FederatedGrids: Federated Learning and Blockchain-Assisted P2P Energy Sharing

Ouns Bouachir, Member, IEEE, Moayad Aloqaily, Senior Member, IEEE, Öznur Özkasap, Senior Member, IEEE, and Faizan Ali

Abstract—Peer-to-Peer (P2P) energy trading platforms envisioned energy sectors to satisfy the increasing demand for energy. The vision of this paper is not only to trade energy but also to have part of it being shared. Therefore, this paper presents FederatedGrids which is a P2P energy trading and sharing platform inside and across microgrids. Energy sharing allows exchanging energy between the categories of consumers and prosumers in return for future benefits. FederatedGrids platform uses blockchain and federated learning to enable autonomous activities while providing trust and privacy among all participants. Indeed, based on various smart contracts using federated learning, FederatedGrids calculates a prediction of the future energy production and demand allowing the system to autonomously switch between trading and sharing, and enabling the prosumers to make decisions related to their participation in the energy sharing process. Up to our knowledge, this work is the first attempt to create a hybrid energy trading and sharing platform, with the real sharing meaning, and that uses federated learning over the smart contract for energy demand prediction. The experimental results showed a 17.8% decrease in energy cost for consumers and a 76.4% decrease in load over utility grids.

Index Terms—P2P energy sharing, blockchain, federated learning, smart contracts, microgrids.

I. INTRODUCTION

IN RECENT years, innovation in the information and communications sector has enabled the emergence of several technologies and applications. This progress is mainly based on the building of intelligent autonomous systems that are able to monitor, collect, process, and take decisions of various possible situations. The demand for these systems for basic living resources such as energy is growing exponentially. Recently, conventional energy resources have received attention as they are more and more decreasing. This is due to the fact that our demand for energy is continuously growing (population growth) while the supply is kind of constant. Thus, the world energy sector is in need of innovative solutions and new strategies for energy production to shift the world to a sustainable state. Renewable energy has always been considered a promising alternative solution. Various efforts have proposed innovative strategies to facilitate and simplify the involvement of this type of energy in the human-like tradition. One of the most advanced solutions to encourage citizens to rely on this particular type of energy is peer-to-peer (P2P) energy trading [1], [2]. This technology allows energy consumers (i.e., homes, vehicles, and buildings) to act as energy generators and buyers (i.e., prosumers and consumers). Indeed, those consumers can produce electricity using Renewable Energy Sources (RES) (e.g., sun, wind, etc.) to satisfy their personal demand and to sell the excess energy to buyers or to the main grid. Such a solution provides several social, economical, and environmental advantages.

Several clean energy platforms are being investigated while some are being commercialized by industry entities. The success of these platforms relies on the participation of the majority of prosumers and consumers to create a dynamic market for energy trading. Thus, many features should be taken into consideration to encourage this participation such as defining a set of rules of the various possible transactions among these systems. Furthermore, profit is an essential feature to attract prosumers and consumers. Thus, various strategies to minimize the energy trading cost are highly needed such as price control [1], [3] and energy consumption monitoring [4]. However, two main cases are not taken into consideration. The first case is related to consumers not being able to buy energy due to a lack of their funds. The second is related to prosumers taking charge of storage management of their excess energy that can be very costly. These expenses may affect the prosumers’ overall profit. These two situations may discourage potential members from participating in this ecosystem. As a result, we envision the concept of energy sharing to support and complement the energy trading concept. Therefore, energy sharing is considered as the amount of energy that participants exchange without a price.

P2P energy trading platforms are designed to provide a balanced (win-win) solution to all the participating entities: prosumers, consumers, and the main utility grid. This balance has two main factors: stable coverage of energy demand and profit maximization. Due to various circumstances, this balance faces some difficulties and instabilities. Indeed, it is possible that some consumers are not able to afford to buy energy at a given time which reduces the satisfaction rate of the trading platform. On the other hand, prosumers endorse
expensive charges to store and manage the excess of the produced energy. This will definitely reduce their profit. This situation can be more frequent and worse in the case of many prosumers trading their excess energy in a small geographic location (i.e., within a microgrid [5]) with a lack of diversified buyers. P2P energy exchange systems need to be looked at from a different angle to reach and maintain an efficient balance among the various participants. For instance, trading across the various microgrids reducing the amount of the stored energy while increasing the satisfaction rate of the overall platform. However, traditional trading cannot be always useful to satisfy consumers with a lack of resources to buy. Therefore, new strategies allowing sharing energy among participants when needed are required to satisfy all consumers and to minimize energy storage costs. These strategies will also allow maximizing the profit and providing a stable coverage of energy demand.

Other crucial issues are related to the security of these platforms as well as the privacy of the prosumers/consumers. Perform energy transaction exchange, billing the user, incentivizing more participants, and guaranteeing the delivery of this service are important aspects to consider. Prosumers/Consumers are always reluctant to engage in new technologies that do not guarantee robust security and privacy features. The solution to these concerns is to provide an efficient mechanism that comprehensively resolves these issues.

P2P energy trading platforms with the support of sharing concept and combined with advanced technologies such as blockchain and federated learning (FL) is the way ahead. This allows distributed storage of all the transactions securely while providing privacy and trust among the various participants. Combined with the concept of smart contracts, it allows increasing the intelligence and transparency level in the system. Various smart contracts are moving towards blockchain-assisted environments using smart meters for better energy management and creating a reliable smart power grid environment [2]. Federated learning is another paradigm that can improve the performance of P2P trading platforms. As a distributed machine learning approach, it is very suitable to deal with sensitive data or a huge amount of local data. It can play a key role in optimization the decision-making in such a platform by providing some predictions related to the future energy demand and production.

