not verified. As can be seen in Fig. 4.3a, the dataset G0600710 has a lower correlation (0.45 and 0.6) than the other two, and for this reason, such dataset was discarded.



(a) Over the humidity feature.

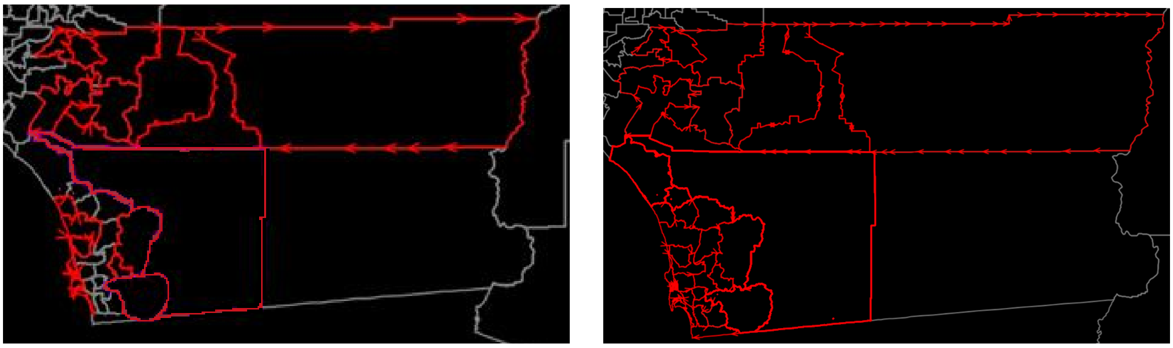
(b) Over the radiation feature.

(c) Over the radiation feature.

Figure 4.3: Pearson correlation matrices.

After selecting the two climate areas, a cross-over was implemented with the list of the regions that contain the highest number of PV systems in buildings. The resultant regions were introduced in the software JOSM using the correspondent GeoJSON file, which is available in the NREL database.

Fig. 4.4a represents the regions resulting from the explained analysis, and can be observed that there is a set of intermediate regions that were not selected. Such intermediate regions were not initially selected due to the lower share of PV, but they have buildings with PV that can be integrated into the considered regions.



(a) Initial set of regions chosen.

(b) Final set of regions chosen.

Figure 4.4: Selected regions printed on JOSM software

Therefore, to select these intermediate regions, the restriction of including only the regions with the highest number of buildings with PV was deleted. Then, Fig. 4.4b represents the final selection of regions. It should be noted that the large area in the lower right corner was not selected, since it belongs to a different climate area.

To have a final collection of datasets to test all functionalities of the proposed framework, it was decided to not use all available buildings without PV, in order to ensure a higher share of buildings with PV in the dataset. Therefore, $15\% - 20\%$ of the buildings without a PV system were selected. Such buildings will only have the forecast models for power demand. Table 4.1 presents the number of selected buildings per climate area with and without a PV system.

Table 4.1: Amount of buildings selected per climate area.

Climate area	**Number of buildings**		**Total**
	with PV	**without PV**	
G0600650	194	38	232
G0600730	303	65	368
Total	497	103	600

4.2.2 Features Analysis

In the selected dataset, there are two types of time series: stationary and non-stationary. In the stationary time series, the statistical properties do not change over time. The variables themselves can be distinguished between numerical and categorical: categorical data represents qualitative properties being more understandable using natural language description, and numerical data is for quantitative approaches. In order to convert categorical data into numeric values, the OHE was used.

Table 4.2 presents the variables chosen from the NREL dataset. The decision of this set was based on the objective of getting variables that are related to the electric energy performance of a building and with characteristics related to the PV system. It should be noticed that the variables with a n/a label in range/categories are considered non-stationary, and from the others variables, only the first three are numerical. Therefore, excluding these variables, the remaining variables needed to be submitted to OHE transformation to be used.

Table 4.2 have double lines to separate three different groups of features. The first one is the set of features that will be used by the system of forecast power demand, the second group will be used by the system of the generation forecast, and the third has the common features of the two systems, being this set provided by the third-party to the aggregator. Behind the presented features, for the system of forecast power demand, one additional feature was added: the labeling of the days, that is if the day belongs to a week or a weekend. This type of information is important due to the lower level of power demand associated with different days of the week.

The samples associated with the climate features are available with an hour frequency, instead of a quarter-hour like the other ones. For the objective of this framework, it is a better solution to do a forecast with a frequency of 15 min, being needed to add more samples to the climate features. To ensure it, a linear interpolation was implemented using a equation of a straight line, Equation 4.1, between two consecutive points, for example (x_1, y_1) and (x_2, y_2). Where m can be calculated using the two selected points, Equation (4.2), and b calculated with m and a single point of the two selected, Equation (4.3).

Table 4.2: Variables that were chosen from the NREL dataset.

Variables discription	**Units**	**Range / Categories**
Number of bedrooms	n/a	[1-5]
Number of occupants	n/a	[1 - 10]
Floor area	m^2	[30 - 1250]
Usage level of electric cooking range	n/a	eletric-100%; eletric-80%; other
Type of building foundation	n/a	ambient; heated basement; unheated bas. slab; unvented crawlspace; vented crawl.
Presence and type of cooling system	n/a	central AC; room AC; heat pump; none
Presence, type, and fuel of heating system	n/a	ASHP; baseboard; electric boiler; elect. furnace; shared heating; other or none
Orientation of the building	n/a	east; north; northeast; northwest; south; southeast; southwest; west
Presence of occupants	n/a	occupied; vancant
Year in which the building was constructed	n/a	<1940; 1940s; 1950s; 1960s; 1970s; 1980s; 1990s; 2000s; 2010s
Total electric energy consumed	kWh	n/a
Orientation of rooftop PV system	n/a	east; north; northeast; northwest; south; southeast; southwest; west
Capacity of rooftop PV system	kW$_{DC}$	0 - 5; 6 - 10; +10
Generation of rooftop PV systems	kWh	n/a
Temperature	C	n/a
Humidity	%	n/a
Global Solar Radiation	W/m^2	n/a

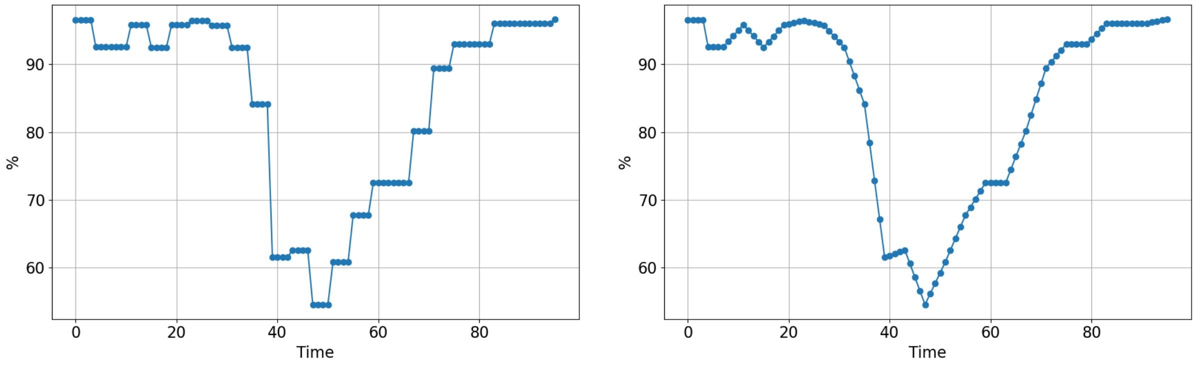
Equations (4.1) to (4.3) were used to calculate three middle points. In this way, instead of having a value per hour, the set of data has four values per hour, ensuring a frequency of 15 min. Another approach considered was to repeat a value for three additional points, but as can be observed in Fig. 4.5 such solution is less linear. This data can be tendentiously more outlying if a comparison with real data with the intended frequency (15”) is made.

$$y = mx + b \ , \tag{4.1}$$

$$m = \frac{y_2 - y_1}{x_2 - x_1} \ , \tag{4.2}$$

$$b = y_1 - mx_1 \ . \tag{4.3}$$

To finalize this adaptation the results were checked making a graphic that plotted the solar radiation and the generation, being identified as an offset of about two and a half hours. This is an error from the original dataset, probably caused by the existence of different time zones, since in normal conditions the radiation and the generation have an almost perfect correlation. After checking and confirming that the process of frequency adaptation does not cause this error, and the same error was verified in the two climate areas selected, it was confirmed that this is an error belonging to the original data. To fix it, an offset in the opposite direction was added. Figure 4.6 presents the solar radiation (blue line), the original generation (grey line), and the updated generation (yellow line).



(a) Direct interpolation method.

(b) Linear interpolation method.

Figure 4.5: Frequency adaptation for climate features, using one day of relative humidity.

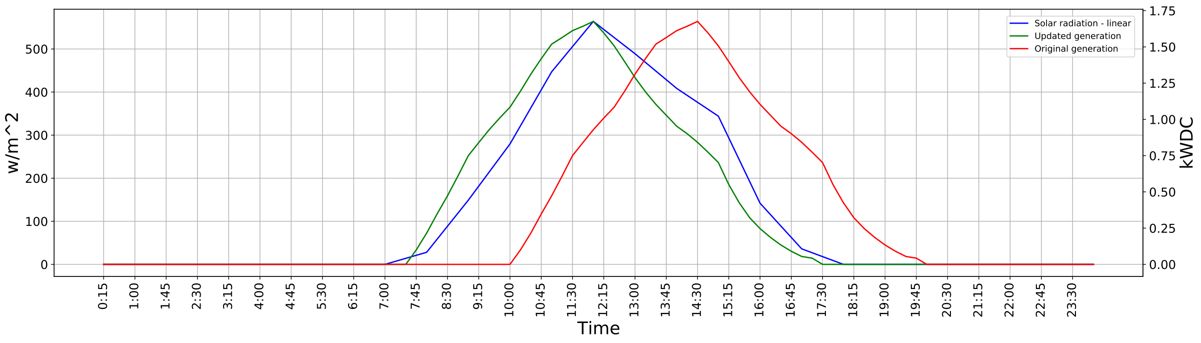


Figure 4.6: Solar radiation (blue line), original generation (red line), and updated generation (green line). Data of a day in the climate area G0600650.

4.2.3 Dataset Split

Each building will have a dataset for each system, meaning that all buildings will have a dataset to demand forecast and the buildings with PV will have another dataset for the generation forecast. These datasets have data from one year and needed to be split to train the systems and to test the learned models.

The test dataset usually is around $20\% - 25\%$ of the total dataset and is also intended to test the generalization of the ANN models relative to the seasonality variations. To accomplish this criterion, it was selected the set of data for April, August, and December for test. Thus a total of $24 \times 4 \times (30 + 31 + 31) = 8832$ samples were selected over a total of $24 \times 4 \times 365 = 35040$ available, achieving around 25% for test, as intended.

Chapter 5

Experimental Results

5.1 Architecture and Parameterization

5.1.1 Architecture

The type of local models needs to be equal for all clients, in terms of features and architecture. Therefore, based on related works (Section 2.4.3), it was decided to use a LSTM architecture. To enhance the performance of the local model, tests were made to decide the number of LSTM hidden layers, between two and three. Table 5.1 presents the ANN model structure with three LSTM hidden layers. For testing with only two layers, the same model is used but without a LSTM layer and the following dropout layer.

Table 5.1: Proposed Artificial Neural Network architecture.

Layer	**Type**	**LSTM cells**	**Activation Function**
LSTM	Input	64	Relu
LSTM	Hidden	32	Relu
Dropout(0.2)	Hidden	-	-
LSTM	Hidden	32	Relu
Dropout(0.2)	Hidden	-	-
LSTM	Hidden	32	Relu
Dropout(0.2)	Hidden	-	-
Dense	Output	-	-

5.1.2 Parameterization

To choose the most promising architecture, a test was realized and the results achieved by the two proposals were compared. The dataset presented in Section 4.1 was used, being employed two thirds of the dataset for training, and one third for testing. The parameters associated with the structure implementation were the number of hidden layers $\mathcal{N}_l$, the number of LSTM cells for the input layer $\mathcal{N}_{ci}$, number of LSTM cells for the hidden layers $\mathcal{N}_{ch}$, number of batch size $\mathcal{B}_s$, number of training cycles (epochs) $\mathcal{E}$, number of past observations $\mathcal{N}_{op}$, and number of forecast observations $\mathcal{N}_{of}$. More details of LSTM model can be founded in [64, 65].

Considering the predicted values $(\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_n)$, the real values $(y_1, y_2, \cdots, y_n)$, and the number of samples (n), the MSE is calculated using Equation (5.1), and the Root

Mean Square Error (RMSE) is calculated using Equation 5.2.

$$\mathrm{MSE} \ = \ \sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n} \ . \tag{5.1}$$

$$\mathrm{RMSE} \ = \ \sqrt{\mathrm{MSE}} \ . \tag{5.2}$$

Three tests were performed for each architecture: one with 2 LSTM, and other with 3 LSTM hidden layers, being the results presented in Table 5.2. The following parameters were used to test the proposed architectures: $\mathcal{N}_{ci}$ = 64, $\mathcal{N}_{ch}$ = 32, $\mathcal{B}_s$ = 32, $\mathcal{E}$ = 6, $\mathcal{N}_{op}$ = 96, and $\mathcal{N}_{of}$ = 1. The presented results demonstrate that the most promising option as architecture for the chosen parameters is a ANN with two LSTM in hidden layers, which managed to score the lower RMSE values on two of the three implemented tests.

Table 5.2: Artificial Neural Network architecture - performance test.

Nr. LSTM hidden layers	**Tests Scores (RMSE)**		
	1	2	3
2	20.619	5.822	13.889
3	11.479	27.182	22.150

5.1.3 Scenario B Parameterization

Scenario B (Section 4.2) has a total of 600 buildings, as presented in Table 4.1. Consequently, it required the train of 1097 ANNs models: 600 associated with the forecast of power demand, and 497 with the forecast of the generation. Due to the increment of models to train, by around 182 times, to implement this train in a reasonable time period, some local parameters of ANN were changed only for this scenario. Table 5.3 presents the made changes in the ANN parameters. The impact on the duration time, and also the achieved minimum loss value calculated using the MSE (Equation 5.1), for one client randomly selected. From the presented results a compromise needs to be done, or higher duration and possible better forecast models or a lower time period with the possible consequence of less reliable forecast models. Being the aspect of duration a crucial aspect due to the increase of ANNs models that needed to be trained, the proposed changes on the ANN parameters were used. The remaining parameters associated with the locals ANNs remain the same, with $\mathcal{N}_{po}$= 96, and $\mathcal{N}_{fo}$= 1. The global variables were set as: $\mathcal{R}_d = \mathcal{R}_g$ = 32, and $\mathcal{K}$= 9. Figure B.1, in Appendix B, presents two prints of an example of two trains, where the first uses the B parameters, and the second uses the A parameters, being outlined in red the time for each epoch.

Table 5.3: Impact by changing the Artificial Neural Network parameters.

	ANN parameters				**Train Time**		**Min. loss value**	
	$\mathcal{N}_{ci}$	$\mathcal{N}_{ch}$	$\mathcal{B}_s$	$\mathcal{E}$	**Demand**	**Generation**	**Demand**	**Generation**
A	64	32	32	6	4' 46"	4' 30"	0.183	0.047
B	32	16	96	5	1' 24"	1' 32"	0.416	0.110

5.2 Forecast Systems

To enhance the forecast of net-demand, different tests were implemented to decide the forecast system. According to Equation 3.3, the net-demand can be predicted using two systems of the forecast, one for generation and a second for power demand. However, another approach would be to directly forecast the net-demand.

