Trustworthy AI-Enabled System and Algorithms for Power-Management in Network of Electric Vehicles

Zhishang Wang

A DISSERTATION
SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN COMPUTER SCIENCE AND ENGINEERING

Graduate Department of Computer and Information Systems
The University of Aizu
2023

The thesis titled

Trustworthy AI-Enabled System and Algorithms for Power-Management in Network of Electric Vehicles

by

Zhishang Wang

is reviewed and approved by:

Chief referee
Professor
Abderazek Ben Abdallah, The University of Aizu, Aizuwakamatsu

Professor
Junji Kitamichi, The University of Aizu, Aizuwakamatsu

Professor
Hiroshi Saito, The University of Aizu, Aizuwakamatsu

Associate Professor
Yuichi Okuyama, The University of Aizu, Aizuwakamatsu

Professor
Shinji Yokogawa, The University of Electro-Communications, Tokyo

The University of Aizu
2023

Contents

List of Figures

List of Tables

List of Abbreviations

Acronym

AC	Alternating Current
AEBIS	AI-Enabled Blockchain-based Electric Vehicle Integration System
AI	Artificial Intelligence
BC	Blockchain
BRAM	Block Random-Access Memory
CAN	Controller Area Network
DC	Direct Current
DER	Distributed Energy Resource
DSP	Digital Signal Processing
ECP	Expected Consumed Power
ECU	Electronic Control Unit
EEL	Extra Electrical Load
EV	Electric Vehicle
FedAvg	Federated Average
FedCS	Federated Client Selection
FF	Flip Flop
FL	Federated Learning
FL-QLMS	Federated Learning for Qualified Local Model Selection
FPGA	Field Programmable Gate Array
ID	Identification
IID	Independent and Identically Distributed
LUT	Lookup Table
NoEV	Network of Electric Vehicles
Non-IID	Non-Independently and Identically Distributed
oVML	on-Vehicle Machine Learning
OP_RETURN	Return Operator
PCP	Power Consumption Prediction
RL	Reinforcement Learning
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
RP	Remaining Power
SoC	State of Charge
TX	Transaction
VPP	Virtual Power Plant
V2G	Vehicle-to-Grid

Symbol

$\{c_i\}_{i \in N}$	A group of energy consumers
$\{ev_j\}_{j \in M}$	A group of EVs
$\{Lat_k\}_{k \in K}$	Latitude of all cities
$\{Long_k\}_{k \in K}$	Longitude of all cities
$\{Weather_{k,t}\}_{k \in K, t \in T}$	Weather information
∇g_G	Global gradient
∇g_L	Local gradient
b^r	Biases in the r-th training round
B_{delay}	Block propagation delay
B_{main}	Number of blocks included in the main chain
B_{size}	Block size
B_{stale}	Blocks that were successfully mined but not inclued in the current best chain
B_{total}	Total amount of blocks generated
$City_d$	Destination city
$City\_ID$	List of city IDs
$City_s$	Start city
C_i	Local client i
D	Data set
DI	Diversity between model parameters
D_{iid}	Local I.I.D data set
D_{local}	Size of the local data set
$D_{non-iid}$	Local non-I.I.D data set
DoD	Duration of driving
D_{train}	Training data
$E(W)$	Loss function with respect to set of weights
ED	Euclidean distance
ED_{min}	Minimum Euclidean distance
H	Hash function
k	Fraction of hacked clients
K	Set of city IDs
Lat_c	Latitude of the calculated
Lat_d	Latitude of the destination city
Lat_p	Latitude of the practical city
Lat_s	Latitude of the start city
$Long_c$	Longitude of the calculated point
$Long_d$	Longitude of the destination city
$Long_p$	Longitude of the practical city
$Long_s$	Longitude of the start city
M	Model for neural network
M_{aux}	Auxiliary model
M_{local}	Local model
$M_{selected}$	List of selected models