In this paper, we propose FederatedGrids, federated learning, and blockchain-assisted P2P energy sharing platform, which allows reaching a balanced level among participants by introducing a new strategy of energy sharing in support of energy trading to exploit the excuses and available amount of energy in the system. This concept of energy sharing allows exchanging energy to satisfy consumers’ and prosumers’ needs with a promise for future benefits. It relies on federated learning to predict future energy production and system load allowing the system to switch automatically to the sharing phase and the prosumers to make optimal decisions related to their energy exchanging strategies.

The novelties in FederatedGrids can be summarized as:

- A blockchain-based P2P energy trading allowing to buy/sell energy from/to peers inside a microgrid and between different microgrids is proposed. This platform uses a pricing mechanism that ensures the increase in utility of the participants;
- A new feature allowing to share energy between prosumers and consumers in exchange for future benefits and advantages;
- A federated learning-based model allowing to provide a prediction of the future energy production and system load is proposed. These predictions allow the system to make crucial decisions related to the amount of energy that can be used in the sharing process;
- An autonomous system based on smart contracts that enables the rules of energy sharing and trading inside and between the various microgrids has been developed;
- The system has been implemented and evaluated against several sharing scenarios.

The remainder of this paper is organized as follows. Section II discusses various research activities focusing on P2P energy trading and federated learning. Section III presents the blockchain-based P2P trading/sharing in FederatedGrids while the federated learning-based energy sharing mechanism is presented in Section IV. The experimental results and evaluations are presented in Section V and Section VI concludes the paper.

II. RELATED WORK

This section discusses various research activities aiming to increase the P2P energy exchange efficiency. Also, it provides a state-of-the-art of the most recent federated learning-based models.

A. P2P Energy Platforms

Recently, P2P energy trading has attracted various researchers. Most of the proposed models focus on energy trading and prosumer management inside and among microgrids [6], [7] aiming to increase the system efficiency. For instance, to rise the profit and encourage participants (i.e., prosumers and consumers), several researches have adopted strategies allowing to reduce the trading cost such as price optimization and consumption management. Indeed, price control and cost minimization are crucial to encourage consumers to buy energy from such platforms. Some researchers have also proposed new strategies to create prosumer groups to maximize the system profit [8]. On the other hand, different research activities uses ML methodologies to schedule the optimal time for energy trading based on a trade-off between the price variation and the prediction of energy demand and production. For instance, in [4], an intelligent P2P system between prosumers and consumers is proposed allowing a day-ahead energy control and generation schedule in order to meet the load demand of the smart grid. The authors in [9] used multi-commodity formulation (MCF) to optimize energy and communication resources in their proposed P2P energy trading platform. They aim to reduce the energy generation cost of the various prosumers and maximize utility with fair resource allocation. Indeed, the energy generation cost is an essential factor to attract prosumers. It includes costs related to the production
and storage of excess energy. In [10], the authors presented a P2P energy trading model that makes the decision of optimal combinations of energy production, storage and consumption strategies. It aims to minimize the electricity cost and make decisions on investments, selling and buying from/to the grid.

Other research activities focus on energy price optimization to reduce the cost and increase the efficiency of the trading platform. Indeed, some solutions allow price bidding before the transaction to encourage more consumers to participate and negotiate the prices such as in [11]. In [3], the authors proposed a model that takes into consideration the flexible behavior of the community members, thus, it allows to calculate the price the day after the energy delivery. It aims at minimizing the effect of uncertainty of demand to reduce the trading cost. The authors in [12], proposed a model that allows consumers to adjust their energy consumption behavior based on the price and the values of energy production. The price in this model is defined by a non-cooperative game based on competition among the prosumers. In these cases, the price depends directly on the energy production and demand. In [13], the authors used an iterative algorithm to determine the energy price and trading amount during each round of an auction. These calculations allow building a pricing strategy to better understand the market variation. The authors in [14], considered load shifting and RES generation amounts to determine the price of energy between P2P traders. In the same context, an alternative direction method of multiplier (ADMM) based distributed approach was proposed in [15] to calculate the energy price. The authors in [16], formulated a distributed algorithm for trading price and consumption optimization that maximizes the player’s profit. On the other hand, in [17], a model was proposed enabling prosumers to adjust their planned power flow based on the variable energy price, load predictions, and the amount of available renewable energy.

All the presented models are summarized in Table I. They aim at increasing the profit and the efficiency of the trading platforms to encourage prosumers and consumers to participate and engage in this process. These solutions did not tackle the energy sharing concept. They only focus on selling/buying energy among the various participants in return for money without taking into consideration the consumers not able to buy energy due to a lack of their funds and, prosumers with excess energy that should afford the high cost of energy storage. FederatedGrids focuses on these two situations and presents an energy sharing strategy allowing to support the traditional trading concept.

B. Federated Learning

Most smart systems require the collection and the processing of large amounts of data that is useful in analyzing, classifying, and predicting future behaviors. Federated learning allows cooperative distributed learning and training of local models to provide more efficient results and better privacy. Recently, various researchers have proposed new strategies to include federated learning in different applications. The authors in [18], proposed the use of distributed learning models based on gradient descent approaches to enhance the performance and scalability of applications that require the analysis of huge amounts of data. Also, in [19], an edge-cloud hierarchical federated learning system was presented allowing multiple edge servers to run partial model aggregation. In [20], the authors proposed a federated learning-based model that involves a sampled subset of the user equipments in the training process performed mainly by the edge nodes. The subset of user equipments is replaced during each round to minimize the energy consumption and the learning completion time. To improve security in federated learning, the authors in [21] proposed blockchain-based federated learning with a committee consensus model (BFLC). The blockchain is used for two main things: global storage and the local model update exchange. This framework allows reducing the consensus computing and malicious attacks.