Using the dataset presented in Section 4.2, since this dataset also has generation information, a random building with a PV system was selected. Tests were made using a local ANN model with the architecture and parameterization defined in the previous section (parameters for Scenario B), for all forecast systems. Table 4.2 has the features of the two forecast systems. In the case of a direct forecast of the net-demand, only the features related to the demand can be used. The generation features were not added because a building can have or does not have a generation system and the ANN model structure needs to have the same features independently of the buildings. Therefore, for that reason, only features related to the forecast of the power demand can be used.

Using the RMSE as an evaluation metric, the three forecast systems were tested: a direct forecast of net-demand, and an indirect way using two systems: one to forecast demand, and a second to forecast generation. The results are presented in Table 5.4. It is possible to conclude that the forecast of the net-demand through the forecast of demand and generation, in an independent way using the Equation 3.3 achieves a decrease of approximately 30% when compared with a direct forecast of net-demand. Such results allow concluding that a separated forecast model will have the most promising results for the intended objective.

Table 5.4: Forecast systems - performance test.

	Single system	**Two systems**		
	net-demand	demand	generation	net-demand
RMSE	0.257	0.132	0.149	0.182

5.3 Framework Debugging

The debugging is a very difficult task, mainly due to the complexity of the framework which can generate multiple different situations. The first trains occurred with a small number of buildings only to test the framework and check possible errors. After completing this debugging phase, the complete framework was tested with all buildings.

Due to the higher number of models operating in the realized test considering all buildings (600 for the demand system and 497 for the generation system), making a total of 1097 models to train, to save time, it was decided to fix the errors and activate the *rst* variable to continue the train. Additionally, a second machine was used to analyze the results achieved during the train. With these strategies, it was necessary to stop the train more 3 times until solving all errors and getting a fully functional framework. Therefore, the first complete test done over the developed framework was not a success in terms of being done over a fully functional framework, since the occurred errors can have influenced the final results. For this reason, it was decided not to use these results, and consider this test only for the debug phase.

The test implemented over the full group of buildings proves the proper function of the implementation of the framework, and the option of continuing the train by choosing a previous set point defining the rst variable as $False$. It allowed also the optimization of some algorithms, as well as the addition of new resources such as the control dataset. Before this implementation, a strategy was being used that did not allow the visualization of critical parameters associated with training. Lastly, it was added the implementation of an algorithm that records all the α values associated with each model. This resultant dataset allows visualizing the behavior of the early stopping algorithm adapted for the developed framework (Algorithm 2) during the training process.

5.4 Results of Scenario A

In that initial stage, some algorithms, like the Local Evaluation, were not yet developed. Therefore, the following parameters were used to test the proposed structure: $\mathcal{R} = 60$, $\mathcal{K} = 3$, $\mathcal{N}_l = 2$, $\mathcal{N}_{ci} = 64$, $\mathcal{N}_{ch} = 32$, $\mathcal{B}_s = 32$, $\mathcal{E} = 6$, $\mathcal{N}_{op} = 96$, and $\mathcal{N}_{of} = 1$. It should be noted that, instead of defining a learning rate value, the Adam optimizer was used, which inherently uses an adaptive learning rate method.

Table 5.5 presents the obtained results: the θ value is the number of rounds that each building participated, $\mathcal{L}_{training}$ is the lower loss value, and r is the corresponding round number of the global training. p is the number of local rounds that each building participated in before round r. Put it differently, a building achieves the lower loss value $\mathcal{L}_{training}$ on round r, where $r \in \mathcal{R}$, and to arrive at that around the building participated on p rounds. However, the same building participated in a total of θ rounds. The remaining variable is the RMSE calculated using the Equatinon (5.2) over the loss results achieved when the test data was applied over the model saved on round r.

Table 5.5: Achieved score results.

Building ID	θ	Results on Training			Results on Test
		$\mathcal{L}_{training}$	p	r	RMSE (r)
DCE	14	0.06878	12	27	5.39172
DChE	17	0.03807	15	27	4.19676
DECE	17	0.02663	8	14	3.25616
DES	13	0.03695	10	23	4.85933
DIE	21	0.03076	20	31	3.02867
DME	17	0.04138	16	31	5.89367

As can be concluded from the results presented in Table 5.5, the proposed FL setup ensures the participation of each client in the global training, in between 13 and 21 local trains. It can also be observed that in the round where achieved the lower loss values occur before the last local train ($p < \theta$). This result shows the need for the implementation of the global evaluation over the loss values, to detect when the models start converging. On the final version of the framework, this algorithm was implemented (Algorithm 2).

Taking the building Department of Chemical Engineering (DChE) as an example, Figure 5.1 presents the loss value as of the global training convergences throughout rounds. It should be noted that in Figure 5.1 the rounds start with the number zero, and in Table 5.5 start with one. Therefore, the r value in Table 5.5 needs a decrement of one unit to match with the number presented in Figure 5.1. MSE (Equation (5.1)) was used as the evaluation metric to obtain the discussed results. The loss graph showcases the decrease throughout the rounds and demonstrates learning over the course of training. Overlapping this graph with the green vertical lines depicts that the loss presents higher values when the Federated Average calculus is implemented.

Figure 5.2 presents the predicted and real values for the test dataset for the building DChE, to show an example of the net-demand prediction achieved by the model, where the lower loss value on the training was achieved. Comparing the obtained results over different months reveals the adaptability of the proposed model with respect to the seasonal variations.

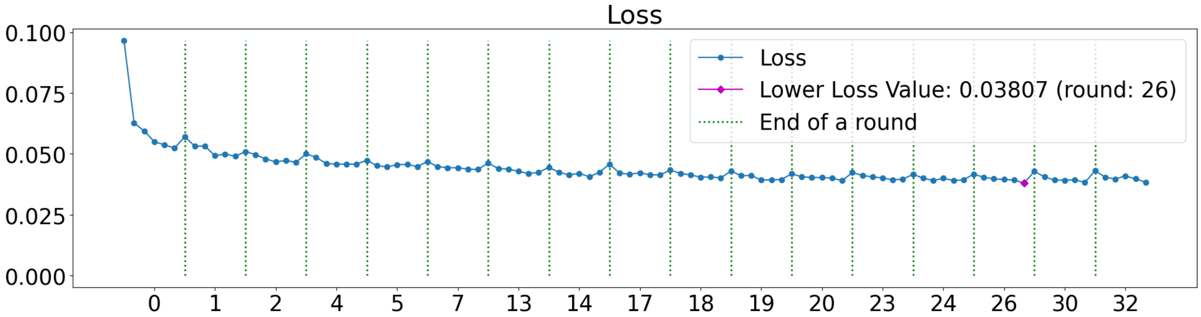


Figure 5.1: MSE values obtained throughout the global training in the Department of Chemical Engineering (DChE) building.

5.5 Results of Scenario B

This section presents the results obtained with the proposed FL framework on Scenario B (Section 4.2). A representative group of buildings was selected to test the developed framework, by selecting 36 buildings with PV system. The group of buildings is split in half according to the climate area. Therefore, 18 buildings are from the G0600650 climate area, and the other 18 belong to the G0600730 climate area.

In Section 3.2.1, the control and evaluation of the FL realized by the server were presented. In this process a dataset of control was created to record different variables, being this dataset called the “control dataset”. The control dataset has information on the total number of rounds that each client participated in. With this information, the Participant Selector Algorithm (Algoritm 3) can be applied, which has the objective of keeping the number of rounds balanced between the clients, but always with associated randomness. Figure 5.3 presents the obtained results. It can be observed that the average number of rounds is 8 (in booth systems), and the number of rounds above and below the average, for about 95% of the buildings, is only one. The higher difference in the number of rounds is caused by a lower number of rounds, but the algorithm prioritizes such clients to be selected in the next rounds. Therefore, it is possible to conclude that the algorithm is reaching its goal. Additionally, the randomness of the code can be verified through the non-linearity of the number of rounds participated by each client.

The control dataset also provides information on the minimum values achieved for the loss, $\mathcal{L}_{min}$, in each client. This values are calculated using the MSE (Equation (5.1). Figure 5.4 presents the minimum loss values, being possible to identify the clients that achieve the lowest MSE values (represented by green color). For the demand system, the client with the lowest values is identified with the ID= 547267 who accomplish an

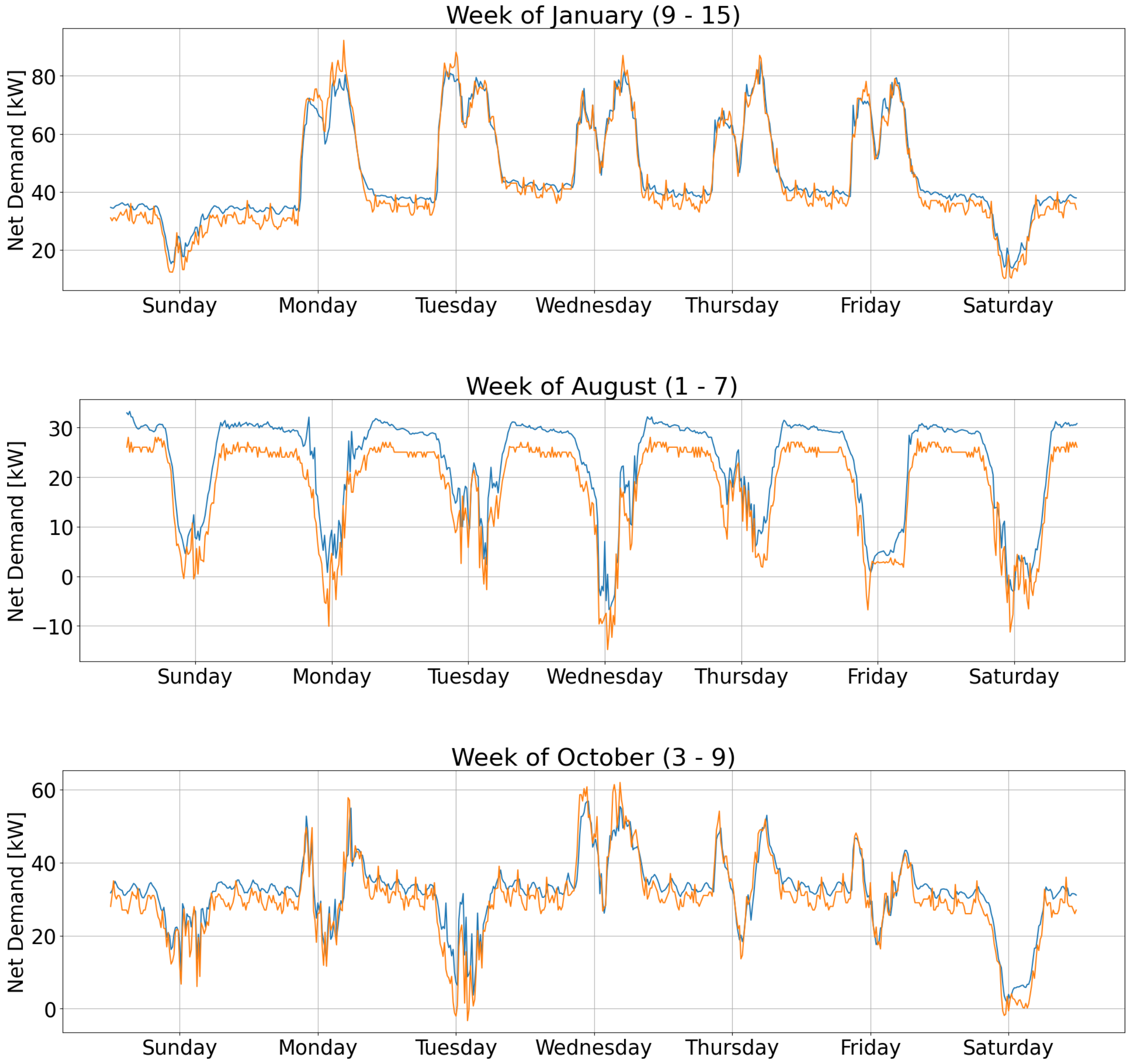
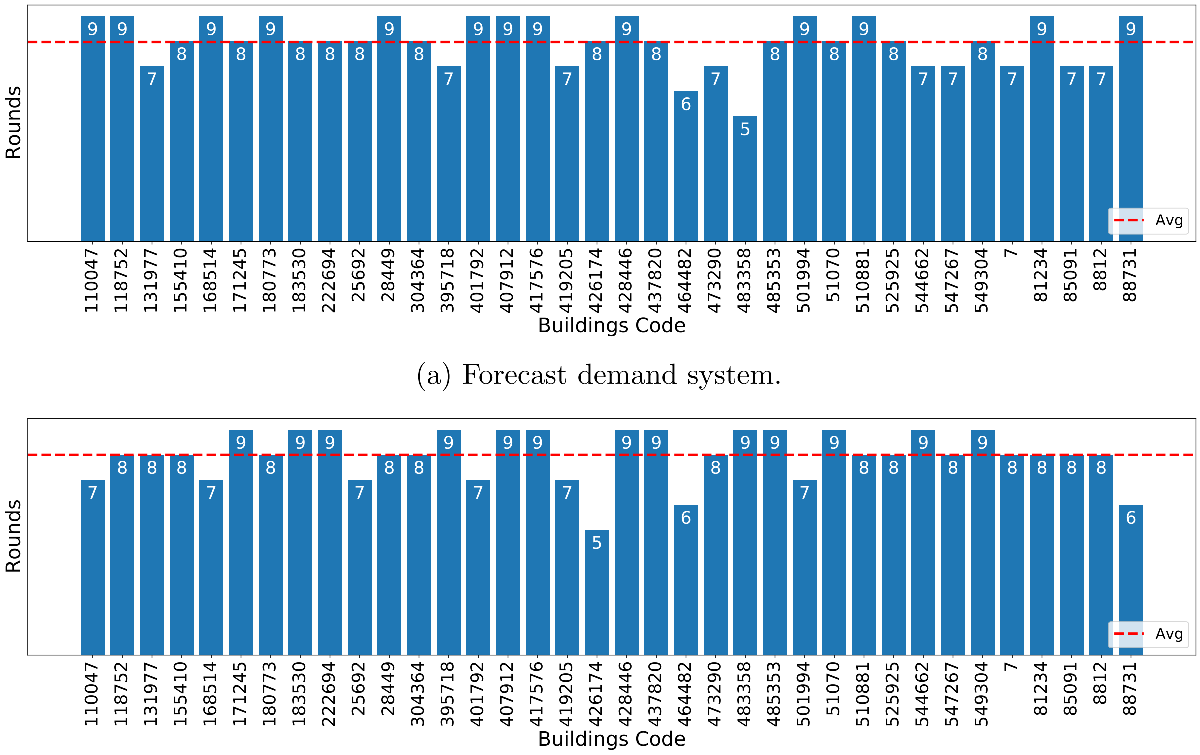


Figure 5.2: Net-demand in the building Department of Chemical Engineering (DChE) in the global round $r = 27$ (real values in blue and prediction in orange).



(a) Forecast demand system.

(b) Forecast generation system.