Symbol

n	Pre-determined value
N	Number of clients, number of models
$N_{0...62}$	Nodes in the blockchain
$N_{selected}$	Number of selected models
N_{total}	Total number of cities
P_a, P_b	Parameters of model a and b
P_{aux}	Parameters of the auxiliary model
P_i	Parameters of the local model i
P^j	A single parameter of parameter vector P
PC_{pred}	Predicted power consumption
r_s	Stale block rate
S	Sample for the neural network model
t	Time
t^{alloc}	Time cost for allocating discharge tasks
t^c	Time cost for receiving energy demand
t^d	Delay in discharging the battery
t^{notif}	Time cost for notifying discharge tasks
t^o	Time cost for transmitting energy offers
t^o_{out}	Timeout for transmitting energy offers
t^{org}	Time cost for organizing energy demand
t^p	Time cost for predicting power consumption
t^r	Time cost for responding to the virtual power plant
$t_i^{g\_c}$	Time cost for transmitting energy from the grid to the consumer
$t_j^{v\_g}$	Time cost for transmitting energy from the vehicle to the grid
t_s	Start time
T	Time period of the weather data
T_1	Time cost for receiving and organizing energy demand
T_2	Time cost for notifying energy requests
T_3	Time cost for predicting available energy and replying of energy offers
T_4	Time cost for allocating and notifying discharge tasks
T_5	Time cost for transmitting energy
T_{res}	Response time
T_{size}	Transaction size
TI	Time interval
$User\_Info$	User information
W	Set of weights
W^{local}	Local weights
W^r	Weights in the r-th training round
α	Fraction for the number of selected models
η	Learning rate
σ^2	Variance

To my wife, my son, my parents and the rest of my family.

Acknowledgments

I would like to express my deepest thanks and gratitude to Prof. Abderazek Ben Abdallah for giving me the opportunity to work on this thesis, for his support, encouragement and guidance, for his good ideas and good criticism, for his patience. I would also like to thank Prof. Junji Kitamichi, Prof. Hiroshi Saito, Prof. Yuichi Okuyama of The University of Aizu and Prof. Shinji Yokogawa of The University of Electro-Communications for taking the time to revise my thesis.

I would like to thank all the members of Ben Abdallah & Dang Laboratory and my friends at the University of Aizu. Their supportive words and encouraging messages have motivated me to work even harder and become a better researcher and person. I would also like to thank the staff of The University of Aizu for their assistance.

My infinite love goes to my parents and family who always provide me with endless support and unconditional love. Last but not least, my heartfelt gratitude goes to my beloved, stunning wife Shiyin. She has always stood by my side and gifted me with her endless love and support.

Zhishang Wang,
February 2023,
Aizuwakamatsu, Japan

Thesis advisor: Professor Abderazek Ben Abdallah Zhishang Wang

Trustworthy AI-Enabled System and Algorithms for Power-Management in Network of Electric Vehicles

Abstract

A virtual power plant (VPP) is a network of distributed power generation units, flexible power consumers, and storage systems that balances load on a power grid by allocating power generated by various interconnected units during periods of peak demand. However, the fluctuation of energy generated by renewable resources makes balancing the energy supply a challenge. Demand-side energy devices such as electric vehicles (EVs) and mobile robots can also balance energy supply and demand if used effectively. With the innovation of bidirectional charging technology, EVs have become not only energy consumers but also energy suppliers.

Efficient energy management between the smart grid and EVs requires a charging mechanism that controls the charging/discharging process of the vehicles. Another challenge is to accurately and quickly predict the energy consumption of the electric vehicles. State-of-the-art research addresses the problem of prediction in vehicular networks using a collaborative learning approach based on neural networks. The prevailing approach is the combination of federated learning and blockchain technology, but it faces the following problems. First, current federated learning approaches pay little attention to attack scenarios. The assumption that a malicious model can be uploaded in any training round leads to a significant degradation in model accuracy. Besides, the constant selection of new models for the blockchain solution leads to a heavy load on the network. The efficiency of the blockchain suffers greatly from this problem, making it challenging to apply in real-world scenarios.

In this dissertation, we propose a trustworthy AI-enabled system and algorithms for power management in network of electric vehicles. We summarize the work in four main contributions.

First, a novel EV charging mechanism is proposed, in which an AI system based on reconfigurable hardware (FPGA) is used to predict the amount of available energy that 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.

Second, a multi-stage power consumption prediction method is proposed based on a fully-connected neural network model. The performance of the prediction demonstrates that the algorithm is accuracy and suitable for both intra and inter-district travel.

Third, a robust federated learning for qualified learning model selection (FL-QLMS) is proposed against malicious data and model attacks. The FL-QLMS performs at each training round that selects a group of best models and filters out the disqualified models.