In the context of federated learning for energy systems, fewer research activities have been proposed. In [22], a federated learning-based model that predicts energy demand for electric vehicles is presented. It allows charging stations to share their trained models without revealing their raw datasets. The collected models are processed by the charging station providers to predict the electric vehicles’ energy demands in order to optimize the consumption and the price. This model aims at minimizing the energy cost and maximizing the participant’s satisfaction rate. In [23], IFed model was presented. It is a federated learning model that allows electric providers to help power IoT users to ensure the privacy of their local data and to minimize resource consumption. Moreover, federated learning was used in smart grids by the authors in [24] to reduce the volume of data used to train a deep learning model. It allows to determining the household forecasting load without compromising their privacy. In the same context, the authors in [25], designed a demand response algorithm based on federated learning among residential users. This decentralized deep learning approach allows controlling the household loads in order to schedule their demand and obtain feasible power flow while respecting users’ privacy. Table II summarizes the various presented activities related to federated learning.

Federated learning has shown improved results in terms of learning efficiency and resource cost while respecting the privacy of the raw datasets. To the best of our knowledge, this article is the first attempt to create a P2P hybrid energy trading and sharing platform, with the real sharing meaning, that uses federated learning over the smart contract for energy demand prediction.

III. SYSTEM ARCHITECTURE

FederatedGrids system overview is presented in Figure 1. The system is composed of various prosumers and consumers connected inside multiple microgrids. Consumers are energy buyers while prosumers represent consumers able to produce energy through renewable sources (e.g., sun or wind) on a small scale and sell their surplus of energy. The participants in this platform can be any component of the smart city that consumes and/or produces energy such as buildings, homes, and vehicles. The prosumers are equipped with a smart
microgrids. Energy trading and sharing in the exchange of energy within and among the various participants and able to buy/sell energy from/to the main grids when needed.

A. Trading in Microgrids Based Architecture

1) Microgrids Architecture: A microgrid is the collection of connected consumers, prosumers, and DERs with electric wires in a given geographic area. These participants share the same medium-voltage/low-voltage (MV/LV) transformer [26]. A microgrid is a smart entity connected with the main power network (utility grids) and their centralized control systems responsible for energy trading and sharing. It can manage, operate, control, and coordinate the distributed energy resources among the various participants and able to buy/sell energy from/to the main grids when needed.

### Table 1: Literature Review Summary

<table>
<thead>
<tr>
<th>Ref</th>
<th>Blockchain</th>
<th>Microgrids Participation</th>
<th>Target</th>
<th>Used Mechanisms</th>
</tr>
</thead>
<tbody>
<tr>
<td>[4]</td>
<td>Yes</td>
<td>Only within peers</td>
<td>Day-ahead DER energy control schedule</td>
<td>Bi-directional LSTM with smart contracts</td>
</tr>
<tr>
<td>[9]</td>
<td>No</td>
<td>Only within peers</td>
<td>Minimizing energy generation cost and maximize utility with fair resource allocation</td>
<td>Multi-commodity formulation technique</td>
</tr>
<tr>
<td>[10]</td>
<td>No</td>
<td>Only within peers</td>
<td>Optimal combination of production, storage, consumption strategy</td>
<td>Stochastic mixed integer programming problem</td>
</tr>
<tr>
<td>[3]</td>
<td>No</td>
<td>Only within peers</td>
<td>Reducing the demand uncertainty effect</td>
<td>Calculating the transaction prices the day after</td>
</tr>
<tr>
<td>[12]</td>
<td>No</td>
<td>Only within peers</td>
<td>Adjusting buyer’s consumption based on the price and quantity of traded energy</td>
<td>Non-Cooperative game</td>
</tr>
<tr>
<td>[13]</td>
<td>No</td>
<td>Only within peers</td>
<td>Designing an adaptive pricing strategy to better understand the market</td>
<td>Iterative algorithm</td>
</tr>
<tr>
<td>[14]</td>
<td>No</td>
<td>Only within peers</td>
<td>Determining the optimal price</td>
<td>Dynamic price using load shifting and RES generation values</td>
</tr>
<tr>
<td>[15]</td>
<td>No</td>
<td>Between peers inside a microgrid</td>
<td>Determining the optimal price</td>
<td>Alternative direction method of multiplier</td>
</tr>
<tr>
<td>[16]</td>
<td>No</td>
<td>Only within peers</td>
<td>Optimize trading price and consumption while maximizing players profits</td>
<td>Distributed algorithm using Stackelberg game</td>
</tr>
<tr>
<td>[17]</td>
<td>No</td>
<td>Only within peers</td>
<td>Prosumers adjust their power flow based on energy price and load predictions</td>
<td>Distributed price-directed optimization mechanism</td>
</tr>
<tr>
<td><strong>FederatedGrids</strong></td>
<td>Yes</td>
<td>Inside and among microgrids</td>
<td>Introducing the energy sharing concept that increases the platform efficiency</td>
<td>CNN based Federated learning</td>
</tr>
</tbody>
</table>

energy management system (EMS) that allows smart monitoring of energy consumption, production, trading, and sharing. A group of connected prosumers and consumers inside a given geographic area defines a microgrid. All the microgrids are connected through the utility grids usually owned by the government. The utility grids generate energy through conventional (i.e., gas, oil, and coal) and renewable sources (i.e., solar panels, windmills, and water power). Also, they provide large batteries that can store surplus energy. They are controlled by a centralized energy manager with computational capacity and cloud storage to handle the energy trading requests.

