Received November 14, 2020, accepted December 6, 2020, date of publication December 14, 2020, date of current version December 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3044612

AEBIS: AI-Enabled Blockchain-Based Electric Vehicle Integration System for Power Management in Smart Grid Platform

ZHISHANG WANG [1], MARK OGBODO [1], (Student Member, IEEE), HUAKUN HUANG [2], CHEN QIU [1], MASAYUKI HISADA [2], (Senior Member, IEEE), AND ABDERAZEK BEN ABDALLAH [1], (Senior Member, IEEE)

[1] Adaptive Systems Laboratory, Graduate School of Computer Science and Engineering, The University of Aizu, Aizuwakamatsu 965-8580, Japan
[2] Aizu Computer Science Laboratories, Inc., Aizuwakamatsu 965-0872, Japan

Corresponding authors: Zhishang Wang (d8221102@u-aizu.ac.jp) and Abderazek Ben Abdallah (benab@u-aizu.ac.jp)

This work was supported by the Fukushima Research and DevelopmentProgram (Demonstration Phase) Fund for Renewable Energy Technology, Japan, under Grant 682.

ABSTRACT A Virtual Power Plant (VPP) is a network of distributed power generating units, flexible power consumers, and storage systems. A VPP balances the load on the grid by allocating the power generated by different linked units during periods of peak load. Demand-side energy equipment, such as Electric Vehicles (EVs) and mobile robots, can also balance the energy supply-demand when effectively deployed. However, fluctuation of the power generated by the various power units makes the supply power balance a challenging goal. Moreover, the communication security between a VPP aggregator and end facilities is critical and has not been carefully investigated. This paper proposes an AI-enabled, blockchain-based electric vehicle integration system, named AEBIS for power management in a smart grid platform. The system is based on an artificial neural-network and federated learning approaches for EV charge prediction, in which the EV fleet is employed as a consumer and as a supplier of electrical energy within a VPP platform. The evaluation results show that the proposed approach achieved high power consumption forecast with R^2 score of 0.938 in the conventional training scenario. When applying a federated learning approach, the accuracy decreased by only 1.7%. Therefore, with the accurate prediction of power consumption, the proposed system produces reliable and timely service to supply extra electricity from the vehicular network, decreasing the power fluctuation level. Also, the employment of AI-chip ensures a cost-efficient performance. Moreover, introducing blockchain technology in the system further achieves a secure and transparent service at the expense of an acceptable memory and latency cost.

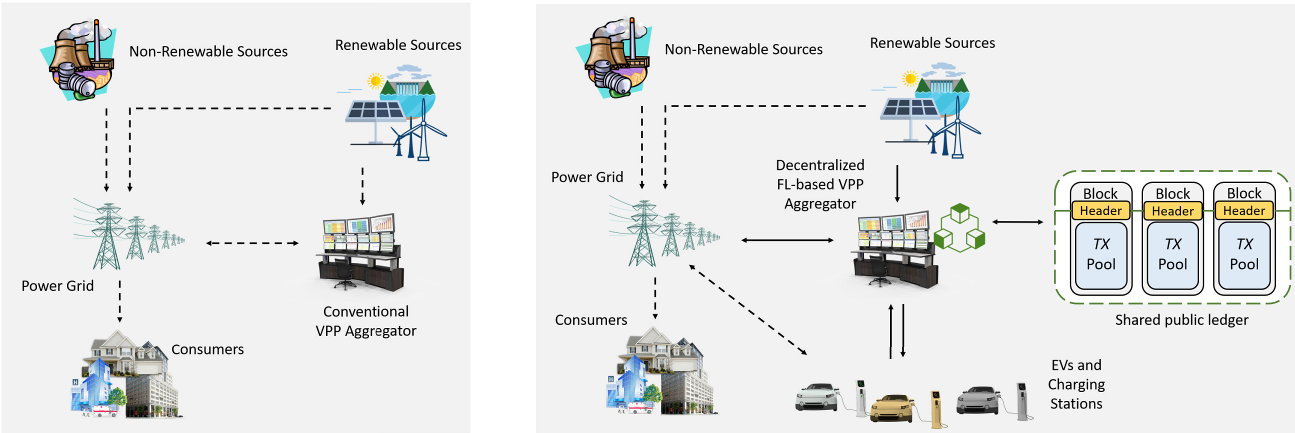
INDEX TERMS AI-enabled, blockchain-based, EVs, power-management, AI-chip, virtual power plant.

I. INTRODUCTION

In recent years, the utilization of renewable resources in the energy matrix has been increasing. At the end of 2019, global renewable generation capacity reached 2536 gigawatts [1]. Variants of renewable resource providers, e.g., wind power [2], photovoltaic power [3], and hydropower [4], serve as the power suppliers and transmit electrical energy from the generating sites to a power grid [5]. The power grid then distributes electric power to all the consumers, including residential areas, hospitals, commercial areas, administrative areas, and Electric Vehicle (EV) fleets. Aiming to achieve efficient dispatch and utilization of renewables, the Virtual Power Plant (VPP) has been proposed to act as an intermediary between distributed energy resources (DERs), power grid, controllable loads, and EVs [6]–[9].

Many VPP projects have been proposed over the last decade [10]–[13]. The current VPP demonstrations target efficient integration and distribution of resources. Nevertheless, they also need to consider the potential security risk of the communication between the aggregator, power grid, and consumers. In addition, the VPP offers demand-side management technology to energy consumers, which helps achieve intelligent storage and consumption on the client-side.

The associate editor coordinating the review of this manuscript and approving it for publication was Lin Zhang.



(a) (b)

FIGURE 1. Virtual Power Plant (VPP): (a) Conventional approach, (b) Proposed method, including the AEBIS system. Compared to the conventional VPP demonstration, the proposed approach deploys EV fleets as energy consumers and suppliers. An AI-enabled battery power consumption system is used for timely energy management, which has not been developed in the current V2G network. A blockchain network is integrated to replace the conventional VPP aggregator, ensuring a more robust and cost-efficient environment for collaborative training.

The efficient utilization of electricity remains a challenge in conventional VPP demonstrations. There have been many studies on the optimal supply and demand-side management of DERs. For the supply side, authors in [14]–[19] studied the optimal strategy against the inherent unpredictability of renewables, while there was a lack of discussion on the integration of the power consumers. For the demand side, the efficient management of consumers, which involves EVs in the vehicle-to-grid (V2G) network, was proposed in [20]–[26]. Such strategies only tailor the vehicles in the parking area to the VPP aggregator. A considerable number of moving cars, which experience power consumption fluctuation, have not been well addressed.

Artificial intelligence (AI) has proven to be effective in numerous applications in VPP studies. Economic dispatch and strategic bidding were investigated in EV and electricity markets [27]–[30]. Works in [31]–[34] employed deep learning techniques for energy generation and consumption forecast. In [35]–[39], intelligent integrated approaches were proposed for efficient demand response. However, the conventional aggregator in these approaches is equipped with a multi-GPU cluster, which requires high-power consumption and long-term maintenance [31], [35]–[37]. Also, there is limited computing capability on local devices [38], [39], which remains a bottleneck for high-speed training. Although it is possible to have a custom-built system for local computing, it comes at a high expense and is not portable.

Cyber-physical security is another concern in VPP systems [40]–[45]. One primary focus is on the vulnerabilities of conventional centralized control algorithms in smart grids [40]–[42]. With the increasing number of distributed energy resources integrated into the power system, researchers have progressed to the exploration of the robust distributed scheme against cyber-attacks [43]–[45]. However, the conventional aggregator in the VPP is still susceptible to malicious attacks, through which information can easily be tampered with. Also, data leakage could happen during the transmission of raw data.

To the best of our knowledge, none of the previous works considered the participation of EVs with electricity consumption forecasting, efficient computation for local devices, and secure communication between VPP aggregator and EV nodes simultaneously. In this work, we propose an AI-enabled blockchain-based electric-vehicle integration system for power management in smart grid platforms to solve the challenges mentioned above. First, we present a neural-network-based EV charge prediction system for power management in VPP. The learning process is based on federated learning (FL) technology [46], which enforces raw data protection and improves communication efficiency. We then establish a novel communication mechanism between the aggregator and each EV node, in which an AI system based on reconfigurable hardware (FPGA) is used to predict the amount of available electricity an EV could supply when idle to mitigate storage during peak load. The reconfigurable AI system, with high-speed computation and low-power consumption, can be packaged into an extended electronic control unit (ECU) connected to the controller area network (CAN) bus of a car [47], [48], as shown in Fig. 2. To elevate the security level, we further integrate blockchain technology [49] into the system.