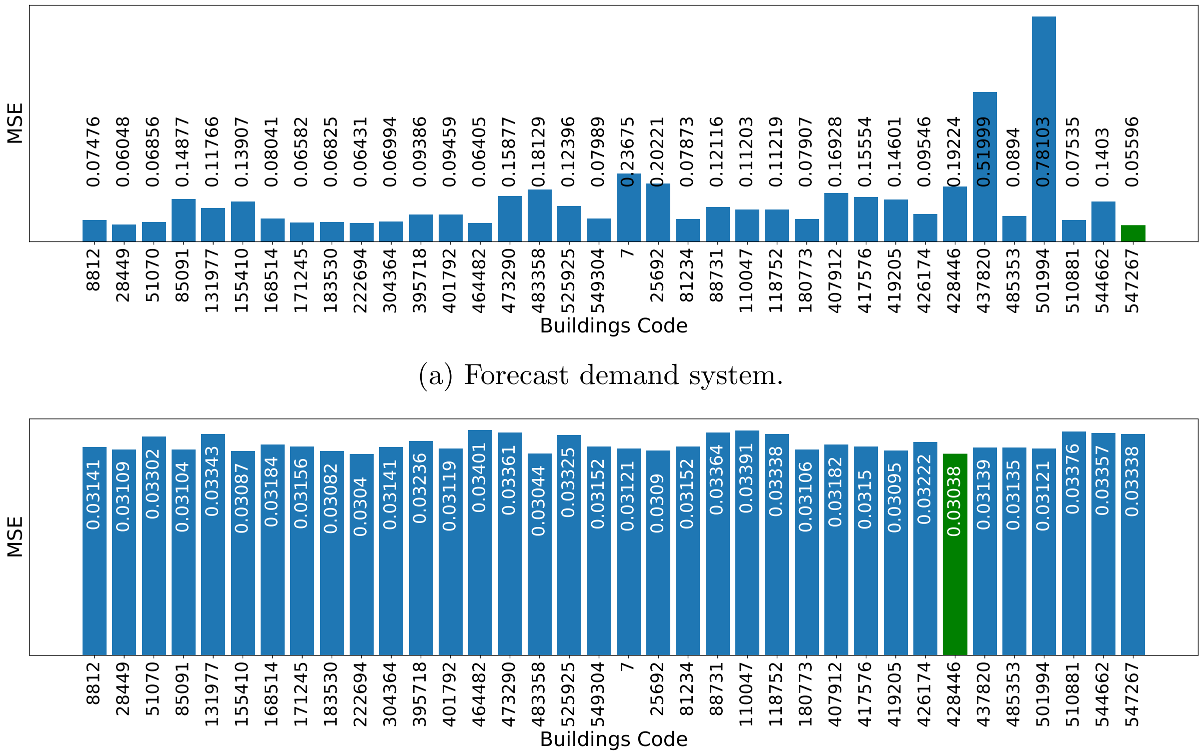
Figure 5.3: Number of rounds per client in the realized test.

$\mathcal{L}_{min}$= 0.5596, and this client participated in 7 rounds in this system. For the generation system, it is the client identified with the ID= 428446 who accomplish an $\mathcal{L}_{min}$= 0.0.03038, and this client participated in 9 rounds.

From Figure 5.4 is noticed that the minimum loss achieved by each client, is more similar in the generation forecast system. This difference is explained since the generation profile has a higher correlation between the different buildings. Additionally, this system has a high correlation with the weather features, who are shared by all buildings, and consequently, the performance of the forecast models is similar.

The results of client 501994 on the demand system can also be highlighted since they present a MSE value significantly higher in comparison with the other clients. During the analyses of the loss results for this client, it was observed the existence of 10 recorded datasets of training, which is not in agreement with the information presented in Figure 5.3a, where only 9 rounds are registered. Checking the control dataset of the server for this client, one local error was registered, justifying the identified mismatch. Since this do not directly justify the high MSE value, the α values were studied to detect the global convergence (Algorithm 2).

Figure 5.5 presents a plot of the values of the loss for the client 501994, as well as the respective α values. The values marked in green are the minimum MSE values in a round. The beginning of a round is marked with a dotted green line. Through this analysis, it was possible to verify the good behavior of the Algorithm 2, since after the first decrement of the α, in the following round the $\mathcal{L}_{min}$ value decreases and a reset is done, respecting the imposed conditions. In the latest two consecutive rounds, the value of $\mathcal{L}_{min}$ does not change, since the train for that client ended. This client presents the train without any indication of a possible error in the FL training. It can then be admitted that this type of



(a) Forecast demand system.

(b) Forecast generation system.

Figure 5.4: Minimum loss values achieved per client.

case can occur because the local models do not get the capability to learn as well as the others. The clients with the lowest $\mathcal{L}_{min}$ value recorded for each system, ID = 547267 for the demand system, and with ID = 428446 for the generation system, were selected to test the forecast of the net-demand. The validation loss recorded in each round was used to choose the most promising model. This selection was done by picking the model of the round where the lowest MSE value of the validation is registered. It should be noted that the algorithm of the local train has a function that saves only the "best" model. Where this "best" model is the model of the epoch where the lowest MSE value was recorded.

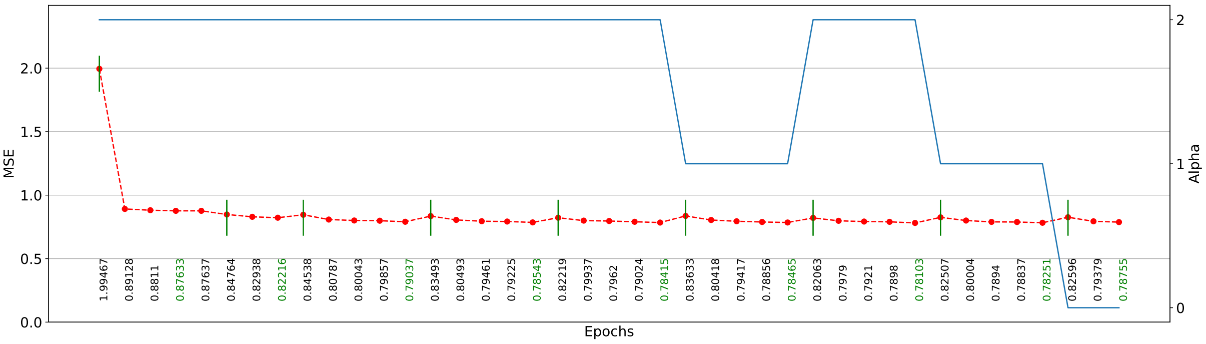


Figure 5.5: Loss (red line) and αs values (blue line) recorded in the forecast demand system for client 501994. The vertical green lines represent the beginning of a round.

Figure 5.6 presents the graphs for the validation for each system to the selected clients (547267 and 428446). In agreement with the implemented strategy, the model was saved using the same logic to select the select round, and consequently the most promising model. From the analysis done over the graphs the following models were selected:

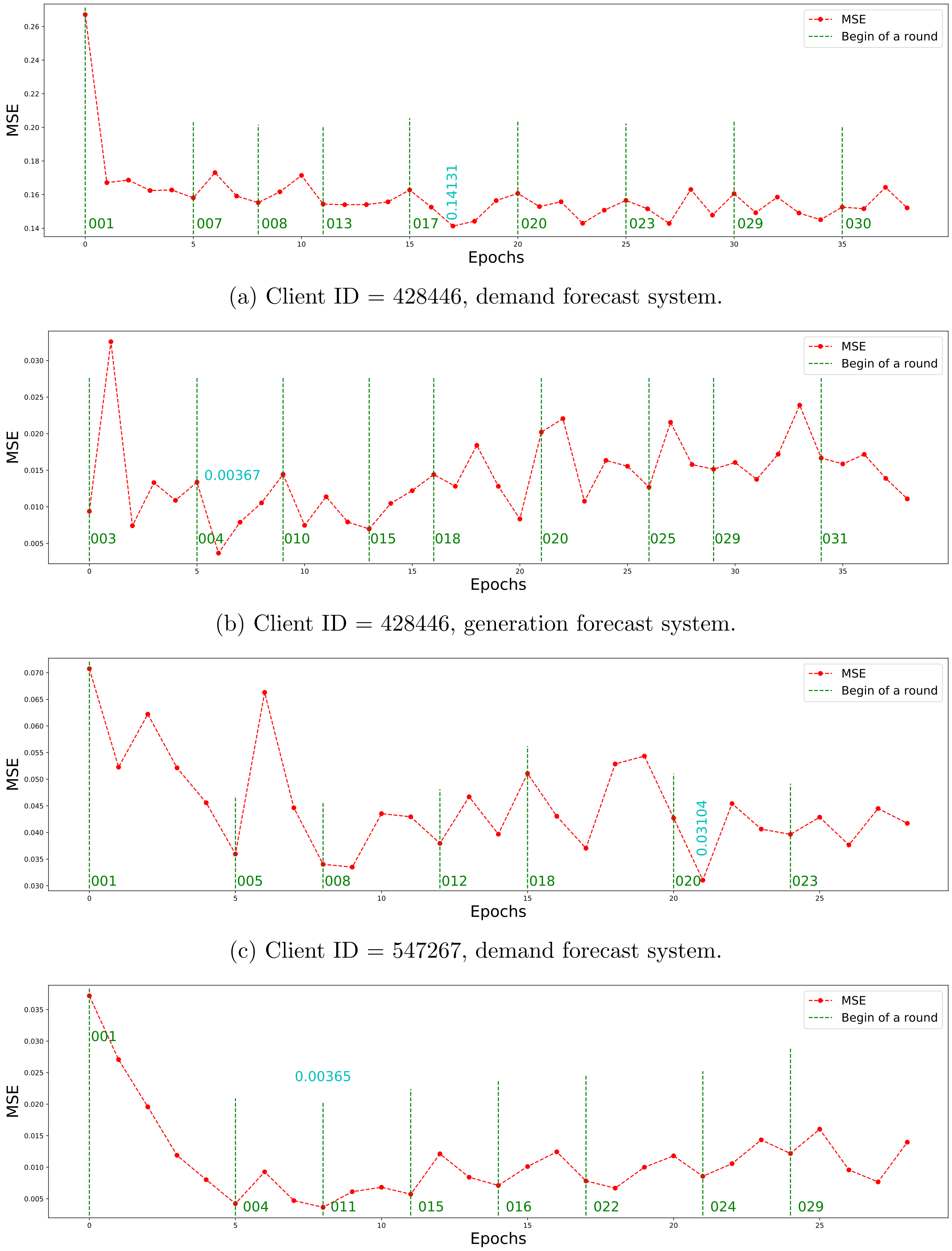
- Client ID = 428446:
  - Demand system: $r = 17$ (Figure 5.6a);
  - Generation system: $r = 4$ (Figure 5.6b).
- Client ID = 547267:
  - Demand system: $r = 20$ (Figure 5.6c);
  - Generation system: $r = 11$ (Figure 5.6d).

Section 4.2.3 exposes how the split of the data was done, being used for testing data from April, August, and December. The $\mathcal{N}_{po}$ and $\mathcal{N}_{fo}$ are the parameters to adjust the data. As $\mathcal{N}_{po} = 96$, and the data have a sampling period of 15 minutes the model will use the data collected for 24 hours to predict a unique sample ($\mathcal{N}_{fo} = 1$). Taking as an example the month of April, as the data is relative to the year 2018, the first day is a Sunday. Therefore, the first sample predicted, for this case, will be the first 15 minutes of Monday. Figure 5.7 and Figure 5.8 present the forecast of the net-demand relative to the first week available on each training dataset (April, August, and December).

Figure 5.7 and Figure 5.8 present the forecast for three different periods of time with different weather conditions. Table 5.6 presents the values of the metrics in the realized test on the selected clients. The metrics are the same used previously: the MSE (Equation (5.1)), and the RMSE (Equation (5.2)). Although these two clients belong to the same climate area. the results allow to observing the good adaptation of the models in terms of seasonality. Another aspect is the selected round number. For example, client ID = 428446 participated in 4 rounds before the round where the model was selected for the demand system. For these clients, in none of the cases the selected model was the first participating round. Therefore, the clients sharing their learning enhances the learning process of other clients, demonstrating s a good collaboration in a TE community. It is also possible to verify in these results the existence of negative values, which happens when the generation is higher than the demand (Equation (3.3)).

A second test of the framework was implemented with the same parameterization. However, this second test uses two other regions, and 5 of the 26 buildings that make part of these regions are buildings without an integrated PV system. The objective of this test was to show the implemented framework running with buildings without the need for generation system. Additionally, it also uses the train previously implemented to study the training process in a similar (but different) community. The buildings without a generation forecast are identified by the following IDs: 662, 2513, 1866, 8156, and 10687.

Figure 5.9 presents the rounds realized per client before ending their training. Making a comparison between Figure 5.3 and Figure 5.9 it is possible to verify that using the FL framework in a new community through continuous training from a pre-implemented framework in another community speeds up the learning process. In the first train, where the implemented framework runs for the first time, the average of the rounds was 8, for booth systems (Figure 5.3). In the second test (Figure 5.9), for the demand forecast, each client needs on average around 5 rounds to implement the training, and for the generation forecast, each client needs on average between 4 and 5 rounds.



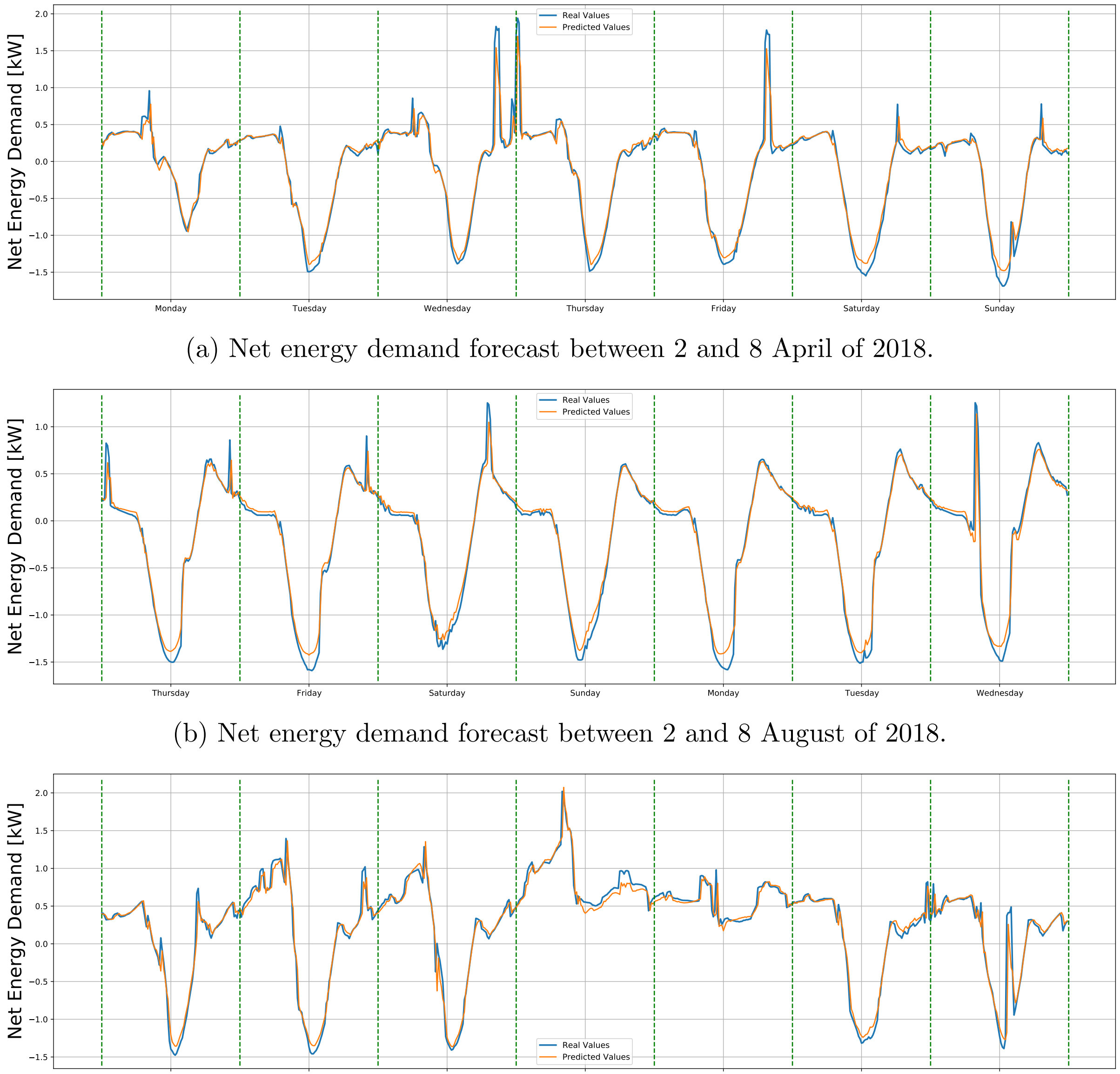
(a) Client ID = 428446, demand forecast system.