FederatedGrids is a federated learning and blockchain-assisted P2P energy trading and sharing platform that allows the exchange of energy within and among the various microgrids. Energy trading and sharing in FederatedGrids and the role of blockchain are explained in the next subsections.
TABLE II
FEDERATED LEARNING RELATED LITERATURE REVIEW SUMMARY

<table>
<thead>
<tr>
<th>Article</th>
<th>Application</th>
<th>Type of Learning</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>[18]</td>
<td>MEC-inspired edge server</td>
<td>Federated learning</td>
<td>Improved aggregation scalability issue</td>
</tr>
<tr>
<td>[19]</td>
<td>Using cloud, edge and clients for improved learning</td>
<td>Hierarchical federated learning, handling Non-IID data</td>
<td>Improved convergence, use of edge computing</td>
</tr>
<tr>
<td>[20]</td>
<td>Edge nodes and User equipment’s</td>
<td>Federated learning based stochastic gradient descent</td>
<td>Minimized energy consumption and federated learning completion time</td>
</tr>
<tr>
<td>[21]</td>
<td>Blockchain based federated learning</td>
<td>Blockchain based federated learning with committee consensus</td>
<td>Reduced the amount of consensus computing and reduced malicious attacks</td>
</tr>
<tr>
<td>[22]</td>
<td>Energy demand prediction (EV)</td>
<td>DNN Federated learning with clusters</td>
<td>Reduced communication overhead</td>
</tr>
<tr>
<td>[23]</td>
<td>Federated learning for Power IoT users</td>
<td>Aggregation model based on stochastic gradient descent</td>
<td>Provided local data privacy and reduced resource consumption</td>
</tr>
<tr>
<td>[24]</td>
<td>Electric load forecast, short term load forecasting with stochastic consumption profiles</td>
<td>LSTM federated learning with edge computing</td>
<td>Improved privacy and reduced networking load</td>
</tr>
<tr>
<td>[25]</td>
<td>Load control for household</td>
<td>Decentralized deep reinforcement Learning with actor critic method</td>
<td>Obtained feasible power flow</td>
</tr>
</tbody>
</table>

Fig. 1. FederatedGrids system overview.

FederatedGrids enables P2P trading inside and between microgrids. Indeed, buyers and sellers can directly trade energy with each other. The trading process can be done in two stages. In the first stage, the participants in a microgrid cover the local energy demand by trading energy between each other (Figure 2). This level can have three different results:

1) **The produced energy is equal to the local demand:** All the surplus of energy is traded and all the local demands inside the microgrid are fulfilled;
2) **The produced energy is less than the local demand:** All the surplus of energy is traded, however, some requests are still active;

3) **The produced energy is more than the local demand:** All the local requests are fulfilled, however, prosumers still have a surplus of energy.

In situation (2), traditionally, the microgrid communicates with the main power grid network to cover the local energy demand that can have higher cost than the energy purchased from the prosumers inside the same microgrid. Besides, prosumers in (3) have to take charge of the costly energy storage since microgrids do not contain centralized storage like utility grids. These two situations are frequent among the various microgrids. FederatedGrids allows creating a balance by allowing the P2P trading between sellers and buyers in different microgrids in a second stage. The main target of this model is to maintain the power and price balance between the microgrids while reducing the dependency on the main utility grid.

Fig. 2. FederatedGrids trading and sharing phases.
grids. This allows to minimize the loss in energy transfer and also the overall price of energy.

After covering the local demands, the prosumers in the situation (3) have two options.

- First, they can sell their energy to the main grids at a price fixed by the utility grids. This price incorporates the cost of the energy transfer through the infrastructure and the cost of energy losses.
- Second, the prosumers can choose to sell their excess energy to other microgrids. This allows to minimize the dependency of the microgrid community on the main grid. Also, sharing energy with other microgrids in the community allows to reduce the load of the utility grid.

2) Price Calculation: The trading price is calculated at the beginning of a time step for each energy trading request received from a consumer. This depends on various factors such as the total energy demand and production inside the microgrid, distance between the consumer and the prosumer and, the type of renewable energy source (e.g., sun, wind, etc.) used by the prosumer. After this, the price of the energy trading between the various microgrids can be determined. Indeed, it takes into consideration the distance between the two microgrids and the utility infrastructure (i.e., wires) used to transport the exchanged energy. The main rule of this system can be defined by:

\[
P_{P2P} \leq P_{P2M} \leq P_{P2U}
\]

where \(P_{P2P}\) is the trading price between the two peers inside the same microgrid, \(P_{P2M}\) is the trading price between the consumer and another microgrid and \(P_{P2U}\) represents the utility price. More details about the trading price calculation can be found in [27].

B. Blockchain-Based Energy Trading

As presented in Figure 1, blockchain is an essential component of the FederatedGrids platform. It handles all the trading and sharing transactions between the prosumers/consumers inside a microgrid, between various microgrids, and with the utility grids. In addition, the blockchain allows the storage of all the information related to the different entities: prosumers, consumers, utility grids, and microgrids. In FederatedGrids, various blockchain smart contracts are used to create agreements and contracts between the traders to facilitate and simplify their transactions.

The system process is based on a cycle that is divided into two main phases: energy trading and energy sharing. In the first phase, the energy is traded among buyers and sellers. Interested prosumers send their participation requests containing their locations, available energy, and microgrid information. All this information should be stored over the blockchain. On the other hand, each consumer sends a registration request containing its energy demand that should be kept over the blockchain. The collected information is used for two main purposes:

1) It allows each microgrid to separate the energy needed for the local consumers. The remaining energy can then be traded and shared across the microgrids.