A. CONTRIBUTIONS

The main contributions of this work are summarized as follows:

- An AI-enabled electric vehicle integration system based on an artificial neural-network (AI-Chip accelerator) and federated learning approach for EV charge prediction, where the EV fleet is employed as a consumer and as a supplier of electrical energy in VPP. The AI-Chip is

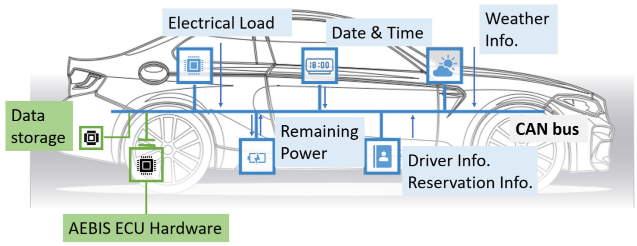


FIGURE 2. The integration of the proposed AEBIS system into the built-in Controller Area Network (CAN) of Electrical Vehicles (EVs). A CAN bus is a robust vehicle interconnect standard allowing microcontrollers and devices to communicate with each other. Each blue box indicates a built-in electronic controller unit (ECU), which shares with other ECUs its data via the CAN bus. The green box on the left shows a customized ECU for data storage, collecting and processing the data from other ECUs. The data storage ECU then transmits the data to the AEBIS ECU hardware for training and inference.

prototyped on FPGA and can be packaged in the CAN bus.

- A novel algorithm of data exchange between the power grid and EV fleet for electrical supply. Whenever the power grid needs electricity and requests vehicular networks, the amount of electrical supply from each EV can be calculated based on its extra electricity and driving status.
- A fully-decentralized architecture based on the blockchain technology to robustly consolidate all the distributed nodes and form a substantial smart power-storage facility.

The rest of this paper is organized as follows. In Section II, we discuss the related works on optimal operations, AI deployments, and security strategies in VPP. In Section III, we present the proposed AEBIS system. Section IV gives the performance evaluation of the proposed system. Section V highlights and discusses issues, that also need to given attention, and in Section VI, a conclusion of the paper is presented. The nomenclature used in this paper are listed in Table 1.

II. RELATED WORK

In this section, we survey related works on VPP demonstrations, focusing mainly on conventional optimal strategy, AI deployment and security in VPP.

A. OPTIMAL OPERATIONS IN VPP

Distributed energy resources and variants of consumer participation are increasingly integrated into current VPP platforms. The fluctuation of resource generation and unpredictable electricity consumption presents a challenge of power balance and economic benefit in VPP. Therefore, the related studies focused on optimal operations in VPP together with efficient integration of DERs and participation of end facilities.

TABLE 1. Nomenclature.

Abbreviation	Meaning
AI	Artificial Intelligence
BC	Blockchain
BRAM	Block Random-Access Memory
CAN	Controller Area Network
DER	Distributed Energy Resource
DSP	Digital Signal Processing
ECP	Expected Consumed Power
ECU	Electronic Control Unit
EEL	Extra Electrical Load
EV	Electric Vehicle
FF	Flip Flop
FL	Federated Learning
FPGA	Field Programmable Gate Array
ID	Identification
IID	Independent and Identically Distributed
LUT	Lookup Table
OP_RETURN	Return Operator
RL	Reinforcement Learning
RNN	Recurrent Neural Network
RP	Remaining Power
SoC	State of Charge
TX	Transaction
VPP	Virtual Power Plant
V2G	Vehicle-to-Grid

Authors in [14] developed an optimal control and bidding strategy for VPP with renewable energy generations and inelastic demand, in which the problem is formulated as a bilevel stochastic optimization. In [15], a quantile regression forest model was applied to the forecast of wind and photovoltaic energy production. In [16]–[19], the information gap decision theory was used to study the uncertainty of wind energy integrated with electricity and natural gas systems. While these studies mainly focused on the power generation and electricity markets, the participation of end consumers was barely investigated.

Some works addressed the importance of participation of EV fleets [20]–[26]. In [20], the optimal operations were proposed for the participation of EV aggregator in day-ahead energy and regulation markets. In [21], the authors proposed optimal scheduling algorithms for V2G energy sales and multiple ancillary services. Authors in [22] investigated the trade-off between energy and reserve markets and proposed an optimal operation for uncertain EV battery degradations. In [23], a look-ahead power scheduling algorithm, was proposed to manage the revenue risk of EV aggregations against fluctuating energy generation and electricity price. However, these studies barely stressed the practical power consumption of EV, which could be forecasted given the static and dynamic information (e.g., the behavior of drivers, duration of use, and weather condition). In [25], [26], a solution to preference quantification based on unknown EV types was studied. However, since the aggregator has to wait for interaction until a set of EVs arrive at the parking area rather than predict the power consumption of the EVs in advance, there exists an unavoidable delay of energy trading, which also affects the utilization of EVs in car-sharing markets [50].

B. AI-BASED VPP ARCHITECTURE

Recent years have witnessed the continuous rise in AI technologies in various VPP applications. The works in [27], [28] approached the economic dispatch using reinforcement learning (RL) or non-dominated sorting genetic algorithm. Variants of intelligent energy management methods have also been proposed based on RL [35]–[37], [51], [52] and recurrent neural network (RNN) [31], [32]. The works in [33] employed explainable AI tools and artificial neural networks for solar photovoltaic power prediction, while [34] proposed an ensemble learning-based model for wind energy prediction. In [53], demand-side energy management with a price forecast was proposed based on a multilayer perceptron. Authors in [29] integrated RL method for EV bidding strategy. A novel centralized energy demand learning algorithm for EV energy demand prediction was introduced in [39]. Considering that most of these works carry out the experiments on a single centralized server, the system faces the following problems unavoidably. First, a latency and cost bottleneck could appear when the center collects all the distributed data and performs learning. Second, the stability of the entire system greatly relies on the centralized server. That is, once the server breaks down, the requests from all distributed nodes will no longer be responding.

Moreover, once attackers access the centralized server, the private data is easily fetched or modified. On the contrary, in some edge computing paradigms [39], [51], the computation is moved from data centers towards the local devices. Nevertheless, there remain limitations in the storage and speed of the edge nodes.

C. SECURE VPP ARCHITECTURE

The security in both centralized and distributed VPP has been investigated using different approaches [25], [26], [40]–[45]. In the centralized architecture of the conventional VPP platform [40]–[42] as shown in Fig. 1(a), there remains two key issues of the security and stability of the VPP system. First, the main server is still prone to data leakage. In addition, the stability of the system extremely relies on the main server. That is, if the central database becomes corrupt, the entire system will face a big challenge. For the robust distributed schemes proposed in [43]–[45], the agents were restricted to communicate only with their neighbors. The communication activity is limited, and thus the global optimization is hard to achieve. Authors in [25], [26] proposed a consortium blockchain-based secure energy trading mechanism for EV-based distributed resources, where the preference of an EV towards discharging is quantified by the available energy and expected traveling distance in the next future. In this scenario, the blockchain network records only the electricity transaction; however, the communication security between the aggregator and EV node was not considered.

III. AEBIS SYSTEM

This section presents a detailed description of the proposed AI-Enabled Blockchain-based Electric Vehicle Integration System (AEBIS) for power management in the smart grid platform.

Fig. 1(b) depicts the overall architecture. The conventional aggregator is replaced with a decentralized FL-based virtual power-aggregator. The power grid collects and stores the electrical energy (renewable and non-renewable) and delivers it to the consumers afterward. When the provided renewable resource becomes insufficient, the power company will inform the power grid to supply extra power through traditional resources. Together with charging stations, the EV fleets offer a solution to provide energy back to the power grid. The interaction between the proposed aggregator, power grid, renewable energy providers, and vehicular network aims to utilize energy and secure communication efficiently. A blockchain network is used to implement the decentralized VPP aggregator. In this work, we mainly focus on two main key issues: 1) the communication between the decentralized VPP aggregator and EV fleets, whereby the FL training for battery charge prediction is performed, 2) introducing a blockchain network that replaces the conventional VPP aggregator.

In Fig. 1(b), the solid lines indicate the communication on the blockchain. The dashed lines indicate the rest of the communication activities. For each EV on the client-side, an AI system based on reconfigurable hardware (FPGA) is used to compute the expected power consumption during the next period. Therefore, given the current remaining power of an EV and the power grid's request, we can calculate the amount of electricity an EV should supply (discharge). When there is insufficient energy on the power grid, the communication between the aggregator and clients will help determine the amount of electrical supply.