(b) Client ID = 428446, generation forecast system.

(c) Client ID = 547267, demand forecast system.

(d) Client ID = 547267, generation forecast system.

Figure 5.6: MSE values obtain from the validation test done in each epoch.

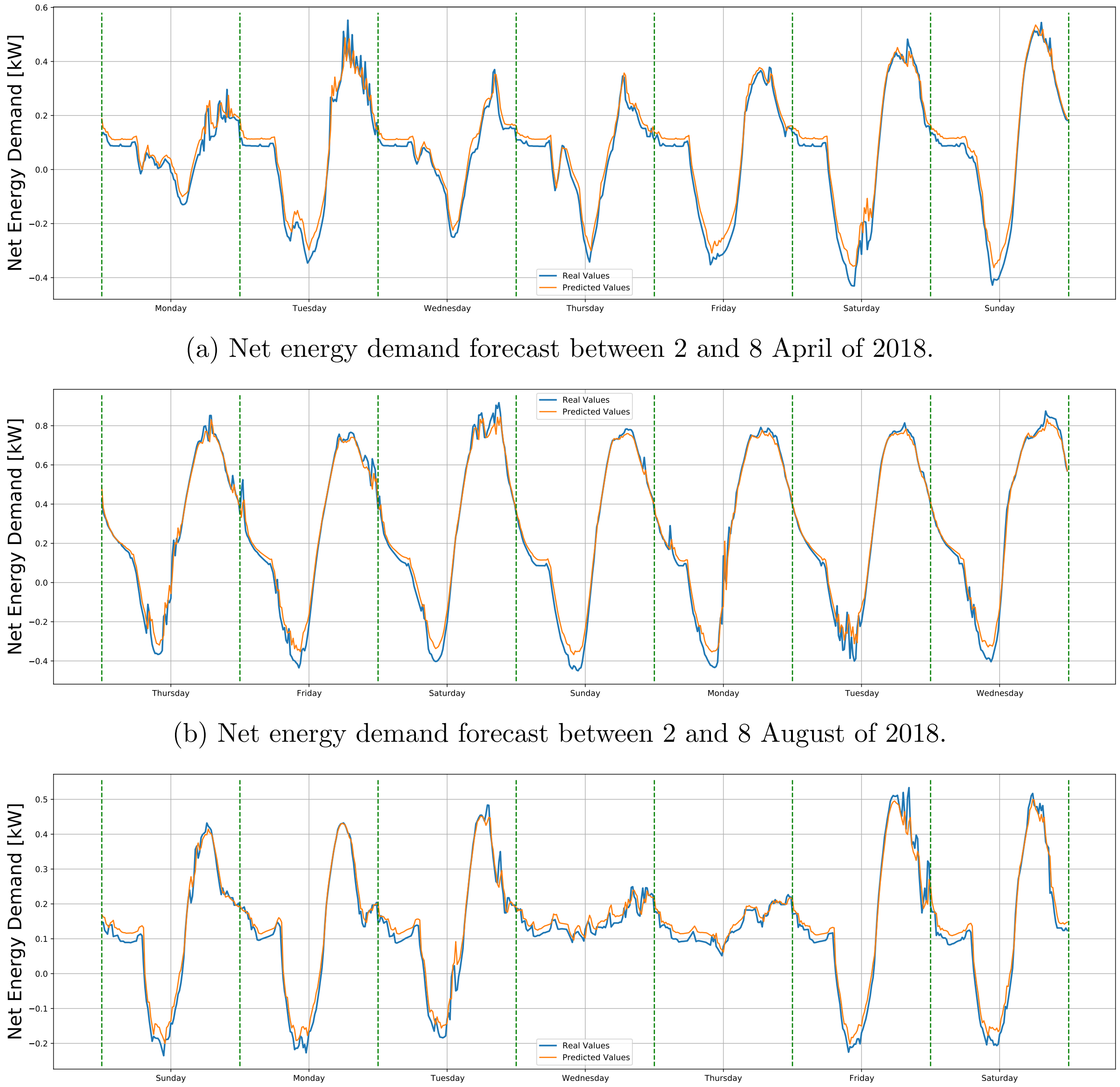


(a) Net energy demand forecast between 2 and 8 April of 2018.

(b) Net energy demand forecast between 2 and 8 August of 2018.

(c) Net energy demand forecast between 2 and 8 December of 2018.

Figure 5.7: Net energy demand forecast during a week for the client ID = 428446, in three different months.



(a) Net energy demand forecast between 2 and 8 April of 2018.

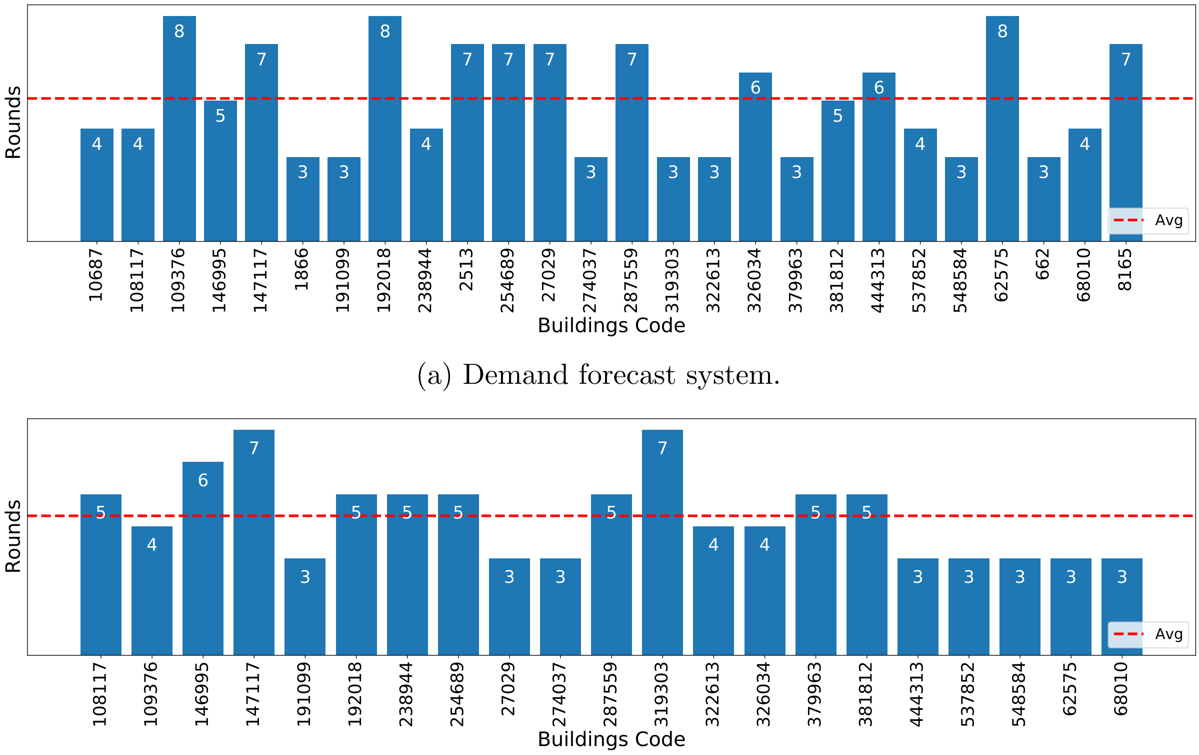
(b) Net energy demand forecast between 2 and 8 August of 2018.

(c) Net energy demand forecast between 2 and 8 December of 2018.

Figure 5.8: Net energy demand forecast during a week for the client ID = 547267, in three different months.

Table 5.6: Achieved score results for two clients, in scenario B.

Client ID	Month	System	MSE	RMSE
547267	**April**	*generation*	0.00036	0.03515
		demand	0.00128	0.03537
		net-demand	0.00211	0.04594
	August	*generation*	0.00025	0.03283
		demand	0.00148	0.03482
		net-demand	0.00191	0.04365
	December	*generation*	6e-05	0.02543
		demand	0.00087	0.03066
		net-demand	0.00101	0.0317
428446	**April**	*generation*	0.00236	0.13105
		demand	0.02274	0.15678
		net-demand	0.02642	0.16254
	August	*generation*	0.00135	0.09334
		demand	0.01174	0.11134
		net-demand	0.01306	0.11427
	December	*generation*	0.00097	0.11991
		demand	0.02068	0.14521
		net-demand	0.02149	0.14661



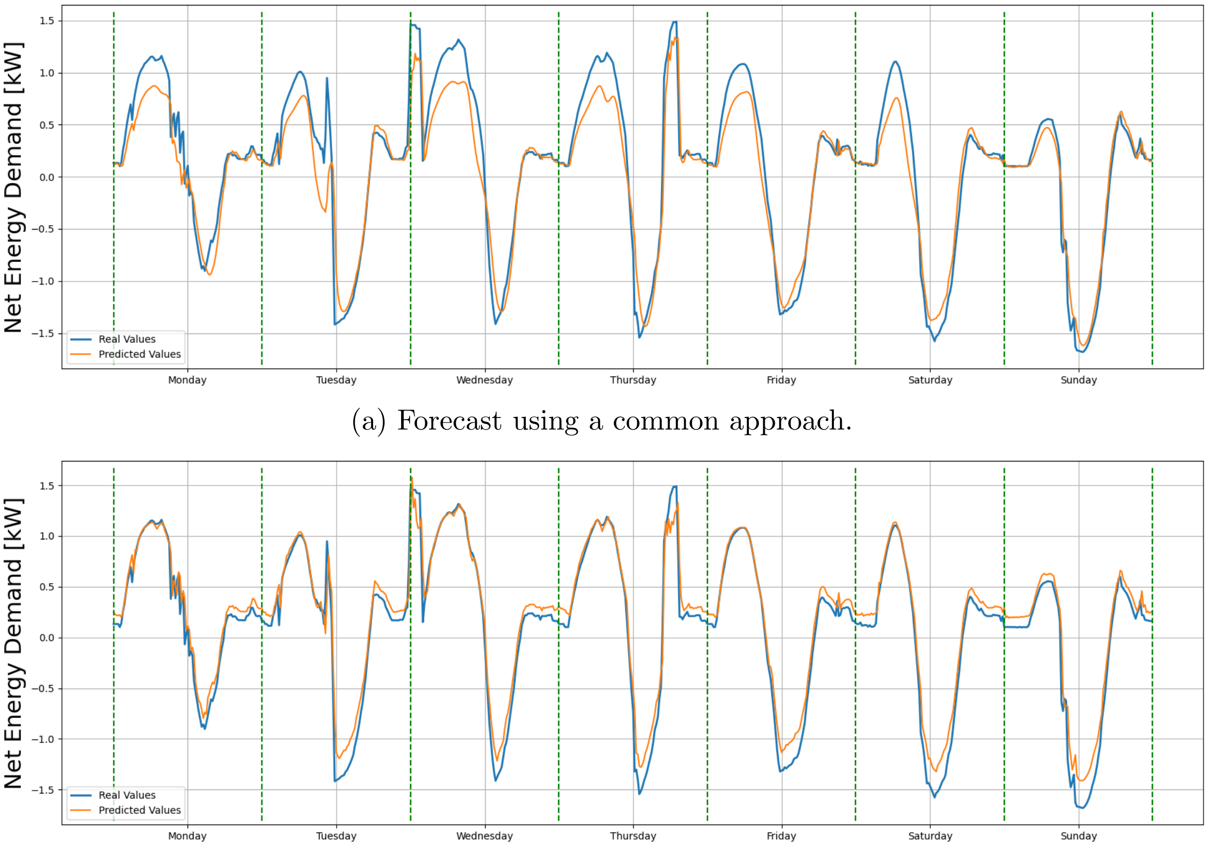
(a) Demand forecast system.

(b) Forecast generation system.

Figure 5.9: Number of rounds per client in the second realized test.

To analyze the performance of the developed FL framework in comparison with a more common approach, a complementary test was done. In this test, a local model was trained for the demand forecast in the building where the minimum value for the loss was achieved in the first test (Figure 5.4a), the client with ID = 547267. In the system of generation forecast, the client with ID = 428446 was the client with the lowest loss value (Figure 5.4b), being the generation model trained using the data from such client.

Using a common approach, when a new building is added to a community, these two trained models are the most promising options to achieve the highest accuracy in the prediction of the net energy demand in the new building. However, in the proposed FL framework, this new client can be any of the clients of the second implemented test. Therefore, by randomly selecting one of these clients, the client with the ID= 68010 was selected. Figure 5.10 compares the forecast using the common approach and the forecast using the model trained using the proposed FL framework. Table 5.7 presents the MSE and RMSE values for the two forecasts, where the metrics achieved with the model using the FL structure are 55% lower than those achieved with the common approach.



(a) Forecast using a common approach.

(b) Forecast using the implemented FL approach.

Figure 5.10: Net energy demand forecast for client ID= 68010, with a common and a FL approach.

Table 5.7: Achieved score results for client ID= 68010, with a common and a FL approach.

Forecast	**MSE**		**RMSE**	
	FL	common	FL	common
Net energy demand	0.02329	0.07536	0.1526	0.27452
Demand	0.01359	0.06597	0.13578	0.26583
Generation	0.00574	0.00589	0.11918	0.22153

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This dissertation starts by introducing the theoretical concepts of Smart Grid (SG), Machine Learning (ML), and Federated Learning (FL) following a literature review of the same concepts. Relatively to the SGs, its concept was introduced, as well as its evolution and the main need to integrate the concept of Transactive Energy (TE) communities. TE will unlock new opportunities for new services, that will be crucial for reliability and optimization of future Power System, where traditional power plants will be replaced by renewal generation, and where the control of flexibility resources will be fundamental.

To enhance the control of such resources, systems to forecast demand and generation are needed. To understand the forecast systems that can be applied to ensure the forecast of variables like energy generation, demand and net-demand, the concepts of ML were studied. ML is a computer program that has the capability to learn from his experience and it was concluded that Artificial Neural Network (ANN) is the principal ML method used for the prediction of energy in buildings. Such a study also allowed to understand that private features and collaboration between buildings can enhance the forecast systems for these variables. However, the sharing of private features can be a concern from the user's point of view. Therefore, to find a solution that embraces collaboration and protection of the data the concept of FL was studied. FL was firstly introduced by Google and applied to Gboard systems for auto-completion of words. In a simple way, FL trains a ML model on edge devices, and instead of sending the data to the server sends the weights of each model. In the server, the weighted average of the weights is applied. In this way, it allows collaboration between multiple end devices without sharing any private data.

This dissertation had the main objective of developing a novel framework to be implemented in a Transactive Energy community based on FL to enhance the prediction of net energy demand. The use of a framework based on FL, will have the advantage of ensuring a local forecast of the main variables crucial for the building management systems, enhancing the forecast models using private data, while ensuring private data protection, as well as ensuring a more accurate forecast of the net energy demand. Therefore, this dissertation had the purpose of developing a FL structure adapted to the architecture of the SG, as defined by the NIST model. To match both systems, it was decided to use a FL system with a client-server architecture, where the server will ensure the connection to the Distribution System Operator. Behind this architecture, a Horizontal Federated Learning approach was selected, since all clients have the same features and the same target. Traditionally, FL has a unique forecast system, being in this work adapted to work with two forecast systems (one for power demand forecast, and the second for generation forecast). A connection between the server and a third-party was also added to provide additional data (non-private and common to all clients).