2) It allows the Federated Module (explained in Section IV-B) to predict the total load in the system for the next time slot that is very useful in making decisions related to energy sharing phase.

After this, the System sets the energy price (as explained in Section III-A2) for energy trading among microgrids. The energy requests should be fulfilled locally inside each microgrid first. Then, a new match loop should be done to find the prosumers in other microgrids able to sell energy. All these transactions are handled in the blockchain.

The main contract enables the payment transfer between the consumer and the prosumer account and the payment of the utility grid for the usage of its resources. During the trading phase, the system keeps tracking of the amount of energy

<table>
<thead>
<tr>
<th>Algorithm 1: Energy Request Processing Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> Number grids, Utility Price, Resource Price, Load</td>
</tr>
<tr>
<td>(N_g \leftarrow \text{number Grids} )</td>
</tr>
<tr>
<td>(P_u \leftarrow \text{price Utility} )</td>
</tr>
<tr>
<td>(P_r \leftarrow \text{Resource Price} )</td>
</tr>
<tr>
<td>(E_{\text{traded}} \leftarrow 0; )</td>
</tr>
<tr>
<td>(\text{Sharing} \leftarrow \text{False}; )</td>
</tr>
<tr>
<td>(\text{Load}_{\text{predicted}} \leftarrow \text{Load} )</td>
</tr>
<tr>
<td><strong>Function</strong> EnergyRequest ((\text{energyAmount, reqType})):</td>
</tr>
<tr>
<td>consumer (\leftarrow \text{request.location}; )</td>
</tr>
<tr>
<td>(C_{\text{total}} \leftarrow 0; )</td>
</tr>
<tr>
<td>(C_{\text{share}} \leftarrow 0; )</td>
</tr>
<tr>
<td>(\text{amount}_\text{left, prosumers} \leftarrow \text{GetProsumers(energyAmount)} )</td>
</tr>
<tr>
<td>(\text{// Determine Prosumers that match energy requirement} )</td>
</tr>
<tr>
<td>foreach ((p \in \text{prosumers})) do</td>
</tr>
<tr>
<td>(p_p \leftarrow \text{calculate price}(); )</td>
</tr>
<tr>
<td>(p.\text{sale} \leftarrow p.\text{sale} + p_p * \text{energyAmount}; )</td>
</tr>
<tr>
<td>(C_{\text{total}} \leftarrow C_{\text{total}} + p_p * \text{energyAmount}; )</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>if ((\text{amount}_\text{left} \neq 0)) then</td>
</tr>
<tr>
<td>(\text{amount}_\text{left, microgrids} \leftarrow \text{FindMG}() )</td>
</tr>
<tr>
<td>(\text{// Determine microgrids that match energy requirement} )</td>
</tr>
<tr>
<td>foreach ((p_m \in \text{prosumers})) do</td>
</tr>
<tr>
<td>(\text{amount}_\text{left, prosumers} \leftarrow \text{FindProsumer(p_m)} )</td>
</tr>
<tr>
<td>(\text{// Find Prosumer within p_m that match energy requirement} )</td>
</tr>
<tr>
<td>(p_g \leftarrow \text{calculate price}(); )</td>
</tr>
<tr>
<td>foreach ((p \in \text{prosumers})) do</td>
</tr>
<tr>
<td>(c_m \leftarrow p_g * \text{energyAmount} )</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>(C_{\text{total}} \leftarrow C_{\text{total}} + c_m + P_r )</td>
</tr>
<tr>
<td>(C_{\text{share}} \leftarrow C_{\text{share}} + P_r )</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>if ((\text{amount}_\text{left} \neq 0)) then</td>
</tr>
<tr>
<td>(c_u \leftarrow \text{amount}_\text{left} \times P_u )</td>
</tr>
<tr>
<td>(C_{\text{total}} \leftarrow C_{\text{total}} + c_u )</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>if ((\text{reqType} = \text{’Trade’})) then</td>
</tr>
<tr>
<td>consumer (c) pays (C_{\text{total}})</td>
</tr>
<tr>
<td>else</td>
</tr>
<tr>
<td>if ((\text{Sharing} = \text{True})) then</td>
</tr>
<tr>
<td>consumer (c) pays (C_{\text{share}})</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>(E_{\text{traded}} \leftarrow E_{\text{traded}} + \text{amount} )</td>
</tr>
<tr>
<td>if ((\text{Load}<em>{\text{predicted}} \leq E</em>{\text{traded}})) then</td>
</tr>
<tr>
<td>(\text{Sharing} \leftarrow \text{True} )</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>End Function</td>
</tr>
</tbody>
</table>
exchanged between the various peers. This phase ends when all the demands are fulfilled allowing the system to move to the sharing phase. At the beginning of the new sharing phase, the consumers with lack of funds send the energy sharing requests that can be covered in case there is still energy in the system.

At the end of the trading/sharing cycle, prosumers update their available produced energy for the next time step. Algorithm 1 presents the mechanism used to handle energy requests in the system and Figure 3 shows the various interactions between the system entities.

The energy sharing process and federated learning module are detailed in the next section.

IV. ENERGY SHARING IN FederatedGrids

FederatedGrids provides two different ways to exchange energy between the various peers: trading and sharing allowing to reach the required “win-win” balance among all the participants. Energy sharing mechanisms are detailed in this section.

A. Energy Sharing Mechanisms

The energy sharing starts at the end of the trading phase. The only case it can work when all the transactions after some time are ended fulfilling the buyers’ demand and there are prosumers with energy left to sell. FederatedGrids has two different aspects of energy sharing: energy sharing between peers and energy resource sharing, as described next.