A. EV BATTERY POWER CONSUMPTION PREDICTION

The expected remaining electricity is required from each EV node for the system's accurate and timely energy management. In practice, the power consumption of an EV is determined by the following factors: (1) weather conditions, (2) characteristics of the vehicle, (3) driving style, and (4) geography [54]. Given all the input features, the expected power consumption is predicted via a fully-connected neural network. The detailed experiment of the power consumption prediction will be introduced in Section IV.

B. EV BATTERY CHARGE MECHANISM

A battery should be charged when there is no reservation or no request from the power grid to provide the electricity back. Therefore, the main task is to predict the amount of electrical supply (discharge) from the EV fleet to the power grid. Fig. 3 describes the proposed algorithm for calculating the energy that each EV should provide back to the power grid when needed.

At first, the current remaining power for each vehicle is calculated given the maximum battery capacity and its state of charge (SoC) beforehand. After that, we compare the current remaining power with the expected power consumption,

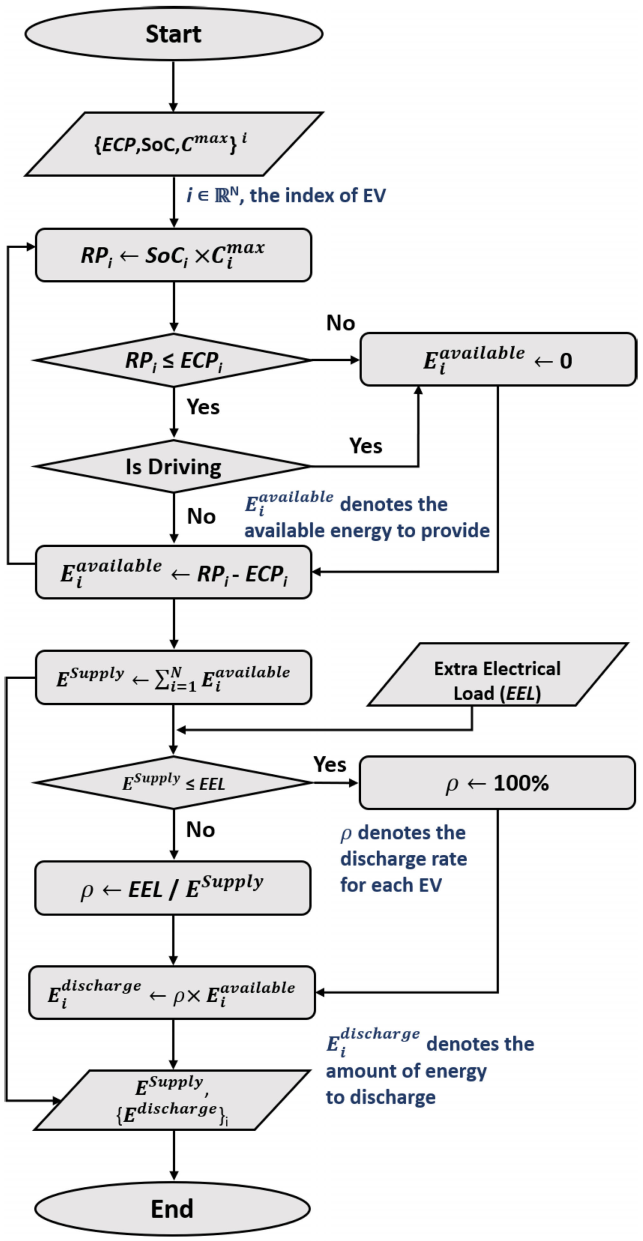


FIGURE 3. Flowchart for calculating the electrical supply from the EVs to power grid. In stage I, the available electricity of the EVs is output. In Stage II, the electrical supply from each EV is then calculated.

which is acquired in Section III-A. If the remaining power is less than the expected consumption or the EV is on the way, it cannot supply the electricity at the given moment. Thus, $E^{available}$, which denotes the maximum electricity the EV can provide, is set to zero. Nevertheless, the remaining power is useful information for the next driver to make a reservation. In contrast, if the vehicle is parked in the charging station and the energy will remain until it is consumed during the next period, the available energy is calculated as:

$$E^{available} = RP - ECP \tag{1}$$

where *RP* denotes the remaining power, and *ECP* denotes the expected consumed power.

At some point, we have the information of the available energy of each EV in the EV fleet, which is referred to as $\{E_i^{available}\}_{i\in\mathbb{R}^N}$, where *i* denotes the identification (ID) of the vehicle, and *N* is the number of EVs. The total power that the EV fleet can supply is simply described as the summation of $\{E_i^{available}\}_{i\in\mathbb{R}^N}$:

$$E^{supply} = \sum_{i=1}^{N} E_i^{available} \tag{2}$$

Following that, a decision rule is needed to decide the amount of electricity the power grid should request. A parameter, ρ, is used to denote the discharge rate for each EV. When there is an extra electrical load (EEL) on the power grid's side, a request to the EV fleet is made. If $E^{supply} \leqslant EEL$, then all the remaining power is required as the countermeasure against the power shortage, in which case the discharge rate ρ is set to 100%. If $E^{supply} > EEL$, it means the available power from the EV fleet is sufficient for electrical supply, and the vehicles do not need to supply 100% of their remaining electricity. The proportion of the supply will be:

$$\rho = EEL/E^{supply} \tag{3}$$

After that, each EV's amount of electrical discharge is the multiplication of the discharge rate and the available power:

$$E_i^{discharge} = \rho \times E_i^{available}. \tag{4}$$

C. FL-BASED FRAMEWORK

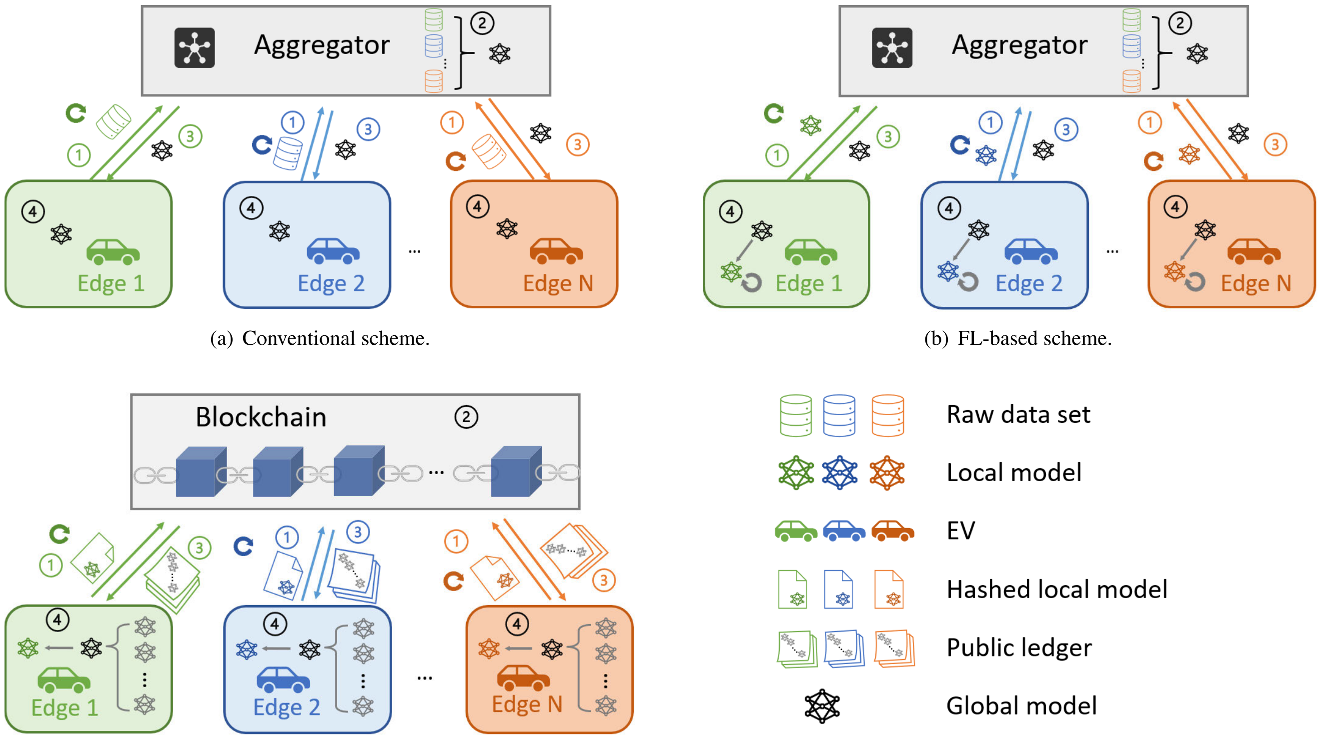
In the conventional neural network model, the aggregator collects the uploaded data sets from the EV nodes and then performs the training task as illustrated in Fig. 4(a). However, the frequent raw data exchange leads to high communication overhead and potential data leakage. The federated learning (FL) architecture offers a solution to this issue, in which each EV node shares and improves the local model via the aggregator, as shown in Fig. 4(b). The whole procedure is comprised of multiple training rounds until convergence. Each round mainly consists of the following four steps:

1) At first, each EV node trains its local model using the collected data set. In each local model, the gradient ∇g_L is calculated using adaptive moment estimation (Adam) optimizer [55] as shown by the following formula:

$$\nabla g_L = \frac{\delta E(W)}{\delta W} \tag{5}$$

where W denotes a set of weights, and $E(W)$ denotes the loss function with respect to W. $E(W)$ is used for measuring the model error and finding an optimal solution. Also, δ indicates partial derivatives.