Fourth, a fully-decentralized and a semi-decentralized blockchain-based collaborative learning are proposed respectively. In the fully-decentralized architecture, the network is formed by a group of EVs, and

a Swarm platform is introduced to store the local models in a secure way. In the semi-decentralized architecture, a VPP aggregator and a group of EVs are integrated together, where the local models are transmitted off-chain from EVs to the aggregator and only the global models are stored in the blockchain. Both proposals provide a highly secure solution while significantly increasing the efficiency of the blockchain network.

The proposed system and algorithms were evaluated with a driving data set and a blockchain simulator. The results demonstrate that the power consumption of EVs can be predicted in an accuracy, efficient, and secure manner, therefore the proposal is a promising countermeasure against peak demand. Besides, the proposed collaborative learning scheme has great potential to be applied in various research fields.

電気自動車のネットワークにおける電力管理のための信頼性の高い AI を用いたシステムとアルゴリズム

概要

仮想発電所（VPP）は、分散型発電装置、柔軟な電力消費装置、蓄電システムからなるネットワークで、需要のピーク時に相互接続されたさまざまな装置で発電した電力を配分し、電力網の負荷を調整するものである。しかし、再生可能資源で発電されたエネルギーは変動するため、エネルギー供給のバランスをとることが課題となっている。電気自動車（EV）や移動ロボットなどの需要側エネルギー機器は、有効に活用することでエネルギー需給を調整できる。このように、双方向充電技術の革新によって、EV はエネルギー消費者だけでなく、エネルギー供給者にもなっている。

スマートグリッドと EV の間で効率的なエネルギーマネジメントを行うためには、EV の充放電プロセスを制御する充電メカニズムが必要であり、電気自動車のエネルギー消費量を正確かつ迅速に予測することは考慮すべき課題となっている。近年、ニューラルネットワークに基づく協調学習アプローチを用いて、車両ネットワークにおけるエネルギー予測の問題に取り組む研究が盛んに行われている。有力なアプローチとして、連合学習とブロックチェーンの組み合わせが挙げられるが、以下のような問題に直面している。まず、現在の連合学習アプローチは、攻撃シナリオにほとんど注意を払っていないため、どの学習ラウンドでも悪意のあるモデルがアップロードされる可能性があるという仮定は、モデルの精度を大きく低下させることに繋がる。また、ブロックチェーンのために常に新しいモデルを選択することは、ネットワークへの大きな負荷となる。このような問題によりブロックチェーンの効率は大きく損なわれ、現実世界のシナリオに適用させることは困難になっている。

本論文では、電気自動車のネットワークにおける、電力管理のための信頼性の高い AI システムとアルゴリズムを提案する。提案される主要な 4 つの貢献の概要は以下の通りである。

まず、再構成可能なハードウェア（FPGA）に基づく AI システムを用いて、アイドル時に EV が供給可能なエネルギー量を予測し、ピークロード時の蓄電を軽減する新しい EV 充電メカニズムを提案する。この再構成可能な AI システムは、高速演算と低消費電力により、自動車の CAN バスに接続された拡張電子制御ユニット（ECU）に搭載することが可能である。

第二に、完全連結型ニューラルネットワークモデルに基づく多段階消費電力予測手法を提案する。予測の性能は、このアルゴリズムが正確であり、地区内および地区間の移動に適していることを実証している。

第三に、悪意のあるデータやモデルへの攻撃に対して頑健な連合学習におけるモデル選択法 Federated Learning for Qualified Learning Model Selection (FL-QLMS) を提案する。

FL-QLMS は、学習ラウンドごとに最適なモデル群を選択し、悪意のあるモデルをフィルタリングする。

第四に、完全分散型と半中央集権型の 2 つのブロックチェーンベースの協調学習をそれぞれ提案する。完全分散型では、EV のグループがネットワークを形成し、Swarm プラットフォームを導入して、ローカルモデルをブロックチェーンに安全に保存する。半中央集権型では、VPP アグリゲータと EV のグループを統合し、ローカルモデルを EV からアグリゲータにオフチェーンで転送することで、グローバルモデルのみをブロックチェーンに保存する。どちらのアルゴリズムも、ブロックチェーンネットワークの効率を大幅に向上させ、高い安全性を実現する。

走行データセットとブロックチェーンシミュレーターによって上記のシステムとアルゴリズムを評価した。その結果、EV の消費電力を正確かつ効率的に、そして安全に予測できることが示され、したがって、本提案はピーク需要への対策として有望であることが証明された。また、提案した協調学習方式は、様々な研究分野で応用できる可能性がある。