1) Energy Sharing Between Peers: This aspect considers the rest of consumers do not have funds to buy energy and weren’t able to participate in the energy trading phase. Other than buying/selling the energy, prosumers can share the energy with the fellow participants to cover their energy needs in return for future benefits discussed later.

Similar to trading, consumers initiate the sharing process by sending their demand requests to the system. The sharing transaction is, then, fulfilled inside and among the microgrids. All the information related to this transaction including the identity of the participant prosumer and consumer is stored in the blockchain for future references.

2) Energy Resources Sharing: The definition of sharing in FederatedGrids differs from P2P trading in this sense. A microgrid includes multiple households where some may have their own storage while some other may have RES and others may have both of them. If the energy load of one generator increases, the RES or the storage capacity may increase. Both solutions are considered costly since they may require all infrastructure modification at the client side. This prosumer, then, can share storage with central energy sharing or with other participants’ infrastructure. Thus, participants should buy a central storage to be used when needed that can be costly. In this context, FederatedGrids, enables producers to share the batteries of other participants to store their surplus energy allowing to reduce the extra storage cost.

3) Energy Sharing Workflow: The two sharing cases mentioned above are indirectly related to each other. Sharing energy to fulfill the demand of a consumer or to store in other participants’ batteries are both similar from the prosumer perspective who is exchanging the energy for future benefits. These benefits can be based on various factors such as the interest, the system location, etc. For instance, providing the prosumers, who participated in the sharing process, high privileges to sell and buy energy when needed, can be one of these benefits. Therefore, we need to focus on energy sharing to reduce the complexity of the problem.

The system goes to the sharing phase at the end of the trading. The consumer sends the sharing request that should be checked by the request handler. The main contract, first verifies if the sharing is enabled in the microgrid so the energy is shared without any cost. If sharing process is not available in the local microgrid, the other microgrids should be contacted for sharing. The consumers sharing energy from a different microgrid must take in charge the cost of the utility.

B. Federated Learning in FederatedGrids

In FederatedGrids, the sharing mechanism requires the global prediction of energy demand and energy generation for all the consumers and the prosumers. Indeed, this prediction allows prosumers to take a crucial decision related to their participation in the sharing process. Indeed, if the predictions show a decrease in the energy production level with an increase in the system demand, the prosumer is encouraged to save the available amount of energy for the coming time step. On the other hand, if the predictions show an increase in the energy production for the coming time step that may saturate the local storage, participating in the sharing process is considered as a wise solution. Centralized machine learning can be useful in calculating such a global model but it has several limitations that can be improved by the use of federated learning.

FederatedGrids is a blockchain-based federated learning solution. It uses Convolution Neural Network (CNN) based federated learning to predict the total load in the system using aggregation over the smart contract. Blockchain has two main tasks: it allows keeping the global model and calculating basic federated averaging as presented in Algorithm 2. The local updates and predictions are performed off-chain (more description is in Section V).

Using smart contracts, the learning model can determine various information by using the appropriate amount of labeled data. It allows to:

- provide a prediction of the energy generation for prosumers;
### Algorithm 2: Blockchain Federated Calculation Algorithm

**Input:** Initial model $M$, number of nodes to select $N_p$, learning rate $\gamma$, number of total Rounds $R$  

selected $= [1]$;  

$M_g \leftarrow \text{SetGM(M)}$ // Global model on blockchain  

set to $M$  

initialize training round $r = 0$;  

while $r < R$ do  

selected $\leftarrow \text{Select}(N_p)$ // Select $N_p$ nodes among participants  

foreach $s \in$ selected do  

if $s$ has local data then  

$m_l \leftarrow s$ fetches the model $M_g$ from blockchain;  

$e \leftarrow s$ get number of epochs from blockchain;  

$s$ computes local gradient $g_l$ for epochs $e$;  

$s$ updates the local model $m_l \leftarrow m_l - \gamma g_l$;  

$s$ sends updated model to blockchain;  

end  

end  

BCFedAvg() // Upon receiving $N_p$ models,  

use federated averaging over blockchain to form the global model  

next round start, $r \leftarrow r + 1$

---

- provide a prediction of the energy load of individual consumers as being done in [28];  
- determine the amount of energy to share/trade in the next time step;  
- provide a prediction of the price of energy in the next time step.

These predictions are calculated using data with specific fields like prosumer and consumer geographic locations and weather information. These calculations are done using Convolution layer and dense layer and can be optimized using a different number of layers.

Federated learning allows reducing the network load. Indeed, in conventional machine learning, all training data, usually in Gigabytes, should be shared with the central server to perform some iterations causing a huge load on the network and affecting its performance. With federated learning, the data is trained locally and only weights of the models, in kilobytes only, are shared with the central server.

The following equation can be used to calculate the gain in networking load:

\[
\text{Load}_{ML} = \sum_{i=1}^{N} \text{Size}_i \tag{2}
\]

\[
\text{Load}_{FL} = 2 \times \text{Size}_{nf} \times N_p \times R \tag{3}
\]

where $\text{Load}_{ML}$ represents the load for machine learning, and $\text{Load}_{FL}$ is the load for federated learning. $\text{Size}_i$ represents the size of data sent by participant $i$, $\text{Size}_{nf}$ is the size of the federated model, $N_p$ represents the number of participating nodes in a round and $R$ defines the number of rounds performed. In equation (3), the parameters are multiplied by ‘2’ because $\text{Size}_{nf}$ amount of load is sent twice, once from server to client (while fetching the model) and then from client to server (for aggregation).