2) Each client uploads the local gradients to the aggregator.



(a) Conventional scheme.

(b) FL-based scheme.

(c) Decentralized FL-based scheme

FIGURE 4. Comparison of conventional scheme, FL-based scheme and the proposed decentralized FL-based scheme. In the conventional scheme, each client uploads the data set to the aggregator and then downloads the trained model after the training process. In the FL-based scheme, the aggregator collects the uploaded models instead of the raw data set and then performs the local models' aggregation. In comparison with these two frameworks, the decentralized scheme's entire process is broken down into four main steps: (1) The hashed local model is uploaded to the blockchain network. (2) In the blockchain network, the entire process involves broadcasting, verification, mining, etc., after which the distributed ledgers are generated, (3) Each edge node receives a corresponding ledger with a set of hashed models, which is used to ask Swarm platform for accessing the models stored by other nodes. (4) As soon as an edge node collects the models, the local model will be updated. The whole process for each node is repeated until local convergence is obtained.

3) The aggregator collects the group of the local gradients and then outputs a global gradient ∇g_L:

$$\nabla g_G = \frac{1}{n} \sum_{i=1}^{n} \nabla g_L^i \tag{6}$$

4) Once the edge nodes receive the global gradient from server site, they update the parameters as follows:

$$W^{r+1} = W^r - \eta \nabla g_G \tag{7}$$
$$b^{r+1} = b^r - \eta \nabla g_G \tag{8}$$

where W^r and b^r denote the weights and biases in the r_{th} training round, respectively. η denotes the learning rate.

D. SECURE DECENTRALIZED FL-BASED FRAMEWORK

As illustrated in Fig. 4(c), in the decentralized architecture, there is no interaction with the conventional aggregator. Each node has the same public ledger in the blockchain network that records all the trained local models stored in the transactions. However, considering that the size of the model can sometimes be large and thus lead to a significant workload on the blockchain, we use the Swarm – a distributed storage platform [56] to store the models. In this way, only the model's address is uploaded on the blockchain, giving us a more efficient system. The whole procedure is summarized in Alg. 1.

1) BLOCKCHAIN AND SWARM PLATFORM

To realize the blockchain network, we begin with creating user accounts on Ethernet. We then initialize the nodes on Swarm, which is used to allocate memory for each client to store the data, as shown in Fig. 5. In the Swarm cluster,

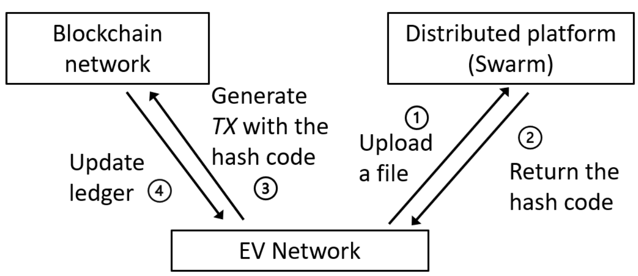


FIGURE 5. The communication between the blockchain network and the Swarm platform. The distributed storage platform is used to record user information and the updated files. *TX*: Transaction.

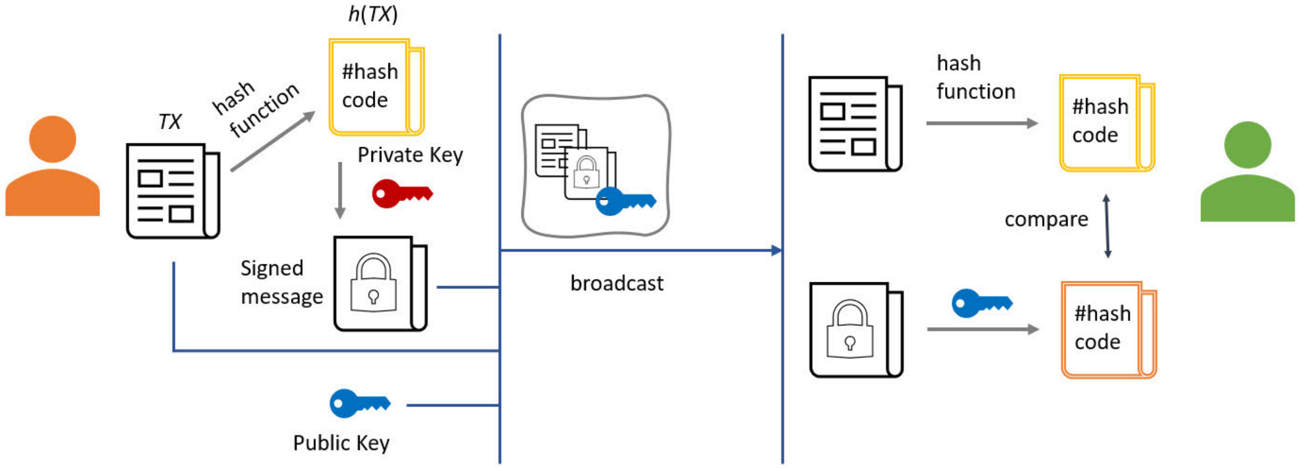


FIGURE 6. The process of verification using digital signatures. The original transaction is fed to a hash function, which is then encrypted using the signer's private key. The signed message, the original message, and the signer's public key will be broadcasted in the network. A receiver will decode the signed message using the public key. By comparing the result with the original message's hash value, the receiver is informed of any tampering.

Algorithm 1 Decentralized FL-Based Learning Scheme

Require: real-time data collected from EV
Ensure: Predicted Power Consumption

Initialize the aggregator to wait for collecting local models when $r = 0$
while $r \leq T$ **and** do not converge **do**
  Train local model in edge AI system
  $\nabla g_L^i \leftarrow \frac{\delta E(W_i)}{\delta W_i}$ /* Calculate local gradient
  Upload ∇g_L^i from AI system to Swarm for storing, obtain hashed value $h_i(addr)$ with respect to the address of stored model
  Record each collection of $h_i(addr)$, its signed message and a pubilc key of node as a whole transaction from node i to j, which is denoted by TX_{ij}
  Broadcast a request to the whole blockchain network
  As soon as TX_{ij} is verified, it will be added to the mempool
  All transactions in a mempool is packaged and then added to the block
  Mining begins. A successful mined block is added to the public ledger
  Participants download local models from the public ledger to update the global model as $\nabla g_G \leftarrow \frac{1}{n}\sum_{i=1}^{n} \nabla g_L^i$
  $W^{r+1} \leftarrow W^r - \eta \nabla g_G$ /* Update local model
  $b^{r+1} = b^r - \eta \nabla g_G$ /* Update local model
end while
For each EV, the predicted power consumption is output

each client is allocated discontinuous storage space. Every time a client uploads data, it will be partitioned into many segments, which are then stored in different volumes. After that, the cluster will generate a hash code representing the address that corresponds to the collection of data fragments. The complete data is accessible given the information of the user account and this hash code. In this work, the data refers to the training model containing parameters of the network.