1

Introduction

1.1 Virtual Power Plant with Integration of Electric Vehicles

In recent years, the utilization of renewable resources has increased in the energy matrix. At the end of 2021, the global renewable electricity generation capacity reached 3068 gigawatts [1]. Meanwhile, European emission standards limit carbon dioxide emissions from regular cars to less than 95 g/km by 2020 [2]. Variants of renewable resource providers, e.g., wind power [3], photovoltaic [4], and hydroelectric [5], serve as power suppliers, directing electrical energy from generation sites to a power grid [6, 7]. The power grid then distributes electrical energy to all consumers, including residential areas, hospitals, commercial areas, administrative areas, and electric vehicle (EV) fleets. To achieve efficient distribution and utilization of renewable energy, the virtual power plant (VPP) was proposed to act as an intermediary between distributed

energy resources (DERs), the power grid, controllable loads, and EVs [8–11].

In the last decade, many VPP projects have been proposed [12–15]. Current VPP demonstrations aim to efficiently integrate and distribute resources. Nevertheless, they must also consider the potential security risk of communication between the aggregator, the power grid, and the consumers. In addition, the VPP provides energy consumers with demand-side management technology that contributes to smart storage and consumption on the customer side.

Efficient utilization of electricity remains a challenge in conventional VPP demonstrations. There have been many studies on the optimal supply-side and demand-side management of DERs. For the supply side, the authors in [16–21] studied the optimal strategy against the inherent unpredictability of renewable energy, while there was a lack of discussion on the integration of electricity consumers. For the demand side, efficient consumer management that incorporates EVs into the vehicle-to-grid (V2G) network was proposed in [22–28]. The integration of EVs provides a promising solution to peak demand, for the reason that bidirectional technology enables EVs to serve as both energy consumers and energy suppliers. The strong relationship between VPP and EVs raises the question of how to efficiently manage energy from electric vehicles.

Economic dispatch and strategic bidding have been studied in EV and electricity markets using artificial intelligence (AI) [29–32]. In [33–36], deep learning techniques were used to predict energy generation and consumption. In [37–39], intelligent integrated approaches for efficient demand-side management were proposed. However, the conventional aggregator in these approaches is equipped with a multi-GPU cluster, which requires high power consumption and long-term maintenance [33, 37–39]. Various efforts have been made to outsource edge computing tasks in vehicles [40, 41]. And a few studies have investigated the framework of vehicle edge computing for the VPP scenario [42–44]. For a complicated smart-vision task in a driving environment, vehicles must be equipped with high-speed systems that process a large amount of sensor data (about 1 Gb/s) [45]. However, the computing capacity of local devices is limited, which remains a bottleneck for high-speed training [46, 47]. Renesas Xtreme, the latest automotive microcontroller family, for example, includes devices with limited memory

ranging from 32K Flash/4K RAM to 8 Flash/512K RAM [48]. While it is possible to have a custom system for local computing, it is very expensive and not portable.

1.2 Secure Centralized Computing in Vehicle-to-Grid Networks

The security issue in centralized V2G networks has been studied using different approaches [49–54]. In the centralized architecture of the conventional VPP platform [49–51], as shown in Fig. 1.1(a), there are still two main problems for 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 is extremely dependent on the main server. That is, if the central database is corrupted, the entire system faces a major challenge. In the robust distributed systems proposed in [52–54], the agents were restricted to communicate only with their neighbors. The communication activity is limited, so global optimization is difficult to achieve One primary focus is on the vulnerabilities of conventional centralized control algorithms in smart grids [49–51]. With the increasing number of distributed energy resources integrated into the power system, researchers have advanced the research of the robust distributed system against cyber attacks [52–54]. However, the conventional aggregator in VPP is still vulnerable to malicious attacks that can easily manipulate information. In addition, data leakage may occur during the transmission of raw data.

1.3 Secure Decentralized Computing in Vehicle-to-Grid Networks

Security and privacy are other concerns in vehicular edge computing (VEC) which has great significance in avoiding traffic collisions, improving road efficiency, and reducing environmental impact [55]. As a concrete example, protecting range anxiety functionality is critical for EV drivers. In addition, a cyberattack on EV or charging stations can result in a large-scale charging outage that can have a significant impact on the vehicle and the power grid. Secure data sharing and management [56–58] has been studied, and various federated learning-based frameworks for vehicular networks [59, 60] have been proposed. Other privacy frameworks, such as differential privacy, attempt to deal with aggregation issues, however, with the challenge of achieving an optimal tradeoff between data utility and data leakage [61].