First, the framework was implemented only to forecast the power demand, using the values of the generation to ensure a complete study of the net-demand. This first phase also ensured the integration of the third-party data provider, as well as the implementation of the algorithms for the aggregator and for the clients. The second stage was the development of the full framework, integrating the second system to also predict the generation, as well as the improvement of some algorithms and the addition of new ones. For example, the Algorithm 2 adopted the early stooping technique to the FL environment, in order to analyze the global learning process of a forecast model.

Scenario A was used to test the first phases of the implementation. This scenario was constituted by six buildings that are part of the Polo 2 campus at the University of Coimbra, using real data values of the net-demand. Only for one building, the Department of Electrical and Computer Engineering, it was possible to collect the demand and the generation, separately. Such data was used as the basis for the created dataset for this scenario, extrapolating the generation for the other buildings and enabling splitting the generation from the demand. The academic calendar and the weekdays were added using the One Hot Encoding process, and with an implemented algorithm for web scrapping weather data information (temperature, radiation, and humidity) were collected from the Wunder Ground website.

Scenario B was used to test the fully developed framework. This scenario uses synthetic data for several buildings from all states of the United States of America for the 2018 year, provided by National Renewable Energy Laboratory. Behind energy data, the dataset also contains other features related to the characteristics of the buildings and the weather. The state of California was selected due to a higher percentage of buildings with a photovoltaic (PV) system. This database also ensured the separation of buildings by climate areas and regions. Therefore, by assessing the correlation, it was possible to reduce the climate areas to two, being selected a set of buildings with and without PV to constitute Scenario B. It should be noted that all these buildings are residential. The same database also has commercial buildings, but such buildings were not included due to the very low penetration of PV system.

The implemented test for the first stage, with the datasets from scenario A, was the first test done over the implemented FL structure. Such a scenario tested the flow of the implemented framework and the capability of learning the local forecast models. Therefore, it was possible to conclude that seasonal variations are well detected by the local forecast models. It was also concluded that such models also reveal good learning capabilities, achieving good results in the forecast.

Scenario B allowed the implementation of a final and complete test over the developed framework, where all implemented algorithms were tested. From these results, it was possible to conclude that the framework is able to ensure all defined objectives. The capability to work with two different forecast systems separately and the algorithms developed to achieve the most promising models resulted in good results for the forecast of net energy demand, as intended. The forecast results show the capabilities of learning from different profiles and collaboratively learning from each other to enhance their forecast models. Additionally, with the results of the implemented tests, it was possible to demonstrate the behavior of crucial algorithms implemented in this phase, namely the adaptation of the early stopping technique into the FL. The second test done over this scenario proves that this approach can be also very important in terms of making easier the implementation in new communities. The FL framework implemented when tested in a new community reveals faster learning of the profiles from the users, and also the capability to work with buildings that do not have an PV system integrated. Finally, the last test shows that the implemented FL framework is able to achieve more promising results than a more common approach. This test achieved a decrease of 55% on the RMSE value when, instead of the more common approach, the framework proposed in this dissertation is adopted.

6.2 Future Work

In future work, the objective is to continue improving the developed framework by integrating new algorithms. In the server, a forecast system for weather needs can be created to have a more complete system of forecasts. Optimization algorithms to achieve more promising models can also be studied and applied to the framework.

In terms of optimization, two main tasks can be defined:

1. Adapt the code to the hardware where it will be running;
2. Tunning of parameters.

The code was developed in python, where there are many options of optimization according to the used hardware. In terms of parameterization tunning, this can be approached by firstly implementing a local tunning in the local models, for example with a grid search technique. After it, instead of applying the Adam optimizer. the approach can be based on the use of a value for the learning rating according to the round number [66].

Additionally, the integration of the developed framework to improve other optimization algorithms and create a full system of energy management can also be studied. For example, the development of a framework for effective management of energy resources (such as battery energy storage and electric vehicles) ensuring economic and technical benefits for buildings, communities, and utilities, taking advantage of the improved predictions provided by the FL models [67, 68, 69, 70, 71].

Bibliography

[1] UE. Acordo de paris sobre alterações climáticas. `https://www.consilium.europa.eu/pt/policies/climate-change/paris-agreement/`, 2 2022. Accessed: 09-06-2022.

[2] Républica Portuguesa. Roteiro para a neutralidade carbónica 2050. `https://unfccc.int/sites/default/files/resource/RNC2050_PT-22-09-2019.pdf`, 6 2019. Acessed: 09-06-2022.

[3] Cheng Feng, Yi Wang, Qixin Chen, Yi Ding, Goran Strbac, and Chongqing Kang. Smart grid encounters edge computing: opportunities and applications. *Advances in Applied Energy*, 1:100006, 2021.

[4] Khizir Mahmud, Behram Khan, Jayashri Ravishankar, Abdollah Ahmadi, and Pierluigi Siano. An internet of energy framework with distributed energy resources, prosumers and small-scale virtual power plants: An overview. *Renewable and Sustainable Energy Reviews*, 127:109840, 2020.

[5] Filomeno M Vieira, Pedro S Moura, and Aníbal T de Almeida. Energy storage system for self-consumption of photovoltaic energy in residential zero energy buildings. *Renewable Energy*, 103:308–320, 2017.

[6] Alberto Sendin, Miguel Sanchez-Fornie, Inigo Berganza, Javier Simon, and Iker Urrutia. *Telecommunication Networks for the Smart Grid*, volume 59. Artech House, 12 2016.

[7] Pierluigi Siano, Giuseppe De Marco, Alejandro Rolán, and Vincenzo Loia. A survey and evaluation of the potentials of distributed ledger technology for peer-to-peer transactive energy exchanges in local energy markets. *IEEE Systems Journal*, 13:3454–3466, 2019.

[8] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G L D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konecný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X Yu, Han Yu,

and Sen Zhao. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14:1–210, 2021.

[9] Yang Chen, Xiaoyan Sun, and Yaochu Jin. Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation. *IEEE Transactions on Neural Networks and Learning Systems*, 31(10):4229–4238, 2020.

[10] Chen Fang, Yuanbo Guo, Na Wang, and Ankang Ju. Highly efficient federated learning with strong privacy preservation in cloud computing. *Computers & Security*, 96:101889, 2020.

[11] Xinghua Zhu, Jianzong Wang, Zhenhou Hong, Tian Xia, and Jing Xiao. Federated learning of unsegmented chinese text recognition model. pages 1341–1345, 2019.

[12] Behrad Bagheri, Maryam Rezapoor, and Jay Lee. A unified data security framework for federated prognostics and health management in smart manufacturing. *Manufacturing Letters*, 24:136–139, 2020.

[13] Theodora S Brisimi, Ruidi Chen, Theofanie Mela, Alex Olshevsky, Ioannis Ch. Paschalidis, and Wei Shi. Federated learning of predictive models from federated electronic health records. *International Journal of Medical Informatics*, 112:59–67, 2018.

[14] Pradip Kumar Sharma, Jong Hyuk Park, and Kyungeun Cho. Blockchain and federated learning-based distributed computing defence framework for sustainable society. *Sustainable Cities and Society*, 59:102220, 2020.

[15] Eric M. Lightner and Steven E. Widergren. An orderly transition to a transformed electricity system. *IEEE Transactions on Smart Grid*, 1(1):3–10, 2010.

[16] CEN-CENELEC-ETSI Smart Grid Coordination Group. Smart grid reference architecture. `https://ec.europa.eu/energy/sites/ener/files/documents/xpert_group1_reference_architecture.pdf`, 11 2012. Acessed: 13-06-2022.

[17] C.W. Gellings. The concept of demand-side management for electric utilities. *Proceedings of the IEEE*, 73(10):1468–1470, 1985.

[18] Ekram(Editor) Hossain, Zhu(Editor) Han, and H Vincent(Editor) Poor. *Smart Grid Communications and Networking*. Cambridge University Press, 2012.

[19] D. Matheson, Chaoying Jing, and F. Monforte. Meter data management for the electricity market. In *2004 International Conference on Probabilistic Methods Applied to Power Systems*, pages 118–122, 2004.

[20] The GridWise Architecture Council. Gridwise transactive energy framework version 1.1. `https://gridwiseac.org/pdfs/pnnl_22946_gwac_te_framework_july_2019_v1_1.pdf`, 7 2019. Acessed: 25-06-2022.

[21] Tom M. Mitchell. *Machine Learning*. MGH, 1997.

[22] Phil Kim. *Matlab Deep Learning*. Apress, 2017.

[23] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[24] Seyedzadeh S. et al. Machine learning for estimation of building energy consumption and performance: a review. *Visualization in Engineering*, 6:5, 2018.

[25] James V Stone. *Artificial Intelligence Engines*. Sebtel Press, 1st edition, 2019.

[26] Jakub Konečný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence, 2016.

[27] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency, 2016.

[28] H. B. McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 2017.

[29] Tianjian Chen Han Yu Qiang Yang Yang Liu Yong Cheng Yan Kang. *Federated Learning*. Morgan & Claypool, 2019.

[30] Aurélie Foucquier, Sylvain Robert, Frédéric Suard, Louis Stéphan, and Arnaud Jay. State of the art in building modelling and energy performances prediction: A review. *Renewable and Sustainable Energy Reviews*, 23:272–288, 2013. Accessed: 27-06-2022.

[31] Deyslen Mariano-Hernández, Luis Hernández-Callejo, Felix Santos García, Oscar Duque-Perez, and Angel L. Zorita-Lamadrid. A review of energy consumption forecasting in smart buildings: Methods, input variables, forecasting horizon and metrics. *Applied Sciences*, 10(23), 2020.

[32] Tanveer Ahmad, Huanxin Chen, Yabin Guo, and Jiangyu Wang. A comprehensive overview on the data driven and large scale based approaches for forecasting of building energy demand: A review. *Energy and Buildings*, 165:301–320, 2018.

[33] Zhuo Yang, Douglas Eddy, Sundar Krishnamurty, Ian Grosse, Peter Denno, Yan Lu, and Paul Witherell. Investigating Grey-Box Modeling for Predictive Analytics in Smart Manufacturing. volume Volume 2B: 43rd Design Automation Conference of *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 08 2017. V02BT03A024.

[34] J.A. White and R. Reichmuth. Simplified method for predicting building energy consumption using average monthly temperatures. In *IECEC 96. Proceedings of the 31st Intersociety Energy Conversion Engineering Conference*, volume 3, pages 1834–1839 vol.3, 1996.

[35] Runming Yao and Koen Steemers. A method of formulating energy load profile for domestic buildings in the uk. *Energy and Buildings*, 37(6):663–671, 2005.

[36] Pierre Tittelein, Etienne Wurtz, and Gilbert Achard. Simspark platform evolution for low-energy building simulation. *International Scientific Journal for Alternative Energy and Ecology*, 2007.

[37] Ivan Korolija, Ljiljana Marjanovic-Halburd, Yi Zhang, and Victor I. Hanby. Uk office buildings archetypal model as methodological approach in development of regression models for predicting building energy consumption from heating and cooling demands. *Energy and Buildings*, 60:152–162, 2013.

[38] Pyonchan Ihm, Abderrezek Nemri, and Moncef Krarti. Estimation of lighting energy savings from daylighting. *Building and Environment*, 44:509–514, 2009.

[39] Che-Chiang Hsu and Chia-Yon Chen. Applications of improved grey prediction model for power demand forecasting. *Energy Conversion and Management*, 44(14):2241–2249, 2003.

[40] Aamer Abbas Shah, Khubab Ahmed, Xueshan Han, and Adil Saleem. A novel prediction error-based power forecasting scheme for real pv system using pvusa model: A grey box-based neural network approach. *IEEE Access*, 9:87196–87206, 2021.

[41] Tao Hong and Pu Wang. Artificial intelligence for load forecasting: History, illusions, and opportunities. *IEEE Power and Energy Magazine*, 20(3):14–23, 2022.

[42] Zulfiqar Ahmad Khan, Tanveer Hussain, Amin Ullah, Seungmin Rho, Miyoung Lee, and Sung Wook Baik. Towards efficient electricity forecasting in residential and commercial buildings: A novel hybrid cnn with a lstm-ae based framework. *Sensors*, 20(5), 2020.

[43] Hari Kiran, Hemanth Varma, Reddy Deeraj, and S Sridevi. Electricity consumption prediction using neural network. *Journal of Huazhong University of Science and Technology*, 50, 05 2021.

[44] Young Tae Chae, Raya Horesh, Youngdeok Hwang, and Young M. Lee. Artificial neural network model for forecasting sub-hourly electricity usage in commercial buildings. *Energy and Buildings*, 111:184–194, 2016.

[45] A. Azadeh, S.F. Ghaderi, and S. Sohrabkhani. Annual electricity consumption forecasting by neural network in high energy consuming industrial sectors. *Energy Conversion and Management*, 49(8):2272–2278, 2008.

[46] Reza Sabzehgar, Diba Zia Amirhosseini, and Mohammad Rasouli. Solar power forecast for a residential smart microgrid based on numerical weather predictions using artificial intelligence methods. *Journal of Building Engineering*, 32:101629, 2020.

[47] Rui Zhang, Minwei Feng, Wei Zhang, Siyuan Lu, and Fei Wang. Forecast of solar energy production - a deep learning approach. In *2018 IEEE International Conference on Big Knowledge (ICBK)*, pages 73–82, 2018.

[48] P. Kobylinski, M. Wierzbowski, and K. Piotrowski. High-resolution net load forecasting for micro-neighbourhoods with high penetration of renewable energy sources. *International Journal of Electrical Power & Energy Systems*, 117:105635, 2020.

[49] Marcel Antal, Liana Toderean, Tudor Cioara, and Ionut Anghel. Hybrid deep neural network model for multi-step energy prediction of prosumers. *Applied Sciences*, 12(11), 2022.

[50] S. Ehsan Razavi, Ali Arefi, Gerard Ledwich, Ghavameddin Nourbakhsh, David B Smith, and Manickam Minakshi. From load to net energy forecasting: Short-term residential forecasting for the blend of load and pv behind the meter. *IEEE Access*, 8:224343–224353, 2020.

[51] Junyang Li, Chaobo Zhang, Yang Zhao, Weikang Qiu, Qi Chen, and Xuejun Zhang. Federated learning-based short-term building energy consumption prediction method for solving the data silos problem. *Building Simulation*, 15:1145–1159, 2022.

[52] Afaf Taïk and Soumaya Cherkaoui. Electrical load forecasting using edge computing and federated learning. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pages 1–6, 2020.

[53] Christopher Briggs, Zhong Fan, and Peter Andras. Federated learning for short-term residential energy demand forecasting. *arXiv preprint arXiv:2105.13325*, 2021.

[54] Xinxin Zhou, Jingru Feng, Jian Wang, and Jianhong Pan. Privacy-preserving household load forecasting based on non-intrusive load monitoring: A federated deep learning approach. *arXiv e-prints*, pages arXiv–2206, 2022.

[55] Yi Wang, Imane Lahmam Bennani, Xiufeng Liu, Mingyang Sun, and Yao Zhou. Electricity consumer characteristics identification: A federated learning approach. *IEEE Transactions on Smart Grid*, 12(4):3637–3647, 2021.

[56] Yuris Mulya Saputra, Dinh Thai Hoang, Diep N Nguyen, Eryk Dutkiewicz, Markus Dominik Mueck, and Srikathyayani Srikanteswara. Energy demand prediction with federated learning for electric vehicle networks. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2019.

[57] Ye Lin Tun, Kyi Thar, Chu Myaet Thwal, and Choong Seon Hong. Federated learning based energy demand prediction with clustered aggregation. In *2021 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 164–167, 2021.

[58] Haizhou Liu, Xuan Zhang, Xinwei Shen, and Hongbin Sun. A federated learning framework for smart grids: Securing power traces in collaborative learning. *arXiv preprint arXiv:2103.11870*, 2021.