At the blockchain side, when a client, C_1, uploads a file to Swarm and obtains the encrypted hash code, it will send the hash value to another node, C_2, through which a new transaction is automatically generated. However, to ensure that this transaction is trusted, the digital signatures are adopted for verification. Fig. 6 illustrates the verification procedure. At first, the client feeds the transaction data to the hash function and generates the hash of data. After that, the hash value is fed to the signature algorithm with the client's private key, whereby an encrypted, signed message is produced. After that, the new transaction will be broadcasted to all nodes, which contains the original information of the transaction (the signed message and the public key in regard to C_1). Each receiver can thus perform an easy operation of verification. First, one will use the same hash function and generate the hash value of the original message. Since the hash mapping always results in the same output, this value is unique and should be identical to those produced by C_1. The signed message will then be decrypted using the public key, through which the resulting value should match the previous hash value. If the decrypted hash matches the re-computed hash given the same data, the digital signature is proven to be valid. Therefore, this transaction is considered trusted and will be added to each node's transaction pool. Otherwise, the different values of the hash will reveal that the message has been tampered with. In this case, the message will be rejected by the receivers.

The transaction pool is where all the valid transactions wait to be confirmed by the blockchain network. However, with the increase of unconfirmed transactions, memory consumption and computational efficiency become challenging. To tackle this problem, Merkle tree [57] is introduced to significantly reduce the requirement concerning both memory

TABLE 2. Comparison among centralized (conventional and FL-based) and decentralized (blockchain-based) system.

	Conventional Method	Federated Learning	Federated Learning with Blockchain
Third Party Involvement	Yes	Yes	No
Data Management	Between server and clients	Kept by clients	Kept by clients
Safety	More prone to hacking and data leakage	Prone to hacking, safe data storage	Less prone to hacking, safe data storage
Stability	Low	Medium	High
System Complexity	Low	Medium	High
Consumption of Time and Energy	High	Low	Medium-high (depends on difficulty of mining)

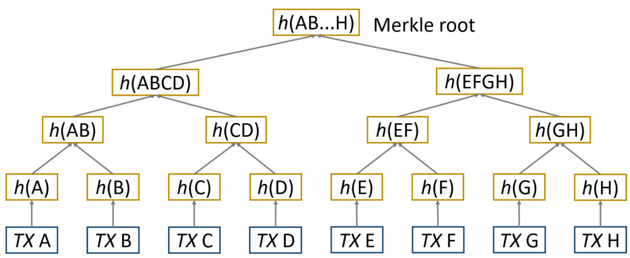


FIGURE 7. Illustration of Merkle tree structure with eight nodes. Each node represents a transaction that is fed into a hash function, and the hashed transaction is denoted by h(A) to h(H) in the figure. Then a bottom-to-up operation is performed, in which $h(AB) = h(h(A)) + h(h(B))$, $h(CD) = h(h(C)) + h(h(D))$, etc. The operation ends up with a single hash value, which is referred to Merkle root. In this case, the Merkle root is h(ABCDEFGH).

and computation as shown in Fig. 7. Given a sequence of transaction $TX_1, TX_2, \ldots, TX_n$, each of them is hashed to form a leaf node of the Merkle tree. The collection of these leaf nodes are denoted by $h(TX_i)_{i\in n}$. Following that, a binary implementation is used to merge every two nodes into a new one which belongs to the next layer as described by equation 9.

$$\begin{aligned} h(TX_1 + TX_2) &= h(H(TX_1) + h(TX_2)) \\ h(TX_3 + TX_4) &= h(H(TX_3) + h(TX_4)) \\ &\cdots \\ h(TX_{n-1} + TX_n) &= h(H(TX_{n-1}) + h(TX_n)) \end{aligned} \tag{9}$$

If n is odd, then $h(TX_n)$ is added to the next layer without binary operation. Recursively, each pair of new nodes in the next layer are hashed until the root node is reached, which is a single hash value of all the nodes below it.

The entire process of building a Merkle tree will result in a single hash value referred to as the Merkle root. The block header consists of a 32-Byte previous block hash, 32-Byte Merkle root, 4-Byte timestamp, 4-Byte difficulty target, and 4-Byte nonce. We denote the set of metadata except for nonce by M. Given a pre-determined value n; the target is to find a nonce that satisfies the requirement shown in equation 10.

$$Hash(M + nonce) = \underbrace{0\ldots0}_{n\ bits}x\ldots x \tag{10}$$

Once a perfect nonce is found, it will be added to the hashed block. The block header will be rehashed along with the successful nonce, then the block, including header and body, will be added to the chain. It is worth noting that, in our case, a relatively high frequency of information exchange among ledgers is required. Therefore, n is chosen to be small so as to smoothen the way for mining. With the public information in the blockchain, each node can quickly get the data that are used to access Swarm. Thus they can download the up-to-date models and update their own. A comparison among the conventional centralized model, federated learning model, and federated learning with blockchain architecture is summarized in Table 2.

2) WRITING METADATA IN TRANSACTIONS

In general, the return operator (OP_RETURN), which is part of the Bitcoin script language, is used to allow saving metadata on the blockchain [58]. However, the limit of storing data in an OP_RETURN has a maximum of 83 bytes according to release 0.12.0 [58]. This reveals a key advantage of using Swarm for data storage. It is also comparatively short and saves time for writing metadata in a transaction. Each time a model is stored in Swarm, it produces a hash value with a fixed length of 32 Bytes regardless of the size of the model. Therefore, it can always be written in a single OP_RETURN. Next, we implement the communication between blockchain and client, where the trained model is uploaded. For the fully-connected network in our experiment, which has 11 input neurons, two hidden layers (8 and 6 neurons, respectively), and one output, the total amount of parameters is the sum of the number of weights and biases, which is $11 \times 8 + 8 \times 6 + 6 \times 1 + (8 + 6 + 1) = 157$. Each parameter in the floating-number format occupies 4 Bytes; therefore, when we extract only the parameters from the model, the data size is $157 \times 4 = 628$ Bytes. For each model, at least eight transactions are required. With an enormous model size, the increased amount of transactions leads to a considerable deterioration of efficiency regarding storage and computation. An alternative way of data storage is to utilize distributed cloud storage instead of Swarm. However, it is still at high risk of data leakage while benefiting from the apparent simplicity. A comparison of the above methods is summarized in Table 3.

IV. EVALUATION

A. EVALUATION METHODOLOGY

The data set for the power consumption prediction contains the features of weather, geography, and user information as explained in the previous section. We collected the weather

TABLE 3. Comparison of three data storage methods on the blockchain.

	Blockchain	Blockchain + Swarm	Blockchain + Cloud
Data Storage	Model on blockchain	Hash on blockchain , model on Swarm	Hash on blockchain, model on Cloud
Safety	Less prone to hacking, safe data leakage	Less prone to hacking, safe data leakage	More prone to hacking and data leakage
Ease of Use	Medium	Not easy to use	Easy to use
System Load on Blockchain	High	Low	Low

TABLE 4. Data set for vehicle energy consumption.

Input Feature	Value	Unit and Datatype
Start Time	0 to 23	Int
Duration of Use	0 to 24	Hours, Float
Weekday	1 to 7	Mon. to Sun., Int
Temperature	-11.61 to 33.83	$^\circ C$, Float
Rainfall	0 to 19.04	mm, Float
Humidity	0.07 to 1	$\%$, Float
Wind Speed	0.24 to 23.45	m/s, Float
Latitude	35.15 to 37.29	m/s, Float
Longitude	139.09 to 139.76	m/s, Float
Gender	0 or 1	Male/Female, Int
Age	21 to 69	Years old, Int
Output	**Value**	**Unit and Datatype**
Power Consumption	0 to 140	kWh, Float

information from January 2020 to July 2020 in Fukushima, Kanagawa, and Tokyo area in Japan [59]. The starting time of the car reservation was set from 0:00 to 23:00, and the duration of use was set from 0 to 24 hours. We considered the driver's age range according to the requirements of Class 2 license [60]. The daily power consumption was measured, given the input features and the measurement model [61]. We summarize the detailed information of the data set in Table 4.

We considered the scenario where each client's data is independent and identically distributed (IID). We allocated the entire data set to three clients; each subset contains 1000 samples following a similar distribution. However, in most practical cases, the local data on each EV node is usually non-IID, which comes from the fact that the data is collected at a different time or by other drivers. Therefore, we investigated the impact of non-IID data distributions on the performance. One interesting case is when each EV is reserved at different times of a day, i.e., morning, afternoon, evening, and night. We considered a set of four clients, each of whom is associated with the period ranging from 6:00 to 11:59, 12:00 to 17:59, 18:00 to 23:59, 0:00 to 5:59, respectively. Besides, we are interested in the scenario where the EVs are reserved by users in a different age range. Five clients are included in this case, each of whom is associated with the age ranging from 21 to 29, 30 to 39, 40 to 49, 50 to 59, 60 to 69, respectively. For each FL training, the simulation was repeated 50 times. We used the R^2 Score to measure the performance of the model.