As a decentralized and secure framework, blockchain is a popular solution to replace the traditional approach in edge computing. It benefits federated learning in secure energy trading, management, and protection of EVs and driver interconnected data Secure bidirectional energy trading (charging and discharging) [62–65] for EVs has been investigated using a blockchain system. Research in [62, 66] examined both blockchain-based energy trading and data exchange in vehicle-to-grid (V2G) networks. Works in [67–69] proposed blockchain-based models for information authentication and trust management in a vehicular network. Other works proposed a variety of incentive-compatible schemes to encourage EV nodes to participate in demand response [70, 71]. While the above works addressed secure blockchain-based decentralized energy trading, EV participation, and data management issues in V2G, they did not specifically investigate secure data communication between the smart grid and the vehicular network. Moreover, the overall load on the network remains a significant challenge as the number of EVs continues to increase.

1.4 Power Management in Vehicle-to-Grid Networks: Problems and Motivation

To the best of our knowledge, none of the previous works have considered the participation of EVs with electricity consumption prediction, 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 system to predict the charge of electric vehicles for power management in VPP. The learning process is based on federated learning (FL) technology [72], which ensures the protection of raw data and improves communication efficiency. We then establish a novel communication mechanism between the aggregator and individual EV nodes using a reconfigurable hardware (FPGA)-based AI system to predict the amount of available electricity that an EV could supply during idling to mitigate storage during peak load. The reconfigurable AI system with high-speed computation and low power consumption can be packaged into an ex-

tended electronic control unit (ECU) connected to the controller area network (CAN) bus of a car [73, 74]. To increase the level of security, we further integrate blockchain technology [75] into the system.

However, the previous approach has the following shortcomings: (1) The constant choice of new models for the blockchain solution leads to a heavy load on the network. The efficiency of the blockchain suffers greatly from this problem, making it difficult to apply in real-world scenarios. (2) The system is only designed to predict power consumption for a local area, along with weather information at the start time. In a practical scenario where an electric vehicle travels to another city, the trained model cannot handle such a complicated case because the geographical and weather information changes during the journey. (3) State-of-the-art federated learning approaches pay little attention to attack scenarios. The assumption that a malicious model can be uploaded in any training round leads to a significant degradation of model accuracy.

1.5 Thesis Objectives and Contributions

Based on all the above facts, in this thesis, we propose a trustworthy AI-based system and algorithms for power management in a network of electric vehicles. First, we propose a novel communication mechanism between the aggregator and each EV node using a reconfigurable hardware (FPGA)-based AI system to predict the amount of available electricity that an EV could supply when idle to mitigate peak load storage. 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 [73, 74], as shown in Fig. 4.8 The proposed mechanism includes a new EV battery power consumption prediction algorithm based on a fully-connected neural network model. The performance of the prediction demonstrates that the algorithm is suitable for both intra-district and inter-district trips.

Second, to ensure learning of the model in an efficient and secure manner, we introduce a robust collaborative learning method that integrates federated learning and blockchain technol-