[59] Mohammad Saidur Rahman. Optimizing smart grid aggregators and measuring degree of privacy in a distributed trust based anonymous aggregation system. *arXiv preprint arXiv:2006.06070*, 2020.

[60] Peter C. Bruce and Andrew Bruce. *Practical statistics for data scientists : 50 essential concepts.* Sebastopol, CA: O'Reily, 2017.

[61] Eric J. H. Wilson, Andrew Parker, Anthony Fontanini, Elaina Present, Janet L. Reyna, Rajendra Adhikari, Carlo Bianchi, Christopher CaraDonna, Matthew Dahlhausen, Janghyun Kim, Amy LeBar, Lixi Liu, Marlena Praprost, Liang Zhang, Peter DeWitt, Noel Merket, Andrew Speake, Tianzhen Hong, Han Li, Natalie Mims Frick, Zhe Wang, Aileen Blair, Henry Horsey, David Roberts, Kim Trenbath, Oluwatobi Adekanye, Eric Bonnema, Rawad El Kontar, Jonathan Gonzalez, Scott Horowitz, Dalton Jones, Ralph T. Muehleisen, Siby Platthotam, Matthew Reynolds, Joseph Robertson, Kevin

Sayers, and Qu Li. End-use load profiles for the u.s. building stock: Methodology and results of model calibration, validation, and uncertainty quantification. 3 2022.

[62] The Weather Company and IBM. `https://www.wunderground.com/`. Accessed: 10-09-2022.

[63] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer, 2009.

[64] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[65] Rodrigo Salles, Jérôme Mendes, Rui Araújo, Carlos Melo, and Pedro Moura. Prediction of key variables in wastewater treatment plants using machine learning models. In *Proc. 2022 IEEE International Joint Conference on Neural Networks (IJCNN 2022), at the 2022 World Congress on Coomputational Intelligence (WCCI 2022)*, pages 1–9, Padova, Italy, July 18-23 2022. IEEE.

[66] Stephanie Holly, Thomas Hiessl, Safoura Rezapour Lakani, Daniel Schall, Clemens Heitzinger, and Jana Kemnitz. Evaluation of hyperparameter-optimization approaches in an industrial federated learning system. *CoRR*, abs/2110.08202, 2021.

[67] Pedro Moura, Uday Sriram, and Javad Mohammadi. Sharing mobile and stationary energy storage resources in transactive energy communities. pages 1–6, 06 2021.

[68] Pedro Moura, Greta K.W. Yu, and Javad Mohammadi. Multi-objective decision-making for transactive interactions in vehicle-to-building systems. In *2021 IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, pages 1–5, 2021.

[69] Javad Mohammadi, Gabriela Hug, and Soummya Kar. A fully distributed cooperative charging approach for plug-in electric vehicles. *IEEE Transactions on Smart Grid*, 9(4):3507–3518, 2018.

[70] Soummya Kar and Gabriela Hug. Distributed robust economic dispatch in power systems: A consensus + innovations approach. In *2012 IEEE Power and Energy Society General Meeting*, pages 1–8, 2012.

[71] Yuhan Du, Meiyi Li, Javad Mohammadi, Erik Blasch, Alex Aved, David Ferris, Philip Morrone, and Erika Ardiles Cruz. Learning assisted agent-based energy optimization: A reinforcement learning based consensus + innovations approach. In *2022 North American Power Symposium*, pages 1–6. IEEE, 2022.

Appendix A

Complete diagram of the proposed framework

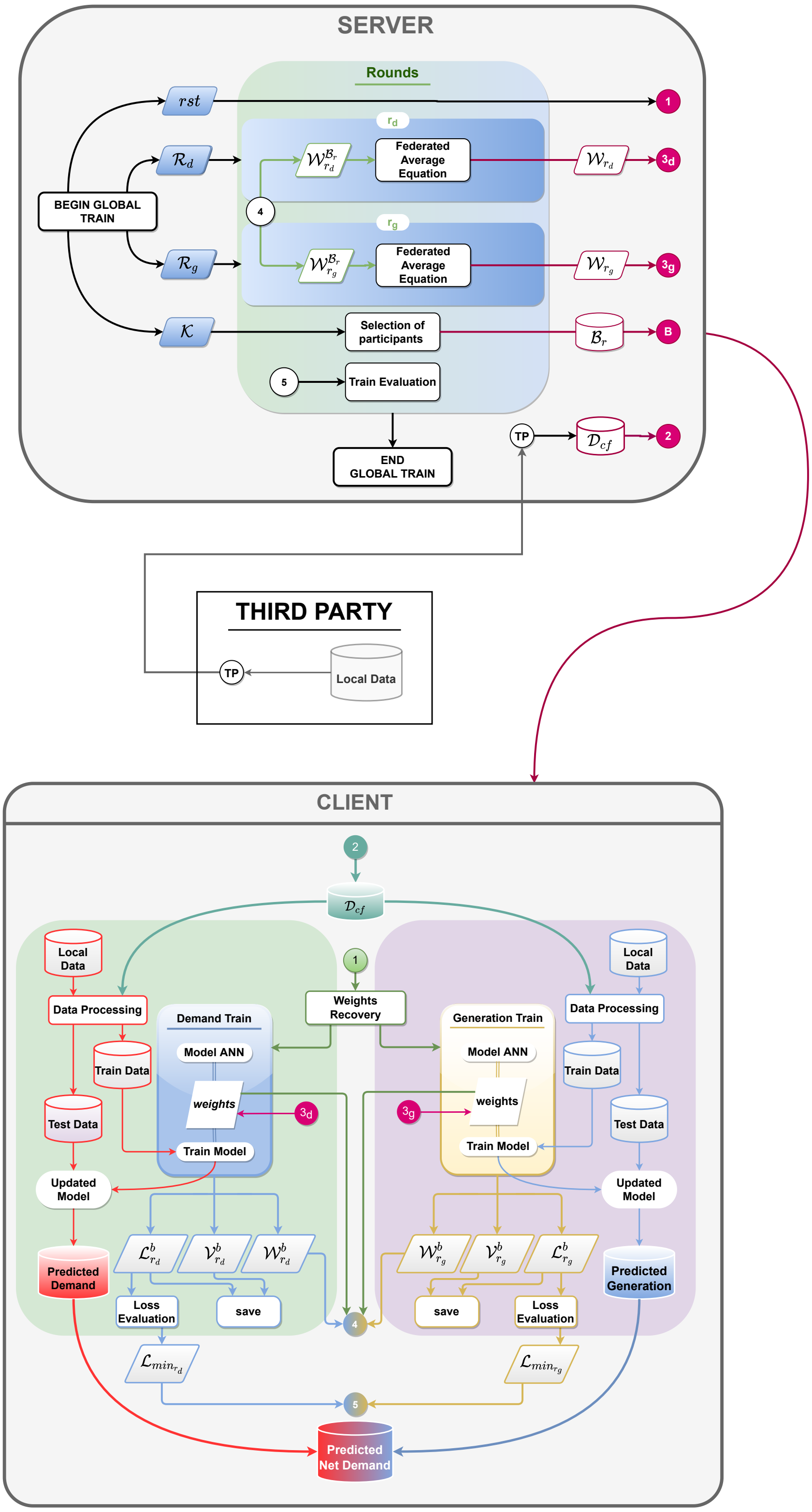
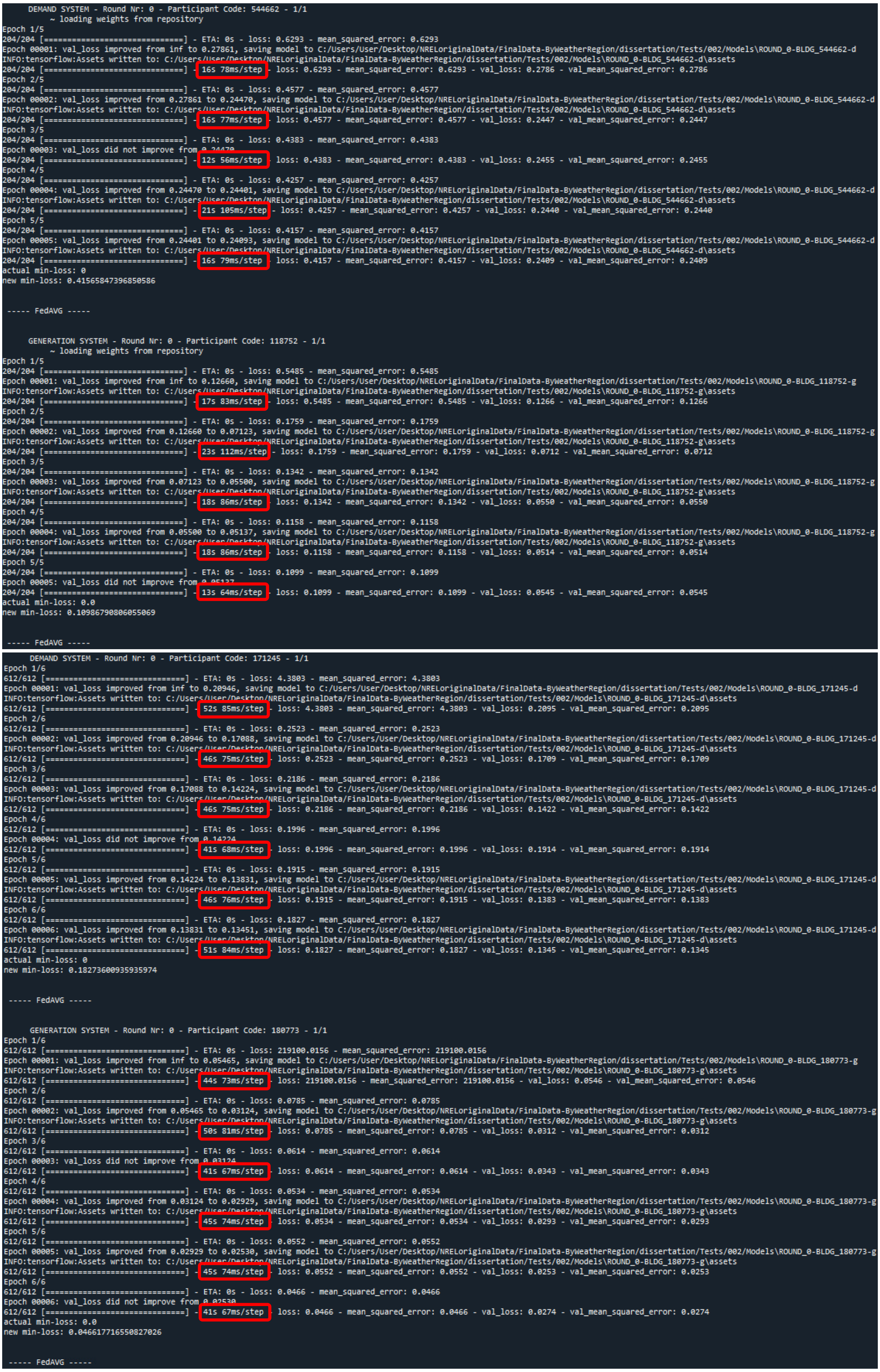


Figure A.1: Complete diagram of the proposed framework. Constituted by the server, the third-party, and the client.