For the blockchain solution, we used Geth [62] which is a Golang implementation of the Ethereum protocol. The Swarm platform was used for data storage and distribution. We conducted the experiments on Ubuntu 18.04.3. We also used Geth to set up a private Ethereum blockchain network, whereby a genesis file was created for each node. A genesis file contains the whole configuration of initial states, of which the information of several vital parameters is described in Table 5.

TABLE 5. Key configurations in a genesis file.

	Description
ChainID	ID of the chain (0 49344). It is unique for each chain
Difficulty	Set to a small value (0x20000) for the ease of mining
ParentHash	Hash of the previous block. For the genesis block, it is set to 0

B. EVALUATION RESULTS

1) CONVENTIONAL VS FL-BASED APPROACHES

For the first experiment, where each client's data is independent and identically distributed, the result is shown in Fig. 8. We observe that the performance of the FL approach is 0.922 in the R^2 Score on average, which is less than the conventional model (0.938). The slightly imbalanced data distribution explains the accuracy degradation.

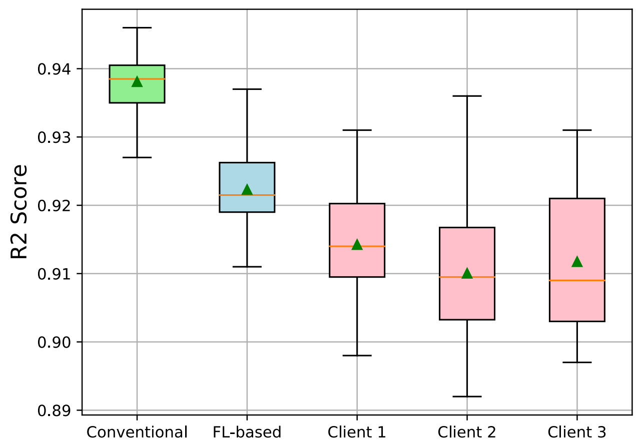


FIGURE 8. Comparison between the conventional model, individual learning model, and the FL-based model using IID data distribution.

Figures 9 and 10 show experiments of the FL model on non-IID data distributions. From this evaluation, we observe

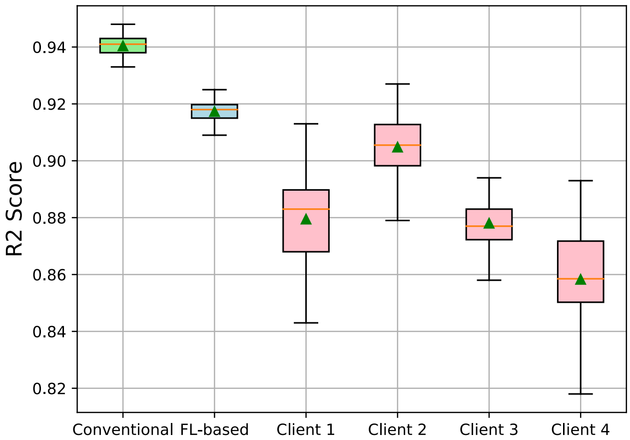


FIGURE 9. Comparison between the conventional model, individual learning model, and the FL-based model using non-IID data distribution.

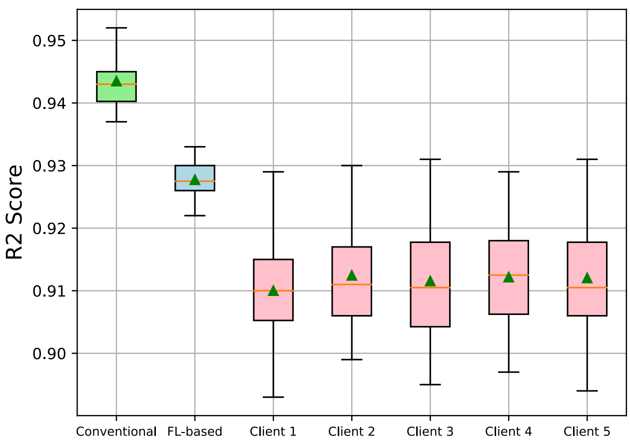


FIGURE 10. Comparison between the conventional model, individual learning model, and the FL-based model using non-IID data distribution. There are five clients in the experiment; each client is associated with a data set concerning ages ranging from 20 to 69. The FL-based model has proven to be robust in the non-IID setting.

that the single client's performance deteriorates significantly compared to the conventional model due to highly-skewed non-IID data. However, the FL-based model has proven to be robust in both cases. For the current EV sharing community, the accurate power forecasting allows drivers to know the remaining electricity of EVs in advance; thus, the reservation management becomes more efficient.

2) BLOCKCHAIN NETWORK ON SWARM PLATFORM

For each node, multiple accounts are created along with a pair of the private and public key of which the first account is for mining and is associated with the node's address. The address is derived as the last 20 bytes of the public key. We created two blockchain nodes, each containing two accounts. Each account address is used to create a node on Swarm. After that, a local directory is automatically generated with a specific Swarm ID. When both services on blockchain and Swarm have been established, the communication between clients, Swarm, and blockchain starts correctly (refer to III-D1).

TABLE 6. Hardware complexity of power consumption prediction system.

Name	BRAM_18K	DSP48E	FF	LUT
Expression	-	-	0	493
Instance	-	5	414	950
Memory	2	-	320	20
Multiplexer	-	-	-	627
Register	-	-	454	-
Total	2	5	1188	2090
Available	120	80	35200	17600
Utilization (%)	1	6	3	11
Weights	**Memory required**			
Weights	568 Bytes			
Biases	60 Bytes			
Inputs	44 Bytes			
Total	672 Bytes			

3) HARDWARE COMPLEXITY

Table 6 shows the hardware complexity of the fully connected network on the Zynq-7010 FPGA. The system utilized 3% of the FF, 11% of the LUT, 6% of the DSP48, and approximately 1% 18k BRAM. Table 6 also shows the memory cost for storing the weights, biases, and input features in the format of a single-precision floating-point. The required BRAM is 672 Bytes in total (568-Byte weights + 60-Byte biases + 44-Byte inputs). We conclude that the hardware complexity of the system is small as it occupies only a fraction of the available FPGA resources.

V. DISCUSSION

A review of three learning architectures is illustrated in Fig. 4. Compared to the conventional learning scheme that requires the raw data exchange between server and clients, the FL-based approach achieves excellent performance while significantly reducing the total training and communication time. The evaluation of hardware complexity offers a possibility to improve the time and cost-efficiency further. The FL-based model can satisfactorily address the issues due to non-IID distributed data and partial device participation. Besides, the mechanism where models are shared without revealing the raw data set is considered to protect user data privacy to some extent. The results indicate that our FL-based network could provide an accurate prediction for EV power consumption, which is necessary for either upcoming electricity supply or EV reservation. However, there are still existing potential pitfalls due to malicious clients and hijacking. Therefore, a trusted decentralized blockchain network is proposed to take the place of a conventional aggregator, guaranteeing the high-level security as well as the stability of the system. Nonetheless, for the decentralized architecture, there are several remaining challenges. With the increase of blocks, the system will suffer from performance deterioration due to enormous memory requirements and slow transaction or mining speed. Also, the very initial transactions, which contain the historical models, are permanently preserved in the network and becomes an extra load to the system.

VI. CONCLUSION

This paper proposed an AI-enabled blockchain-based electric vehicle integration scheme (AEBIS) for power management in the smart grid platform. The core software and hardware components of the proposed approach are presented in detail. We showed that a reliable prediction network for power consumption could help reduce energy monitoring and management delay. Also, accurate power forecasting provides the necessary information to the car-sharing market, making reservation management more efficient. While the FL-based framework achieves nearly equivalent performance compared to the conventional model, it protects the user's raw data and speeds up the learning stage. We carried out the training process on a customized AI processor with low power consumption and low latency. Moreover, it is more portable and cost-efficient. Besides, the simulation of a simple blockchain platform is conducted. In practice, the current blockchain network provides a secure and transparent service at the expense of memory and time cost. In our future work, the efficiency of deploying the blockchain network will be improved by investigating the blockchain protocol's storage and communication mechanism.

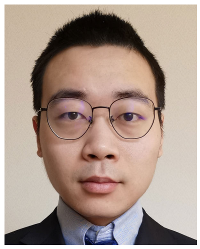
ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive comments, which helped us to improve the manuscript.