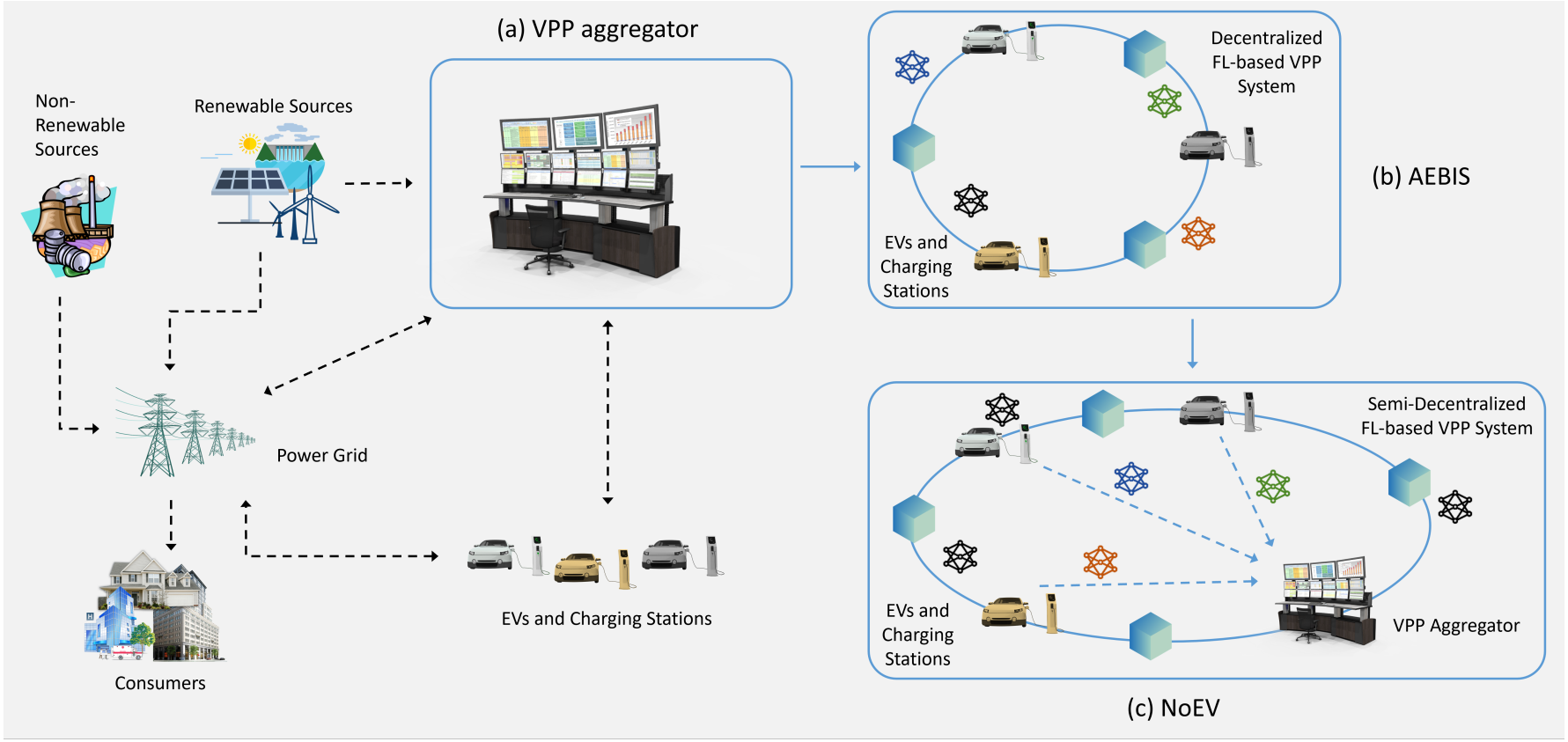


Figure 1.1: **Virtual Power Plant (VPP): (a) conventional VPP aggregator, (b) AEBIS, (c) NoEV. In the conventional VPP architecture, the EV fleet is generally considered as a type of end consumers. A VPP aggregator monitors activity on the vehicular network. In AEBIS, where each EV participates in FL by sharing local models via the blockchain, the EV fleets form a blockchain network and the VPP aggregator is thus replaced. Compared to AEBIS, NoEV introduces a combination of VPP aggregator and EVs. The aggregator first aggregates the local models of EVs and then uploads the global models to the blockchain. In the proposed system, a substantial number of local models are not stored in the blockchain, which ensures a more efficient environment for collaborative learning. The colored models represent local models and the black models represent global models.**

ogy. We proposed an algorithm called federated learning for qualified learning model selection (FL-QLMS) that is robust to both data and model attacks. Moreover, the novel blockchain architecture enables the entire system to maintain a high level of security while significantly increasing the efficiency of the blockchain network.

The main contributions of this research are as follows:

- A trustworthy network of electric vehicle (NoEV) system for power management in smart grid. The blockchain-enabled system is 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 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 multi-stage power consumption prediction method which ensures the accurate prediction performance for intra and inter-district travel.

- A fully-decentralized architecture based on the blockchain technology to robustly consolidate all the distributed nodes and form a substantial smart power-storage facility.

- A semi-decentralized collaborative learning scheme. The system maintains a high-security level while significantly increasing the efficiency of the blockchain network.

- A novel algorithm for robust federated learning, named federated learning for qualified local model selection (FL-QLMS).