The gain is defined as follows using equations (2) and (3):

\[
\text{Gain}_{FL} = 1 - \frac{\text{Load}_{FL}}{\text{Load}_{ML}}. \tag{4}
\]

### C. Smart Contracts in FederatedGrids

The use of blockchain allows running several smart contracts to ensure transparent energy trading and sharing markets where all participants trust the contracts hence eliminating the requirement of central parties to control energy management. Due to this nature of the smart contract, FederatedGrids develops a system for efficient P2P and Microgrid-to-Microgrid (M2M) energy trading and sharing simply deployed by any body with access to the smart contracts. These smart contracts can be deployed over a public or a private blockchain. The public blockchain allows the participation of many miners making faster block creating. On the other hand, the private blockchain enables limited number of participants and as most of the miners are self-owned, there is less probability of attack occurrence.

Five different types of smart contracts are designed in this platform.

1) **Main Smart Contract:** All operations of energy trading and sharing in the system are monitored by the main smart contract. All participants can communicate directly with the smart contract since it has a key role in trading and sharing mechanisms. It allows to perform several main tasks.

- Registering the various participants (consumers and prosumers) and microgrids: Before generating any requests, all prosumers and consumers should be registered. After that, these participants will be able to send their energy surplus and deficit requests that are handled through the main smart contract. M2M and P2P contracts are updated based on the energy surplus received requests.
- Determine the list of prosumers that can match with the consumers within and across the microgrids for energy trading and sharing: For each energy buying request received, it uses the function `findProsumer` of P2P smart contract that matches the consumer with local prosumers. When the local demand exceeds the available local energy, the `findMicrogrid` function is called. It aims to find the microgrid with surplus energy for energy trade. In case there is no more energy to cover the needs, it contacts the grids to buy the energy. In the case of the energy sharing requests, the main contract first checks if the sharing phase is enabled, then uses the same matching mechanism. Sharing phase doesn’t involve any transfer of resources since all prosumers and consumers, participating in this process, are recorded in a list to be compensated in the future.
- All the required price calculation for the next time step and the needed prosumers’ compensation for their shared energy.

2) **P2P Smart Contracts:** This contract maintains the information of the local peers when they register through the main contract. Only the main contract can invoke the P2P contract. The `findProsumer` function of the main smart contract allows checking the microgrid id of the consumer and finding...
the list of prosumers that can cover the energy demands of the requester.

3) **M2M Smart Contracts:** This contract maintains all the microgrids’ information in a dictionary-like data structure called maps. It is used in case of energy demand between microgrids to facilitate this exchange. The `findGrid` function in the main smart contract allows finding an suitable grid able to fulfill the energy demand of the consumer.

4) **P2G Smart Contract:** Once all the energy transactions between peers and microgrids are completed, P2G smart contract is used by the consumers to buy the energy from grids. All the system participants can directly access this smart contract deployed by the utility grid.

5) **Federated Learning Smart Contract:** In addition to the global model, Federated learning contract performs several operations: it allows to select the various participants for the coming round. Also, it shares and gathers the global model to/from the participants and perform the aggregation at the end of each round. It uses three different functions: `GetModel` gathers the global learning model parameters from the contract. Only the selected participants are able to receive the model. Then, `SetModel` allows the users to send the updated parameters. It keeps counting the submitted models and performs a comparison with the number of selected participants in the round ‘n’.

V. EXPERIMENTS AND RESULTS

The smart contracts were created on Remix IDE\(^1\) to test the accuracy of the contracts. The smart contracts were implemented on local Ethereum using the Ganache framework and Geth. The prototype system is based on the following assumptions:

1) All the participants register themselves to the system through the main contract.

2) A participant is either a prosumer or a consumer in each time step. There is no relationship between the prosumers and the consumers.

3) A consumer is either a one able to buy energy or the one without available resource to cover energy buying requirements.

4) The microgrid to microgrid trading cost is defined at the beginning of each time step.

5) The P2P trading price should be lower than the trading cost between microgrids and lower than the utility price as presented in equation (1).

The participants are able to communicate with the main smart contract through Energy Management Systems (EMS) developed in Python Language with web3 library.\(^2\) Each participant has an e-wallet provided by Ganache having different addresses. The system is also connected with the utility grid to get energy when needed.

All the transactions are executed through the blockchain-based on the various smart contracts defined above. In these experiments, the Hourly Energy Demand Generation data-set,\(^3\) containing 4 years of electrical consumption, generation, pricing and weather data for Spain is used. It contains two major pieces of information needed for the proposed model for each timestamp: the energy generated by reusable energy resources such as wind and solar energy and the data for energy demand. The dataset also involves information related to the weather as it was part of a project from the Open Weather API to study the impact of the weather conditions on renewable energy in the area. The prediction method can, then, use this information to determine future energy consumption and production in the area.

A. Federated Learning Simulation Setup

The simulations were conducted over HP Spectre X360 with an Intel i7 processor and 16GB memory. To run the local training, Keras’s CNN model was used. The resultant weights are then shared with Ethereum Smart Contracts (developed over Soli di ty) using the function `set_model` taking the various weights as input. After gathering the N models, the `federated_averaging` function is initiated to calculate the weight averaging. The global model is then shared with all participants through `get_model`. This function allows to send back model weights that will be then plugged in the Keras model to run epochs for the next round.

Hyperparameters and optimizations are important for deep learning models, but, in this paper, we focus on providing proof of concept of federated learning paradigm for load prediction using blockchain. Furthermore, a simple model that uses 1 CNN layer in addition to 2 dense layers are used to reduce the number of weights (thus, reduce the size) of the model. To evaluate the model, RMSE and MAPE are considered (Table III) The expressions for RMSE and MAPE are:

\[
RMSE = \sqrt{\frac{\sum_{i=1}^{N_P} (y_{ai} - y_{pi})^2}{N_P}}
\]  
\[MAPE = \frac{100\%}{N_P} \sum_{i=1}^{N_P} \left| \frac{y_{ai} - y_{pi}}{y_{ai}} \right|
\]

where \(y_{pi}\) represents the predicted value, \(y_{ai}\) is the actual value and \(N_P\) represents the number of predicted values.