REFERENCES

[1] (2020). *Renewable Capacity Statistics*. [Online]. Available: https://www.irena.org/publications/2020/Mar/Renewable-Capacity-Statistics-2020
[2] Y. Lin, M. Yang, C. Wan, J. Wang, and Y. Song, "A multi-model combination approach for probabilistic wind power forecasting," *IEEE Trans. Sustain. Energy*, vol. 10, no. 1, pp. 226–237, Jan. 2019.
[3] H. D. Tafti, A. I. Maswood, G. Konstantinou, J. Pou, and F. Blaabjerg, "A general constant power generation algorithm for photovoltaic systems," *IEEE Trans. Power Electron.*, vol. 33, no. 5, pp. 4088–4101, May 2018.
[4] M. Chazarra, J. I. Perez-Diaz, and J. Garcia-Gonzalez, "Optimal joint energy and secondary regulation reserve hourly scheduling of variable speed pumped storage hydropower plants," *IEEE Trans. Power Syst.*, vol. 33, no. 1, pp. 103–115, Jan. 2018.
[5] U. K. Kalla, B. Singh, S. S. Murthy, C. Jain, and K. Kant, "Adaptive sliding mode control of standalone single-phase microgrid using hydro, wind, and solar PV array-based generation," *IEEE Trans. Smart Grid*, vol. 9, no. 6, pp. 6806–6814, Nov. 2018.
[6] D. Koraki and K. Strunz, "Wind and solar power integration in electricity markets and distribution networks through service-centric virtual power plants," *IEEE Trans. Power Syst.*, vol. 33, no. 1, pp. 473–485, Jan. 2018.
[7] N. Ruiz, I. Cobelo, and J. Oyarzabal, "A direct load control model for virtual power plant management," *IEEE Trans. Power Syst.*, vol. 24, no. 2, pp. 959–966, May 2009.
[8] E. Dall'Anese, S. S. Guggilam, A. Simonetto, Y. C. Chen, and S. V. Dhople, "Optimal regulation of virtual power plants," *IEEE Trans. Power Syst.*, vol. 33, no. 2, pp. 1868–1881, Mar. 2018.
[9] G. Zhang, C. Jiang, and X. Wang, "Comprehensive review on structure and operation of virtual power plant in electrical system," *IET Gener., Transmiss. Distribution*, vol. 13, no. 2, pp. 145–156, Jan. 2019.
[10] B. Coyne, "Limejump: Aggregators will need an electricity supply licence to survivel theenergyst. com," Tech. Rep., 2017.
[11] J. Yao, S. Yang, K. Wang, Z. Yang, and X. Song, "Concept and research framework of smart grid," source-grid-load" interactive operation and control," *Autom. Electr. Power Syst.*, vol. 36, no. 21, pp. 1–6, 2012.
[12] M. Wache and D. C. Murray, "Application of synchrophasor measurements for distribution networks," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, Jul. 2011, pp. 1–4.
[13] M. Hisada and A. B. Abdallah, "Virtual power platform control system," Japanese Patent 58 007, 2019.
[14] Q. Zhao, Y. Shen, and M. Li, "Control and bidding strategy for virtual power plants with renewable generation and inelastic demand in electricity markets," *IEEE Trans. Sustain. Energy*, vol. 7, no. 2, pp. 562–575, Apr. 2016.
[15] S. Camal, A. Michiorri, and G. Kariniotakis, "Optimal offer of automatic frequency restoration reserve from a combined PV/Wind virtual power plant," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6155–6170, Nov. 2018.
[16] M. A. Mirzaei, M. Nazari-Heris, K. Zare, B. Mohammadi-Ivatloo, M. Marzband, S. Asadi, and A. Anvari-Moghaddam, "Evaluating the impact of multi-carrier energy storage systems in optimal operation of integrated electricity, gas and district heating networks," *Appl. Thermal Eng.*, vol. 176, Jul. 2020, Art. no. 115413.
[17] M. A. Mirzaei, M. Hemmati, K. Zare, M. Abapour, B. Mohammadi-Ivatloo, M. Marzband, and A. Anvari-Moghaddam, "A novel hybrid two-stage framework for flexible bidding strategy of reconfigurable micro-grid in day-ahead and real-time markets," *Int. J. Electr. Power Energy Syst.*, vol. 123, Dec. 2020, Art. no. 106293.
[18] M. A. Mirzaei, A. Sadeghi-Yazdankhah, B. Mohammadi-Ivatloo, M. Marzband, M. Shafie-khah, and J. P. S. Catalão, "Integration of emerging resources in IGDT-based robust scheduling of combined power and natural gas systems considering flexible ramping products," *Energy*, vol. 189, Dec. 2019, Art. no. 116195.
[19] M. Jadidbonab, B. Mohammadi-Ivatloo, M. Marzband, and P. Siano, "Short-term self-scheduling of virtual energy hub plant within thermal energy market," *IEEE Trans. Ind. Electron.*, early access, Mar. 11, 2020, doi: 10.1109/TIE.2020.2978707.
[20] M. Gonzalez Vaya and G. Andersson, "Self scheduling of plug-in electric vehicle aggregator to provide balancing services for wind power," *IEEE Trans. Sustain. Energy*, vol. 7, no. 2, pp. 886–899, Apr. 2016.
[21] A. Baringo, L. Baringo, and J. M. Arroyo, "Day-ahead self-scheduling of a virtual power plant in energy and reserve electricity markets under uncertainty," *IEEE Trans. Power Syst.*, vol. 34, no. 3, pp. 1881–1894, May 2019.
[22] M. R. Sarker, Y. Dvorkin, and M. A. Ortega-Vazquez, "Optimal participation of an electric vehicle aggregator in day-ahead energy and reserve markets," *IEEE Trans. Power Syst.*, vol. 31, no. 5, pp. 3506–3515, Sep. 2016.
[23] M. Shamshirband, J. Salehi, and F. S. Gazijahani, "Look-ahead risk-averse power scheduling of heterogeneous electric vehicles aggregations enabling V2G and G2 V systems based on information gap decision theory," *Electr. Power Syst. Res.*, vol. 173, pp. 56–70, Aug. 2019.
[24] W. Wang, P. Chen, D. Zeng, and J. Liu, "Electric vehicle fleet integration in a virtual power plant with large-scale wind power," *IEEE Trans. Ind. Appl.*, vol. 56, no. 5, pp. 5924–5931, Sep. 2020.
[25] Z. Zhou, B. Wang, M. Dong, and K. Ota, "Secure and efficient vehicle-to-grid energy trading in cyber physical systems: Integration of blockchain and edge computing," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 1, pp. 43–57, Jan. 2020.
[26] Z. Zhou, B. Wang, Y. Guo, and Y. Zhang, "Blockchain and computational intelligence inspired incentive-compatible demand response in Internet of electric vehicles," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 3, no. 3, pp. 205–216, Jun. 2019.
[27] L. Lin, X. Guan, Y. Peng, N. Wang, S. Maharjan, and T. Ohtsuki, "Deep reinforcement learning for economic dispatch of virtual power plant in Internet of energy," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6288–6301, Jul. 2020.
[28] F. Li, J. Qin, and W. X. Zheng, "Distributed Q-learning-based online optimization algorithm for unit commitment and dispatch in smart grid," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 4146–4156, Sep. 2020.
[29] S. Najafi, M. Shafie-khah, P. Siano, W. Wei, and J. P. S. Catalão, "Reinforcement learning method for plug-in electric vehicle bidding," *IET Smart Grid*, vol. 2, no. 4, pp. 529–536, Dec. 2019.
[30] Y. Ye, D. Qiu, M. Sun, D. Papadaskalopoulos, and G. Strbac, "Deep reinforcement learning for strategic bidding in electricity markets," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1343–1355, Mar. 2020.
[31] H. Shi, M. Xu, and R. Li, "Deep learning for household load forecasting—A novel pooling deep RNN," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 5271–5280, Sep. 2018.