1.6 Thesis Outline

The rest of the thesis is organized as follows:

- In Chapter 2, we first provide an overview of energy management in vehicle-to-grid networks. We then introduce the basic idea of federated learning and blockchain.
- In Chapter 3, we present related works on optimal operations in VPP, AI deployment in VPP, EV power consumption prediction, integration of blockchain and FL in vehicular networks, and client selection in federated learning.
- In Chapter 4, we present the proposed network of EV (NoEV) for power management in smart grid and the novel algorithm for power consumption prediction of EVs.
- In Chapter 5, we introduce the robust federated learning for qualified local model selection (FL-QLMS).
- Chapter 6 presents the proposed blockchain-enabled systems.
- Finally, in Chapter 7, we end this thesis with the conclusion and plan for future work.

2

Background

2.1 Virtual Power Plant

The influx of renewable energy sources in response to climate change and to protect the environment has led to a reduction in the use of traditional energy sources [76]. However, due to dependence on weather conditions, fluctuations in renewable energy sources remain a challenge in balancing the use of renewable and non-renewable energy sources as well as energy demand and supply. Therefore, there is a need to remotely coordinate and optimally and quickly control generation and storage systems. In addition, a platform is needed to respond quickly to energy demand from electricity users and consumers [77, 78].

As illustrated in Figure 2.1, a virtual power plant is a cloud-based control system that ag-

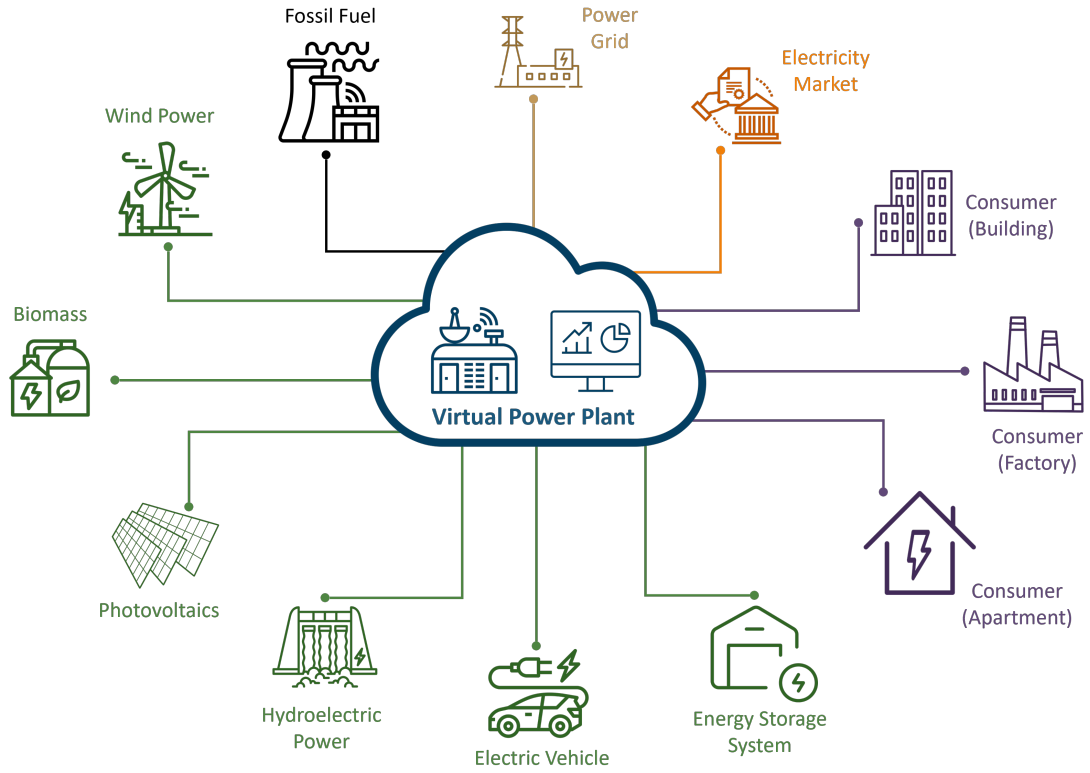


Figure 2.1: **An illustration of a virtual power plant (VPP). A VPP integrates power grid, electricity market, renewable and non-renewable resources, energy storage system, and energy consumers, etc.**

gregates various distributed energy resources (DERs) to provide reliable power. These energy resources include wind farms, photovoltaics, hydropower, heat pumps, and storage facilities such as electric vehicles. The VPP acts as an intermediary between DERs and the wholesale electricity market. It trades energy on behalf of DER owners who cannot participate in this market themselves, and provides greater efficiency and flexibility in energy distribution than traditional power plants. The main objective of the VPP is to manage electricity peaks by intelligently managing the generation capacity of individual units. The interconnected units are coordinated by the VPP control system but operate independently [79].