Appendix B

Example of a train in the Federated Learning devoloped framework.



```
     DEMAND SYSTEM - Round Nr: 0 - Participant Code: 544662 - 1/1
         ~ loading weights from repository
Epoch 1/5
204/204 [==============================] - ETA: 0s - loss: 0.6293 - mean_squared_error: 0.6293
Epoch 00001: val_loss improved from inf to 0.27861, saving model to C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_544662-d
INFO:tensorflow:Assets written to: C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_544662-d\assets
204/204 [==============================] - 16s 78ms/step - loss: 0.6293 - mean_squared_error: 0.6293 - val_loss: 0.2786 - val_mean_squared_error: 0.2786
Epoch 2/5
204/204 [==============================] - ETA: 0s - loss: 0.4577 - mean_squared_error: 0.4577
Epoch 00002: val_loss improved from 0.27861 to 0.24470, saving model to C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_544662-d
INFO:tensorflow:Assets written to: C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_544662-d\assets
204/204 [==============================] - 16s 77ms/step - loss: 0.4577 - mean_squared_error: 0.4577 - val_loss: 0.2447 - val_mean_squared_error: 0.2447
Epoch 3/5
204/204 [==============================] - ETA: 0s - loss: 0.4383 - mean_squared_error: 0.4383
Epoch 00003: val_loss did not improve from 0.24470
204/204 [==============================] - 12s 56ms/step - loss: 0.4383 - mean_squared_error: 0.4383 - val_loss: 0.2455 - val_mean_squared_error: 0.2455
Epoch 4/5
204/204 [==============================] - ETA: 0s - loss: 0.4257 - mean_squared_error: 0.4257
Epoch 00004: val_loss improved from 0.24470 to 0.24401, saving model to C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_544662-d
INFO:tensorflow:Assets written to: C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_544662-d\assets
204/204 [==============================] - 21s 105ms/step - loss: 0.4257 - mean_squared_error: 0.4257 - val_loss: 0.2440 - val_mean_squared_error: 0.2440
Epoch 5/5
204/204 [==============================] - ETA: 0s - loss: 0.4157 - mean_squared_error: 0.4157
Epoch 00005: val_loss improved from 0.24401 to 0.24093, saving model to C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_544662-d
INFO:tensorflow:Assets written to: C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_544662-d\assets
204/204 [==============================] - 16s 79ms/step - loss: 0.4157 - mean_squared_error: 0.4157 - val_loss: 0.2409 - val_mean_squared_error: 0.2409
actual min-loss: 0
new min-loss: 0.41565847396850586


 ----- FedAVG -----


     GENERATION SYSTEM - Round Nr: 0 - Participant Code: 118752 - 1/1
         ~ loading weights from repository
Epoch 1/5
204/204 [==============================] - ETA: 0s - loss: 0.5485 - mean_squared_error: 0.5485
Epoch 00001: val_loss improved from inf to 0.12660, saving model to C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_118752-g
INFO:tensorflow:Assets written to: C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_118752-g\assets
204/204 [==============================] - 17s 83ms/step - loss: 0.5485 - mean_squared_error: 0.5485 - val_loss: 0.1266 - val_mean_squared_error: 0.1266
Epoch 2/5
204/204 [==============================] - ETA: 0s - loss: 0.1759 - mean_squared_error: 0.1759
Epoch 00002: val_loss improved from 0.12660 to 0.07123, saving model to C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_118752-g
INFO:tensorflow:Assets written to: C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_118752-g\assets
204/204 [==============================] - 23s 112ms/step - loss: 0.1759 - mean_squared_error: 0.1759 - val_loss: 0.0712 - val_mean_squared_error: 0.0712
Epoch 3/5
204/204 [==============================] - ETA: 0s - loss: 0.1342 - mean_squared_error: 0.1342
Epoch 00003: val_loss improved from 0.07123 to 0.05500, saving model to C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_118752-g
INFO:tensorflow:Assets written to: C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_118752-g\assets
204/204 [==============================] - 18s 86ms/step - loss: 0.1342 - mean_squared_error: 0.1342 - val_loss: 0.0550 - val_mean_squared_error: 0.0550
Epoch 4/5
204/204 [==============================] - ETA: 0s - loss: 0.1158 - mean_squared_error: 0.1158
Epoch 00004: val_loss improved from 0.05500 to 0.05137, saving model to C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_118752-g
INFO:tensorflow:Assets written to: C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_118752-g\assets
204/204 [==============================] - 18s 86ms/step - loss: 0.1158 - mean_squared_error: 0.1158 - val_loss: 0.0514 - val_mean_squared_error: 0.0514
Epoch 5/5
204/204 [==============================] - ETA: 0s - loss: 0.1099 - mean_squared_error: 0.1099
Epoch 00005: val_loss did not improve from 0.05137
204/204 [==============================] - 13s 64ms/step - loss: 0.1099 - mean_squared_error: 0.1099 - val_loss: 0.0545 - val_mean_squared_error: 0.0545
actual min-loss: 0.0
new min-loss: 0.10986790806055069


 ----- FedAVG -----
```

```
      DEMAND SYSTEM - Round Nr: 0 - Participant Code: 171245 - 1/1
Epoch 1/6
612/612 [==============================] - ETA: 0s - loss: 4.3803 - mean_squared_error: 4.3803
Epoch 00001: val_loss improved from inf to 0.20946, saving model to C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_171245-d
INFO:tensorflow:Assets written to: C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_171245-d\assets
612/612 [==============================] - 52s 85ms/step - loss: 4.3803 - mean_squared_error: 4.3803 - val_loss: 0.2095 - val_mean_squared_error: 0.2095
Epoch 2/6
612/612 [==============================] - ETA: 0s - loss: 0.2523 - mean_squared_error: 0.2523
Epoch 00002: val_loss improved from 0.20946 to 0.17088, saving model to C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_171245-d
INFO:tensorflow:Assets written to: C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_171245-d\assets
612/612 [==============================] - 46s 75ms/step - loss: 0.2523 - mean_squared_error: 0.2523 - val_loss: 0.1709 - val_mean_squared_error: 0.1709
Epoch 3/6
612/612 [==============================] - ETA: 0s - loss: 0.2186 - mean_squared_error: 0.2186
Epoch 00003: val_loss improved from 0.17088 to 0.14224, saving model to C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_171245-d
INFO:tensorflow:Assets written to: C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_171245-d\assets
612/612 [==============================] - 46s 75ms/step - loss: 0.2186 - mean_squared_error: 0.2186 - val_loss: 0.1422 - val_mean_squared_error: 0.1422
Epoch 4/6
612/612 [==============================] - ETA: 0s - loss: 0.1996 - mean_squared_error: 0.1996
Epoch 00004: val_loss did not improve from 0.14224
612/612 [==============================] - 41s 68ms/step - loss: 0.1996 - mean_squared_error: 0.1996 - val_loss: 0.1914 - val_mean_squared_error: 0.1914
Epoch 5/6
612/612 [==============================] - ETA: 0s - loss: 0.1915 - mean_squared_error: 0.1915
Epoch 00005: val_loss improved from 0.14224 to 0.13831, saving model to C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_171245-d
INFO:tensorflow:Assets written to: C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_171245-d\assets
612/612 [==============================] - 46s 76ms/step - loss: 0.1915 - mean_squared_error: 0.1915 - val_loss: 0.1383 - val_mean_squared_error: 0.1383
Epoch 6/6
612/612 [==============================] - ETA: 0s - loss: 0.1827 - mean_squared_error: 0.1827
Epoch 00006: val_loss improved from 0.13831 to 0.13451, saving model to C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_171245-d
INFO:tensorflow:Assets written to: C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_171245-d\assets
612/612 [==============================] - 51s 84ms/step - loss: 0.1827 - mean_squared_error: 0.1827 - val_loss: 0.1345 - val_mean_squared_error: 0.1345
actual min-loss: 0
new min-loss: 0.18273600935935974


 ----- FedAVG -----


      GENERATION SYSTEM - Round Nr: 0 - Participant Code: 180773 - 1/1
Epoch 1/6
612/612 [==============================] - ETA: 0s - loss: 219100.0156 - mean_squared_error: 219100.0156
Epoch 00001: val_loss improved from inf to 0.05465, saving model to C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_180773-g
INFO:tensorflow:Assets written to: C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_180773-g\assets
612/612 [==============================] - 44s 73ms/step - loss: 219100.0156 - mean_squared_error: 219100.0156 - val_loss: 0.0546 - val_mean_squared_error: 0.0546
Epoch 2/6
612/612 [==============================] - ETA: 0s - loss: 0.0785 - mean_squared_error: 0.0785
Epoch 00002: val_loss improved from 0.05465 to 0.03124, saving model to C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_180773-g
INFO:tensorflow:Assets written to: C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_180773-g\assets
612/612 [==============================] - 50s 81ms/step - loss: 0.0785 - mean_squared_error: 0.0785 - val_loss: 0.0312 - val_mean_squared_error: 0.0312
Epoch 3/6
612/612 [==============================] - ETA: 0s - loss: 0.0614 - mean_squared_error: 0.0614
Epoch 00003: val_loss did not improve from 0.03124
612/612 [==============================] - 41s 67ms/step - loss: 0.0614 - mean_squared_error: 0.0614 - val_loss: 0.0343 - val_mean_squared_error: 0.0343
Epoch 4/6
612/612 [==============================] - ETA: 0s - loss: 0.0534 - mean_squared_error: 0.0534
Epoch 00004: val_loss improved from 0.03124 to 0.02929, saving model to C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_180773-g
INFO:tensorflow:Assets written to: C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_180773-g\assets
612/612 [==============================] - 45s 74ms/step - loss: 0.0534 - mean_squared_error: 0.0534 - val_loss: 0.0293 - val_mean_squared_error: 0.0293
Epoch 5/6
612/612 [==============================] - ETA: 0s - loss: 0.0552 - mean_squared_error: 0.0552
Epoch 00005: val_loss improved from 0.02929 to 0.02530, saving model to C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_180773-g
INFO:tensorflow:Assets written to: C:/Users/User/Desktop/NRELoriginalData/FinalData-ByWeatherRegion/dissertation/Tests/002/Models\ROUND_0-BLDG_180773-g\assets
612/612 [==============================] - 45s 74ms/step - loss: 0.0552 - mean_squared_error: 0.0552 - val_loss: 0.0253 - val_mean_squared_error: 0.0253
Epoch 6/6
612/612 [==============================] - ETA: 0s - loss: 0.0466 - mean_squared_error: 0.0466
Epoch 00006: val_loss did not improve from 0.02530
612/612 [==============================] - 41s 67ms/step - loss: 0.0466 - mean_squared_error: 0.0466 - val_loss: 0.0274 - val_mean_squared_error: 0.0274
actual min-loss: 0.0
new min-loss: 0.046617716550827026


 ----- FedAVG -----
```

Figure B.1: Comparison between time performances with different parameters.

Appendix C

Paper accepted for publication in the conference IEEE ISGT 2022

Federated Learning Enabled Prediction of Energy Consumption in Transactive Energy Communities

Nuno Mendes
ISR, Electrical and Computer Eng.
University of Coimbra
Coimbra, Portugal
nuno.mendes@isr.uc.pt

Pedro Moura
ISR, Electrical and Computer Eng.
University of Coimbra
Coimbra, Portugal
pmoura@isr.uc.pt

Jérôme Mendes
ISR, Electrical and Computer Eng.
University of Coimbra
Coimbra, Portugal
jermendes@isr.uc.pt

Rodrigo Salles
ISR, Electrical and Computer Eng.
University of Coimbra
Coimbra, Portugal
rodrigo.salles@isr.uc.pt

Javad Mohammadi
Cockrell School of Engineering
University of Texas at Austin
Austin, USA
javadm@utexas.edu

***Abstract*—The prediction of the net electricity demand is crucial to the management and optimization of transactive energy communities. Such prediction usually relies on net-demand information, but each building can have additional information, such as separated generation and demand profiles, weather, or occupancy data. Such information is not only relevant for the net-demand prediction of each building, but also to other buildings with the same type of use. However, buildings avoid sharing such information due to privacy concerns. This paper proposes a novel federated learning framework for predicting building temporal net-demand in transactive energy communities. The proposed approach leverages centralized oversight of a central agent (aggregator) to inform distributed collaboration among each client (buildings), which are willing to collaborate to improve their prediction accuracy. The proposed approach was tested using a dataset collected from several buildings from a University campus (from the University of Coimbra in Portugal), predicting the electricity demand, and then using the local generation data to evaluate the net-demand, in the community of buildings.**

***Index Terms*—Federated Learning, Distributed Computation, Transactive Energy, Energy Consumption, Energy Community.**

I. Introduction

A. Motivation

The increasing penetration of renewable generation has been leading to new challenges in electrical grid management, due to the intermittency and variability of the generation sources. In the context of buildings and communities, the use of solar photovoltaic (PV) generation is strongly increasing. However, typically, the profiles of PV generation and the demand in buildings can have a strong mismatch, and new technologies are needed to ensure the required flexibility for the coordination between the available generation and demand. Flexibility options, such as battery energy storage and demand response, including the management of charging of electrical vehicles (EV) have been integrated into buildings. The flexibility technologies can be used to ensure the optimization of the self-consumption and costs at the building and community levels [1]. However, to ensure such an objective, a traditional approach with the Distribution System Operator (DSO) predicting the load at the substation level and adapting the grid resources is not enough. Therefore, to ensure effective management of resources the prediction of the net-demand in each individual building is essential [2].

These issues are more critical in future Transactive Energy (TE) systems that are defined as economic and control mechanisms for managing consumption and generation through enabling end-use energy trading [3]. TE systems enable scalable operation and optimization of heterogeneous producers and consumers (as known as prosumers) assets in communities, and to ensure such an objective the prediction of net-demand is critical. In a community of buildings, managed by an aggregator, the buildings' interests are aligned, hence, they are willing to collaborate to increase the prediction accuracy in order to achieve better resource management and energy cost reduction.

However, most buildings are not interested in sharing their data with other buildings, since such data can reveal sensitive commercial data such as the number of occupants or strategies for participating in the TE market. To solve such an issue, this paper presents a novel federated learning (FL) model for predicting the net demand of several buildings in transactive energy communities. In FL, multiple entities collaborate in solving a machine learning problem, under the coordination of an aggregator or in a peer-to-peer scheme, being the raw data of each client locally stored and not exchanged or transferred [4]. This paper proposes an approach with the centralized oversight of a central agent (aggregator) to inform distributed collaboration among each client agent (buildings). With such an approach, clients maintain their private information as confidential, but simultaneously contributed not only to improving the prediction in their building, but also in other buildings of the community.

This research was supported by FCT through the project ML@GridEdge (UTAP-EXPL/CA/0065/2021) and by the ERDF and national funds through the project EVAICharge (CENTRO-01-0247-FEDER-047196).

978-1-6654-8032-1/22/$31.00 ©2022 IEEE

B. Related Works

Energy prediction models have a critical role in energy policy and energy management in buildings. For demand prediction, the prevailing techniques used in large-scale building applications, include white-box, black-box, and grey-box based methods [5]. Recently, the booming development of deep learning techniques brings more exactitude to the "black-box" approaches, bringing promising alternatives to conventional data-driven approaches [6]. For instance, [7] proposes deep learning-based techniques for day-ahead multi-step load forecasting in commercial buildings.

However, with the integration of renewable generation in buildings, load forecasting is not enough, being fundamental to take into account the generation, by predicting the net-load. The net-load is often predicted for communities of residential buildings and in [8] an Artificial Neural Network based model was designed to predict short-term micro-scale residential net-load profiles. The most accurate solutions are mainly found for the prediction of aggregated loads at the substation level, predicting the load for the entire community. However, for the management of communities using flexibility resources at the building level, the prediction of energy consumption at every building level is required.

For instance, in [9] a hybrid convolutional neural network with a long short-term memory autoencoder model was used for future energy prediction in residential and commercial buildings. However, such works are focused on individual buildings, without taking advantage of the collaborative opportunities between buildings with similar characteristics each one with private data resources that can be important to the other buildings. Therefore, such methods are only based on the net-metering data, therefore losing the differentiated impact on demand and on the generation or they just assume that the individual generation and demand data can be available without taking into account that most buildings do not intend to publicly share such data. Such methods also do not take into account the availability of other private information that can be available in some buildings (e.g. weather, comfort, or occupancy data).

The use of private data in collaborative machine learning problems on decentralized data where privacy is paramount has been ensured using FL [4]. Such approaches have been used primarily in mobile and edge device applications, but the use of FL in power systems is limited. In [10] FL is used to identify the socio-demographic characteristics of electricity consumers, in [11] FL is used for the prediction of energy demand in EV Networks, and in [12] a clustered aggregation is implemented for the prediction of electricity demand. However, FL has not yet been used to enhance the prediction of net demand in communities of buildings.

C. Contribution

The main contribution of this work includes introducing a federated learning framework to enhance the prediction accuracy of buildings' net demand. The proposed solution lends itself well to TE systems where buildings, aggregators, and DSO are decision-making entities. It is assumed that DSO shares the net-demand metering data with aggregators. The clients will collaborate to fine-tune their models by executing the training program using their data while keeping their data private. The aggregator will then update the model weights to improve the prediction of all buildings in the community. The implemented approach is based on a Horizontal Federated Learning (HFL) model. Still, it introduces as a novelty the use of a third party to provide information in common to all buildings (e.g., weather data). While the proposed approach is presented in the context of net-demand prediction, it is envisioned that it can be extended to transactive energy communities' coordination problems.

D. Paper Organization

The remainder of the paper is structured as follows. The Federated Learning framework is presented in Section II. The data and scenarios are described in Section III, and the achieved results are presented in Section IV. Finally, the main conclusions are highlighted in Section V.

II. Federated Learning Framework

This paper considers a TE environment that includes buildings, aggregators, DSO, and third-party data providers, as presented in Fig. 1. In this setup, the DSO oversees the power flow between each building and the grid and has access to all buildings' historical net-demand metering data. Each building can sign up with one aggregator, and with the building's permission, the aggregator will get access to net-demand metering of that building collected by the DSO. Additionally, each building may access additional information, such as separated generation and demand profiles, sub-metering, and occupancy data. Other relevant data in common for all buildings, such as weather data, can be obtained from third-party data providers. Due to privacy concerns, the building does not intend to share its data with the aggregator. However, such information is relevant for the building owning the data and other buildings. In fact, there is a high correlation between the PV generation in buildings in the same community, and the demand correlation between the same type of buildings can also be high.

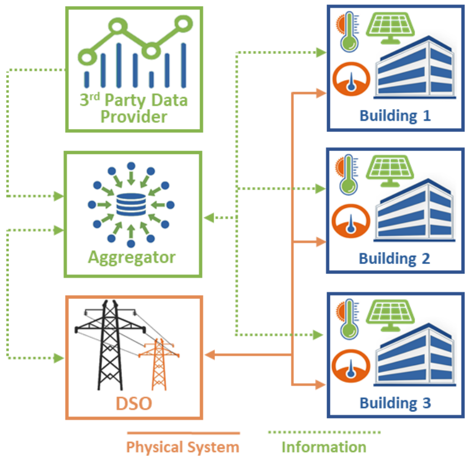


Fig. 1: Power and data flows in the community.

A separate prediction for generation and demand increases the reliability of the net-demand prediction. In the same community in a given location, there is almost a perfect correlation between the PV generation variation in the several buildings (the only factors affecting it can be slightly different slops and orientation of panels or an eventual shading in some buildings). In the case of demand, even considering the same type of buildings, the correlation is not so high, since it is affected by factors with a different impact on the several buildings, such as temperature, schedule of services and users. Considering directly the net-demand would attenuate such correlations. For instance, on a sunny day with high temperatures, the PV generation and demand will be high, with the generation increase attenuating the demand increase, but with a differentiated impact in different buildings.

In order to integrate the proposed system into the setup present in Fig. 1, it is necessary a model that ensures global training with specific characteristics, namely: all the buildings will have the same features, the structure of the model will be the same for all buildings, and it is necessary to have a coordinator, the aggregator. To ensure such characteristics, this approach is based on an HFL model [13] with a client-server architecture. FL is an edge collaborative machine learning technique that allows the training of an algorithm that already has been trained in a certain device. In such a context, HFL is a model that is used when the participants have the same features with different samples, to train an equal model. In terms of learning, this will improve the models, since they will be trained in multiple different situations.