B. Federated Learning Results

To evaluate the prediction mechanism, three types of models were used: (1) Machine learning model, (2) Federated Learning centralized, and (3) Federated Learning blockchain. The values of RMSE and MAPE when models are trained are shown in Table III. For machine learning models 20 epochs were used while 100 epochs per participant were considered for the two federated learning models. Indeed, the decrease in the amount of data increases the need for more epochs to get desired results. In addition, in each round, 5 rounds with 20 nodes were used in federated learning scenarios. The

\(^1\)Remix - Ethereum IDE. [online] Available at https://remix.ethereum.org [Accessed March 2021].


TABLE III
RMSE AND MAPE RESULTS FOR GLOBAL MODELS IN THE SELECTED SCENARIO (20 NODES AND 5 ROUNDS)

<table>
<thead>
<tr>
<th>Model</th>
<th>RMSE</th>
<th>MAPE</th>
<th>Epochs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine Learning</td>
<td>0.0116</td>
<td>1.39%</td>
<td>20</td>
</tr>
<tr>
<td>Federated Learning</td>
<td>0.0237</td>
<td>2.23%</td>
<td>100</td>
</tr>
<tr>
<td>Federated Learning</td>
<td>0.0332</td>
<td>3.34%</td>
<td>100</td>
</tr>
<tr>
<td>Blockchain</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

MAPE results show that federated learning performs comparably to machine learning. The MAPE for the blockchain model is high because Solidity does not handle floating-point numbers, hence techniques were used to store weights over Solidity such as multiplication with $2^{32}$. The prediction results and training/validation loss are shown in Figures 4 and 5. In those experiments, the predictions obtained by federated learning predictions and machine learning were compared.

To evaluate the network load gain, the size of the model and the size of data were calculated. The size of the model is 0.312Kb, and the size of data is 25.6Mb. They allow to determine the gain using equation (4). As shown in Figure 6, various values for $R$ and $N_P$ were used to understand the effect number of rounds and the number of participants overload. The figure shows the network gain with (1) variable $R$ values (from 1 to 10) when $N_P$ is equal to 20 (the red curve) and with (2) variable $N_P$ values (from 1 to 10) when $R$ is equal to 20 (the blue curve). We can see that the impact of the number of rounds over gain is less important than that of the number of participating nodes. Also, the maximum of 99.7% gain was obtained with 1 round containing 5 participants.

C. Energy Sharing Results

The energy sharing mechanism is discussed in this section. As presented in Section IV-A, it may happen to have situations where the consumers are not able to buy energy due to a lack of their funds and the prosumers have to afford the costly storage of the unsold energy.

In this case, the consumer can take advantage of the prosumers’ unsold energy in case the sharing phase is active. To evaluate this, a system of 5 microgrids grouping the total of 100 prosumers and 300 consumers is considered. Figure 7 shows the total energy production in the system. At the end of the trading phase, 100 energy sharing requests are shared by the interested consumers. As shown in Figure 8, several requests were covered and many consumers took advantage of energy sharing. The figure also shows that when a small amount of sharing energy is requested, more consumers are satisfied. This can be explained by the fact that the system works on first-come-first-served basis. Indeed, if the requested amount of energy sharing is large, the system will run out of energy faster. Also, the figure shows that more consumers’ requests are covered when the energy is shared...
inside the microgrid and between the microgrids. Indeed, in this case, more energy is available to be shared (Figure 9). Figure 11 confirms the availability of prosumers able to contribute towards sharing. The amount of energy shared inside a microgrid depends directly on the number of available prosumers during this phase. Again, the system covers the local requests inside the microgrid before sharing the energy with consumers in other microgrids. Figure 10 shows the total energy traded and shared, disregarding the non-covered sharing requests, in each microgrid. We can notice that microgrids 2 and 5 end the trading phase with no energy left but they are able to benefit from the surplus of unsold energy of microgrid 4 (participant with the largest amount of energy). Figure 12 presents the total shared and traded energy for each prosumer in microgrid 3. It shows that the prosumer having the largest amount of energy, has the highest contribution in both trading and sharing. Indeed, the system matches the energy requests with the prosumer with the most sharing participation and then the prosumer with the most available energy. The evaluation of FederatedGrids demonstrated an enhancement in the satisfaction rate since energy sharing enables the creation of a collaborative environment.

VI. CONCLUSION

Maximizing the satisfaction and the profit of all participants is the target of all P2P energy exchanging systems. This is possible by optimizing the energy exchanging cost that allows reaching a balance among the various participants. This paper presents FederatedGrids which is a federated learning and blockchain-assisted P2P energy trading and sharing platform creates a collaborative environment that maintain a satisfaction balance among all the participants in various microgrids. After fulfilling all the trading demands inside and between microgrids, consumers with a lack of fund resources are able to get their demand from prosumers with energy left to sell, in return for future benefits. FederatedGrids relies on two main advanced paradigms to guarantee an efficient performance: Blockchain and Federated learning. Federated learning enables predicting future energy production and system load allowing the prosumers to make optimal decisions related to their energy exchanging strategies. On the other hand, blockchain and various smart contracts are implemented enabling trustworthy and transparent transactions in the system. In the
future, we plan to focus on the energy sharing rewarding mechanism and further evaluation of the overall energy trading and sharing system with an emphasis on the privacy of the various participants.

REFERENCES