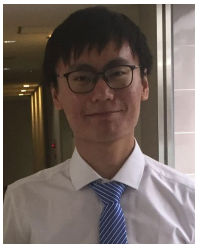
[32] A. Rosato, M. Panella, R. Araneo, and A. Andreotti, "A neural network based prediction system of distributed generation for the management of microgrids," *IEEE Trans. Ind. Appl.*, vol. 55, no. 6, pp. 7092–7102, Nov. 2019.
[33] M. Kuzlu, U. Cali, V. Sharma, and O. Guler, "Gaining insight into solar photovoltaic power generation forecasting utilizing explainable artificial intelligence tools," *IEEE Access*, vol. 8, pp. 187814–187823, 2020.
[34] J. Lee, W. Wang, F. Harrou, and Y. Sun, "Wind power prediction using ensemble learning-based models," *IEEE Access*, vol. 8, pp. 61517–61527, 2020.
[35] X. Xu, Y. Jia, Y. Xu, Z. Xu, S. Chai, and C. S. Lai, "A multi-agent reinforcement learning-based data-driven method for home energy management," *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 3201–3211, Jul. 2020.
[36] R. Lu, S. H. Hong, and M. Yu, "Demand response for home energy management using reinforcement learning and artificial neural network," *IEEE Trans. Smart Grid*, vol. 10, no. 6, pp. 6629–6639, Nov. 2019.
[37] Z. Wan, H. Li, H. He, and D. Prokhorov, "Model-free real-time EV charging scheduling based on deep reinforcement learning," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5246–5257, Sep. 2019.
[38] *Toshiba Energy Systems Solutions Corporation*. [Online]. Available: https://www.toshiba-energy.com/en/renewable-energy/product/vpp.htm
[39] 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 *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.
[40] S. Fan, Q. Ai, and L. Piao, "Fuzzy day-ahead scheduling of virtual power plant with optimal confidence level," *IET Gener., Transmiss. Distrib.*, vol. 10, no. 1, pp. 205–212, Jan. 2016.
[41] G. N. Ericsson, "Cyber security and power system communication—Essential parts of a smart grid infrastructure," *IEEE Trans. Power Del.*, vol. 25, no. 3, pp. 1501–1507, Jul. 2010.
[42] H. Khurana, M. Hadley, N. Lu, and D. A. Frincke, "Smart-grid security issues," *IEEE Secur. Privacy Mag.*, vol. 8, no. 1, pp. 81–85, Jan. 2010.
[43] P. Li, Y. Liu, H. Xin, and X. Jiang, "A robust distributed economic dispatch strategy of virtual power plant under cyber-attacks," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4343–4352, Oct. 2018.
[44] Y. Liu, H. Xin, Z. Qu, and D. Gan, "An attack-resilient cooperative control strategy of multiple distributed generators in distribution networks," *IEEE Trans. Smart Grid*, vol. 7, no. 6, pp. 2923–2932, Nov. 2016.
[45] W. Zeng, Y. Zhang, and M.-Y. Chow, "Resilient distributed energy management subject to unexpected misbehaving generation units," *IEEE Trans. Ind. Informat.*, vol. 13, no. 1, pp. 208–216, Feb. 2017.
[46] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol. (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.
[47] T. H. Vu, O. M. Ikechukwu, and A. Ben Abdallah, "Fault-tolerant spike routing algorithm and architecture for three dimensional NoC-based neuromorphic systems," *IEEE Access*, vol. 7, pp. 90436–90452, 2019.
[48] T. H. Vu, Y. Okuyama, and A. B. Abdallah, "Comprehensive analytic performance assessment and K-means based multicast routing algorithm and architecture for 3D-NoC of spiking neurons," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 15, no. 4, pp. 1–28, Dec. 2019.
[49] S. Nakamoto. (Mar. 2009). *Bitcoin: A Peer-to-Peer Electronic Cash System*. Cryptography Mailing list. [Online]. Available: https://metzdowd.com
[50] *Japan Car Sharing Association*. [Online]. Available: https://www.japan-csa.org/
[51] F. L. D. Silva, C. E. H. Nishida, D. M. Roijers, and A. H. R. Costa, "Coordination of electric vehicle charging through multiagent reinforcement learning," *IEEE Trans. Smart Grid*, vol. 11, no. 3, pp. 2347–2356, May 2020.
[52] Y. Du and F. Li, "Intelligent multi-microgrid energy management based on deep neural network and model-free reinforcement learning," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1066–1076, Mar. 2020.
[53] C. Zhang, R. Li, H. Shi, and F. Li, "Deep learning for day-ahead electricity price forecasting," *IET Smart Grid*, vol. 3, no. 4, pp. 462–469, Aug. 2020.
[54] C. De Cauwer, W. Verbeke, T. Coosemans, S. Faid, and J. Van Mierlo, "A data-driven method for energy consumption prediction and energy-efficient routing of electric vehicles in real-world conditions," *Energies*, vol. 10, no. 5, p. 608, May 2017.
[55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[56] *Swarm*. [Online]. Available: https://swarm.ethereum.org/
[57] H. Li, R. Lu, L. Zhou, B. Yang, and X. Shen, "An efficient Merkle-Tree-Based authentication scheme for smart grid," *IEEE Syst. J.*, vol. 8, no. 2, pp. 655–663, Jun. 2014.
[58] (2016). *Bitcoin Release 0.12.0*. Accessed: Jul. 15, 2020. [Online]. Available: https://bitcoin.org/en/release/v0.12.0
[59] *Japan Meteorological Agency*. [Online]. Available: http://www.data.jma.go.jp/gmd/risk/obsdl/index.php
[60] *My License*. [Online]. Available: https://www.mylicense.co.jp/search/license/whatnormal2.php
[61] *REXEV*. [Online]. Available: https://rexev.co.jp/
[62] *Go Ethereum*. [Online]. Available: https://geth.ethereum.org/



ZHISHANG WANG received the B.S. degree in computer science from Wuhan University, China, in 2014, and the M.S. degree in computer science from the University of Freiburg, Germany, in 2019. He is currently pursuing the Ph.D. degree with the Adaptive Systems Laboratory (ASL), The University of Aizu. He is also a member of the ASL. His current research interests include machine learning systems, collaborative learning, blockchain, and trustworthy AI. He is also interested in event-driven neuromorphic systems targeted for a new generation of brain-inspired computing technologies and adaptive edge computing systems.

MARK OGBODO (Student Member, IEEE) received the master's degree in computer science from the African University of Science and Technology (AUST). He is currently pursuing the Ph.D. degree with the Adaptive Systems Laboratory (ASL), The University of Aizu. He is also a member of the ASL. His current research interests include neuromorphic and machine learning systems.

HUAKUN HUANG received the bachelor's and master's degrees from Guangzhou University, Guangzhou, China, in 2014 and 2016, respectively, and the Ph.D. degree in computer science and engineering from The University of Aizu, Japan, in 2019. He is currently a Research Fellow with The University of Aizu. His research interests include blockchain, the Internet of Things (IoTs), intelligent edge computing, signal processing, and machine learning.



CHEN QIU received the B.S. and M.S. degrees in computer science from The University of Aizu, Japan, in 2018 and 2020, respectively. He is currently pursuing the Ph.D. degree with the Information Security Laboratory, The University of Aizu. He is also a member of the Information Security Laboratory. His current research interests include blockchain and machine learning and privacy-preserving.

MASAYUKI HISADA (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from The University of Aizu, Japan, in 2002. From 2002 to 2003, he was a Director with the Network Security Technology Japan, Inc. From 2003 to 2006, he was an Assistant Professor with the Kanazawa Institute of Technology, Japan. From 2007 to 2018, he was the President and the CEO of the Aizu Laboratory, Inc., Japan. He is currently the Director and the CEO of Aizu Computer Science Laboratories, Inc., Japan. His research interests include cloud systems, IoT, and blockchain field. Business development focused in advanced computer science and engineering, industrial utilization in the power trading and autonomous driving fields.

ABDERAZEK BEN ABDALLAH (Senior Member, IEEE) received the Ph.D. degree in computer engineering from The University of Electro-Communications, Tokyo, in 2002. He has been a Faculty Member with The University of Aizu, since 2007. Before joining The University of Aizu, he was a Research Associate with the Graduate School of Information Systems, The University of Electro-Communications, from 2002 to 2007. He is currently a Full Professor of computer science and engineering and the Head of the Division of Computer Engineering, The University of Aizu. He has authored three books, published more than 150 journal articles and conference papers in these areas, and given invited talks as well as courses at several universities. He has been a PI or a CoPI of several projects for developing next generation high-performance reliable computing systems for applications in general purpose and pervasive computing. His research interest includes computer system and architecture, with an emphasis on adaptive/self-organizing systems, networks-onchip/SoCs, processor micro-architecture, and power and reliability-aware architectures. He is also interested in neuro-inspired systems and VLSI design for 3D-ICs. He is a Senior Member of ACM and a member of IEICE.