The selected architecture fits on the global structure presented in Fig. 1, where the server will be the aggregator, who will be responsible for the system coordination. The clients will be the buildings $\mathcal{B}$ and each building has its own system for training its personal predicting model. Additionally, a third party was added to the system, which will be responsible for providing weather data. This approach intends to reduce the number of entities connected to the buildings, increasing their protection. The proposed Federated Average Algorithm was adapted from [14], and the pseudocode associated with the aggregator is presented in Algorithm 1.

Algorithm 1 Aggregator Algorithm

```
• Input: $\mathcal{K}$; $\mathcal{R}$; $\mathcal{D}_{cf}$.
 1: for $r = 1, 2, \ldots, \mathcal{R}$ do
 2:     $\mathcal{B}_r = \{b_1, b_2, \ldots b_k\} \in \mathcal{B}$
 3:     if $\mathcal{F}_{gt}$ then
 4:         $\mathcal{W}_a \leftarrow \mathcal{W}_0$
 5:     else
 6:         $\mathcal{W}_a \leftarrow \frac{1}{\mathcal{K}} \sum_{b=1}^{\mathcal{B}_r} \mathcal{W}_r^b$
 7:     end if
 8:     for $b = 1, 2, \ldots, \mathcal{B}_r$ in parallel do
 9:         $\mathcal{W}_{r+1}^b$, $\mathcal{V}_{r+1}^b$, $\mathcal{L}_{r+1}^b \leftarrow$BuildingUpdate($\mathcal{W}_a$ , $\mathcal{D}_{cf}$)
10:         $\mathcal{W}_r^b \leftarrow \mathcal{W}_{r+1}^b$
11:         $\mathcal{V}^\frown \mathcal{V}_{r+1}^b$
12:         $\mathcal{L}^\frown \mathcal{L}_{r+1}^b$
13:     end for
14: end for
```

This architecture ensures global training by averaging the weights $\mathcal{W}_r^b$ of the predicted model in each participant. This assurance is preserved by interactions between the participants and the respective aggregators. The aggregator implements $\mathcal{R}$ rounds iteratively. For each round, $\mathcal{K}$ buildings are selected from $\mathcal{B}$, know as participants $\mathcal{B}_r$, where $\mathcal{K} < n(\mathcal{B})$. This is done to require less computation capacity, and since this selection is made randomly there is a high probability that all buildings being selected for at least one round in the global training. There is only one interaction where the averaging is not implemented, which is the first time that the system initializes. In this specific case, random weights are sent to all participants. $\mathcal{F}_{gt}$ indicates if this situation occurs or not.

The aggregator received from a third-party a dataset with common features $\mathcal{D}_{cf}$ to the buildings, namely the dataset with the weather information. The objective of this approach is to reduce the number of entities connected to the final user, as well as the redundancy between users. In [13], HFL is introduced with all data allocated on the client's system. However, in the implemented setup the weather data will be sent from the aggregator, and when the aggregator communicates the updated weights $\mathcal{W}_a$ to the participants, the weather data $\mathcal{D}_{cf}$ is also sent.

In [14], the calculation of the updated weights is done considering the number of samples. Therefore, each participant influences the average depending on the ratio between their quantity of samples and the total number of samples in all aggregated participants. There is then the assumption that the larger the dataset, the greater is their importance in this calculation. However, in this work, all the participants have the same "importance" since the averaging is independent of the number of samples. Line 6 of Algorithm 1 shows the selected function to implement the averaging of the weights.

The next step is to execute local training (i.e., (2)) in parallel for all the participants. Each building b returns its final weights $\mathcal{W}_{r+1}^b$, loss $\mathcal{L}_{r+1}^b$, and validation loss $\mathcal{V}_{r+1}^b$ associated with its local training. $\mathcal{W}_r^b$ gets the values of all weights received from all participants in each round, $\mathcal{L}$ and $\mathcal{V}$ concatenates all $\mathcal{L}_{r+1}^b$ and $\mathcal{V}_{r+1}^b$, respectively, for all rounds. At the end of the global train, these variables contain all values of loss and validation loss from all participants in all rounds.

Algorithm 2 Building Update Algorithm

```
• Executed by: all buildings seleted, the participants $\mathcal{B}_r$,
  Algorithm 1.
• Input: $\mathcal{W}_a$ , $\mathcal{D}_{cf}$.
1: $\mathcal{D}_b \leftarrow \mathcal{D}_{cf}{}^\frown \mathcal{D}_{pd}$
2: $\mathcal{W}_{r+1}^b$,$\mathcal{V}_{r+1}^b$,$\mathcal{L}_{r+1}^b \leftarrow$ ParticipantUpdate($\mathcal{W}_a$ , $\mathcal{D}_b$)
```

The diagram of the system associated with the buildings is presented in Fig. 2. This structure presents the flow of the proposed Federated Average Algorithm and represents the algorithm for predicting the net-demand. Line 2 of Algorithm 2, receives the weights ($\mathcal{W}_a$) and these are sent directly to the Artificial Neural Network (ANN) model, which will be responsible for predicting the demand. The ANN model proposed is presented in Table 1, with a Long Short-Term Memory (LSTM) architecture.

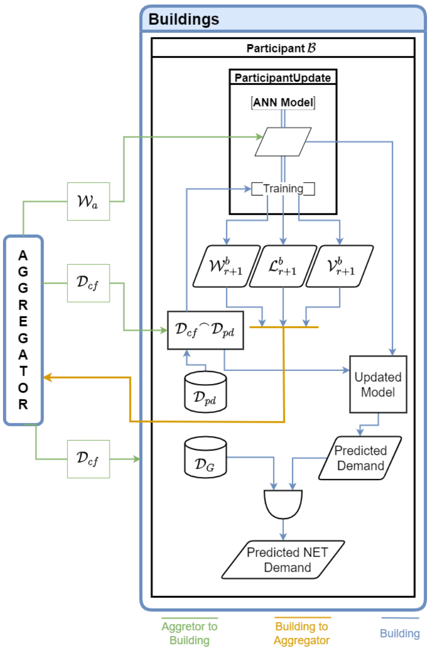


Fig. 2: Structure implemented in the buildings.

TABLE I: Proposed LSTM architecture.

Layer	Type	LSTM cells	Activation Function
LSTM	Input	64	Relu
LSTM	Hidden	32	Relu
Dropout(0.2)	Hidden	-	-
LSTM	Hidden	32	Relu
Dropout(0.2)	Hidden	-	-
Dense	Output	-	-

The datasets of the common features ($\mathcal{D}_{cf}$) are concatenated to the private dataset ($\mathcal{D}_{pd}$), which is represented in Fig. 2 as $\mathcal{D}_{cf} \frown \mathcal{D}_{pd}$, and in Algorithm 2 on line 1. Both datasets are constituted by time series, and to proceed with the aggregation an algorithm that implements the time matching between both was employed. The resulting dataset is used to train the model, and the final loss $\mathcal{L}^b_{r+1}$, validation loss $\mathcal{V}^b_{r+1}$, and weights $\mathcal{W}^b_{r+1}$ returned by the model are sent back to the aggregator. In order to calculate the net-demand, a dataset $\mathcal{D}_G$ with the predicted generation was used.

The parameters associated with the structure implementation were the number of hidden layers $\mathcal{N}_l$, the number of LSTM cells for the input layer $\mathcal{N}_{ci}$, number of LSTM cells for the hidden layers $\mathcal{N}_{ch}$, number of batch size $\mathcal{B}_s$, number of training cycles (epochs) $\mathcal{E}$, number of past observations $\mathcal{N}_{op}$, and number of forecast observations $\mathcal{N}_{of}$.

III. Data and Scenarios

The used data belongs to the University campus. In particular, six buildings from the University of Coimbra were selected. This dataset is identified by the department IDs: DCE (Department of Civil Engineering), DChE (Department of Chemical Engineering), DECE (Department of Electrical and Computer Engineering), DES (Department of Earth Sciences), DIE (Department of Informatics Engineering), and DME (Department of Mechanical Engineering).

The private dataset for all buildings needs to match in terms of features. The common feature set includes: (i) each building's net-demand, and (ii) on-site PV generation data. This information allows for calculating electricity demand. However, since the buildings belong to a University, there is a clear seasonality of activities and therefore other included features were the academic calendar (classes, exams and vacations) and the days of the week (weekends and weekdays).

The buildings do not directly collect any weather data, therefore requiring the participation of a third party to provide such data. Additionally, due to the physical proximity of the buildings, only one weather dataset is needed, as defined in the previous section as $\mathcal{D}_{cf}$. The "Wunder Ground" website served as a third-party temperature, humidity, and solar radiation data provider.

The net-demand and PV generation data have a periodicity of 15 min, and therefore one year corresponds to a total of 35040 samples per building. The academic calendar and days of the week are categorical data. In order to convert this data into numeric values, a process known as One Hot Encoding [15] was implemented. This process consists of splitting the categories associated with variables and each sample. Put differently; correspondent sub-categories are identified with one while others are assigned zero. In the simulations, three sub-categories were considered for the academic calendars: classes, exams, and vacations. Weekends and weekdays are separated as well.

The proposed FL framework was validated by splitting the dataset of each building into two parts: 74% dedicated to training (used to train the local models) and the rest to testing (utilized to test the obtained models after the global training). The months of January, August, and October were selected for testing the models in different seasonal conditions. The remaining months were allocated to the training dataset. The model uses 25% of the training dataset for validation in the local training.

IV. Results

The following parameters are used to test the proposed structure: $\mathcal{R} = 60$, $\mathcal{K} = 3$, $\mathcal{N}_l = 2$, $\mathcal{N}_{ci} = 64$, $\mathcal{N}_{ch} = 32$, $\mathcal{B}_s = 32$, $\mathcal{E} = 6$, $\mathcal{N}_{op} = 96$, and $\mathcal{N}_{of} = 1$. It should be noted that instead of defining a learning rate value, the Adam optimizer was used, which inherently uses an adaptive learning rate method.

The local models start converging after round 33, imposed by the stopping criteria of the global model. Table II presents the obtained results: the θ value is the number of rounds in which each building participated, $\mathcal{L}_{training}$ is the lower loss value, r is the corresponding round number of the global training. Also, p is the number of local rounds before stopping the global model at round r. The Root Mean Square Error (RMSE) of the model corresponding to the round r over the test dataset was calculated.

As can be inferred from the results of Table II, the proposed FL setup is implemented between 13 and 21 local trains in each building. It can also be observed that the round with

TABLE II: Achieved score results.

Building ID	θ	Results on Training			Results on Test
		$\mathcal{L}_{training}$	p	r	RMSE (r)
DCE	14	0.06878	12	27	5.39172
DChE	17	0.03807	15	27	4.19676
DECE	17	0.02663	8	14	3.25616
DES	13	0.03695	10	23	4.85933
DIE	21	0.03076	20	31	3.02867
DME	17	0.04138	16	31	5.89367

lower loss values occurs before the last local train. Taking the building DChE as an example, Fig 3 presents the loss value as of the global training convergences throughout rounds. Mean Squared Error (MSE) was used as the evaluation metric to obtain the discussed results. The loss graph showcases the decrease throughout the rounds and demonstrates learning over the course of training. Overlapping this graph with the green vertical lines depicts that the loss presents higher values when the Fevederated Average calculus is implemented.

Fig. 4 presents the predicted and real values for the test dataset for the building DChE, to show an example of the net-demand prediction achieved by the model where the lower loss value on the training was achieved. Comparing the obtained results over different months reveals the adaptability of the proposed model with respect to the seasonal variations.

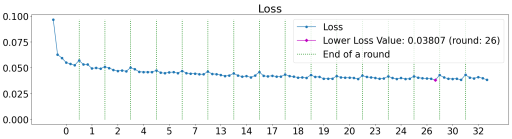


Fig. 3: MSE values obtained throughout the global training in the DChE building.

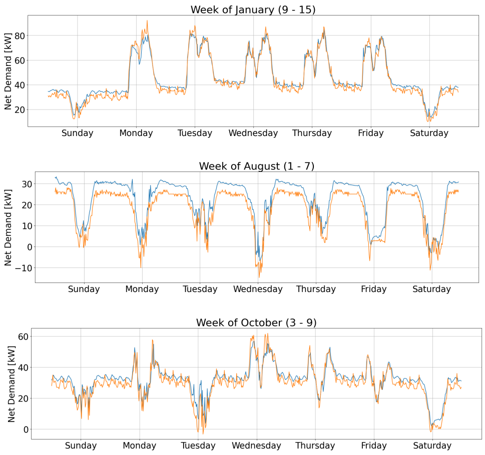


Fig. 4: Net-demand in the building DChE in the global round $r = 27$ (real values in blue and prediction in orange).

V. Conclusions

This paper proposes a novel approach for predicting net-demand in transactive energy communities based on Federated Learning. The developed structure allows the integration of third-party data providers, coordination by an aggregator, and collaborative learning among the buildings without sharing private data. The results present a high level of accuracy and adaptability to different situations, for example, seasonal variations.

In future work, the objective is to (i) develop a local model to predict on-site generation, (ii) devise a model for the aggregator to predict the weather accurately, and (iii) enhance local information processing units to select the best models for each global training.

References

[1] P. Moura, U. Sriram, and J. Mohammadi, "Sharing mobile and stationary energy storage resources in transactive energy communities," in *2021 IEEE Madrid PowerTech*, 2021, pp. 1–6.

[2] W. Tushar, T. K. Saha, C. Yuen, D. Smith, and H. V. Poor, "Peer-to-peer trading in electricity networks: An overview," *IEEE Transactions on Smart Grid*, pp. 1–1, 2020.

[3] P. Siano, G. De Marco, A. Rolán, and V. Loia, "A survey and evaluation of the potentials of distributed ledger technology for peer-to-peer transactive energy exchanges in local energy markets," *IEEE Systems Journal*, vol. 13, no. 3, pp. 3454–3466, 2019.

[4] P. Kairouz and H. B. M. et al., "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[5] W. Stainsby, D. Zimmerle, and G. P. Duggan, "A method to estimate residential pv generation from net-metered load data and system install date," *Applied Energy*, vol. 267, p. 114895, 2020.

[6] D. Mariano-Hernández, L. Hernández-Callejo, F. S. García, O. Duque-Perez, and A. L. Zorita-Lamadrid, "A review of energy consumption forecasting in smart buildings: Methods, input variables, forecasting horizon and metrics," *Applied Sciences*, vol. 10, no. 23, 2020.

[7] M. Cai, M. Pipattanasomporn, and S. Rahman, "Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques," *Applied Energy*, vol. 236, pp. 1078 – 1088, 2019.

[8] P. Kobylinski, M. Wierzbowski, and K. Piotrowski, "High-resolution net load forecasting for micro-neighbourhoods with high penetration of renewable energy sources," *International Journal of Electrical Power & Energy Systems*, vol. 117, p. 105635, 2020.

[9] Z. A. Khan, T. Hussain, A. Ullah, S. Rho, M. Lee, and S. W. Baik, "Towards efficient electricity forecasting in residential and commercial buildings: A novel hybrid cnn with a lstm-ae based framework," *Sensors*, vol. 20, no. 5, 2020.

[10] Y. Wang, I. L. Bennani, X. Liu, M. Sun, and Y. Zhou, "Electricity consumer characteristics identification: A federated learning approach," *IEEE Transactions on Smart Grid*, vol. 12, no. 4, pp. 3637–3647, 2021.

[11] Y. M. Saputra, D. T. Hoang, D. N. Nguyen, E. Dutkiewicz, M. D. Mueck, and S. Srikanteswara, "Energy demand prediction with federated learning for electric vehicle networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[12] Y. L. Tun, K. Thar, C. M. Thwal, and C. S. Hong, "Federated learning based energy demand prediction with clustered aggregation," in *2021 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2021, pp. 164–167.

[13] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, *Federated Learning*. Morgan & Claypool, 2019, vol. 13.

[14] 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.

[15] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.