In addition, the VPP provides ancillary services to the grid operator to maintain grid stability [80, 81]. A successful VPP should consider several key factors in its ancillary services. (1) the generation capacity of the DER; (2) the consumption of energy consumers; (3) the geographical information of the power plant; (4) the status of large energy storage facilities such

as gas or heat storage (size capacity and electricity storage); and (5) the status of smaller energy storage facilities such as local houses and electric vehicles. When it receives signals from the grid operator, the VPP algorithm must be able to integrate different units and respond in a timely manner with precise control instructions and schedules.

Another important point of VPP is the business model [82, 83]. The main advantage of VPP is flexibility, which helps stablizing the grid. This flexibility comes from distributed energy producers who are paid to ensure the reliability of the energy flow. The VPP interacts with energy markers while hosting a variety of energy storage facilities that generate revenue by providing power through the ancillary services market.

In March 2011, Japan's Fukushima nuclear power plant was damaged by an earthquake and tsunami, causing widespread power outages. Automaker Nissan sent a fleet of first-generation LEAFs to the disaster area and started exploring how electric vehicles could be used to share their power [84]. Also, Mitsubishi provided 45 i-MiEV electric cars to assist rescue workers, transport relief supplies and provide heating [85]. This was also the launch of a new technology that allows electricity stored in batteries to be shared with buildings and homes. Vehicle-to-grid (V2G) technology allows stationary vehicles to be integrated into smart grid systems to sell electricity back to the grid at a higher price or at times of peak demand.

V2G technology, also known as bidirectional charging, not only draws power from the grid to charge the vehicle's battery, but also uses the energy from the vehicle's battery to supply power into the grid. Charging a conventional electric vehicle requires a one-way charger that converts AC power from the grid to DC power. With a bidirectional charger, the energy stored in the EV battery can be fed back into the grid by converting DC to AC. Vehicles with bidirectional charging capabilities allow users to store excess energy that can then be used to power their homes or sold back to the grid, demonstrably saving users money. A consortium including energy companies OVO Energy and Nissan conducted a three-year trial, installing more than three hundred bi-directional chargers in UK homes. Charging an electric car costs an average of just over 500 pounds per year, nearly 35% compared to the cost of gasoline or diesel. Charging an electric car at home is generally cheaper and sometimes free on campuses or workplaces [86].

2.2 AI-Enabled Vehicular Network

The emergence of artificial intelligence technology has replaced the traditional manual production model, and the accuracy and efficiency of this technology has made it a popular choice for all sectors of society [87]. In the automotive industry, the use of artificial intelligence technology has not only improved production efficiency, but also optimized the performance of all aspects of the vehicle, providing significant economic benefits to the automotive industry. AI can be quite beneficial, as shown in Fig. 2.2 at the following key points:

- Autonomous Driving
- Route Optimization
- Big Data in Internet of Vehicle
- Battery Maintenance and Charging

Self-driving cars have been around for decades in the 20th century and are showing a trend toward near-practicality at the beginning of the 21st century [88, 89]. Self-driving cars rely on artificial intelligence, visual computing, radar, surveillance devices, and global positioning systems working together to allow computers to control motor vehicles automatically and safely without human initiative. The intelligence of the car is expressed in the degree of separation between the car and the driver. The less the driver is involved in decision making while driving, the more intelligent the car. If a person does not need to be involved at all in the entire process of driving the car, then it can be considered that the car is truly driverless. The Society of Automotive Engineers has developed the classification standard for autonomous driving, which divides autonomous driving technology into six levels from L0-L5 [90]. Each level describes the extent to which a car takes over tasks and responsibilities from its driver, and how car and driver interact. Levels 0 through 5 are defined according to the relative degree of automation. Level 0, "No Automation," means that the driver controls the car without assistance from a driver assistance system. Level 1, "Driver Assistance," means that driver assistance systems support the driver but do not take control. In level 2, "Partly Automated Driving," the driver remains responsible

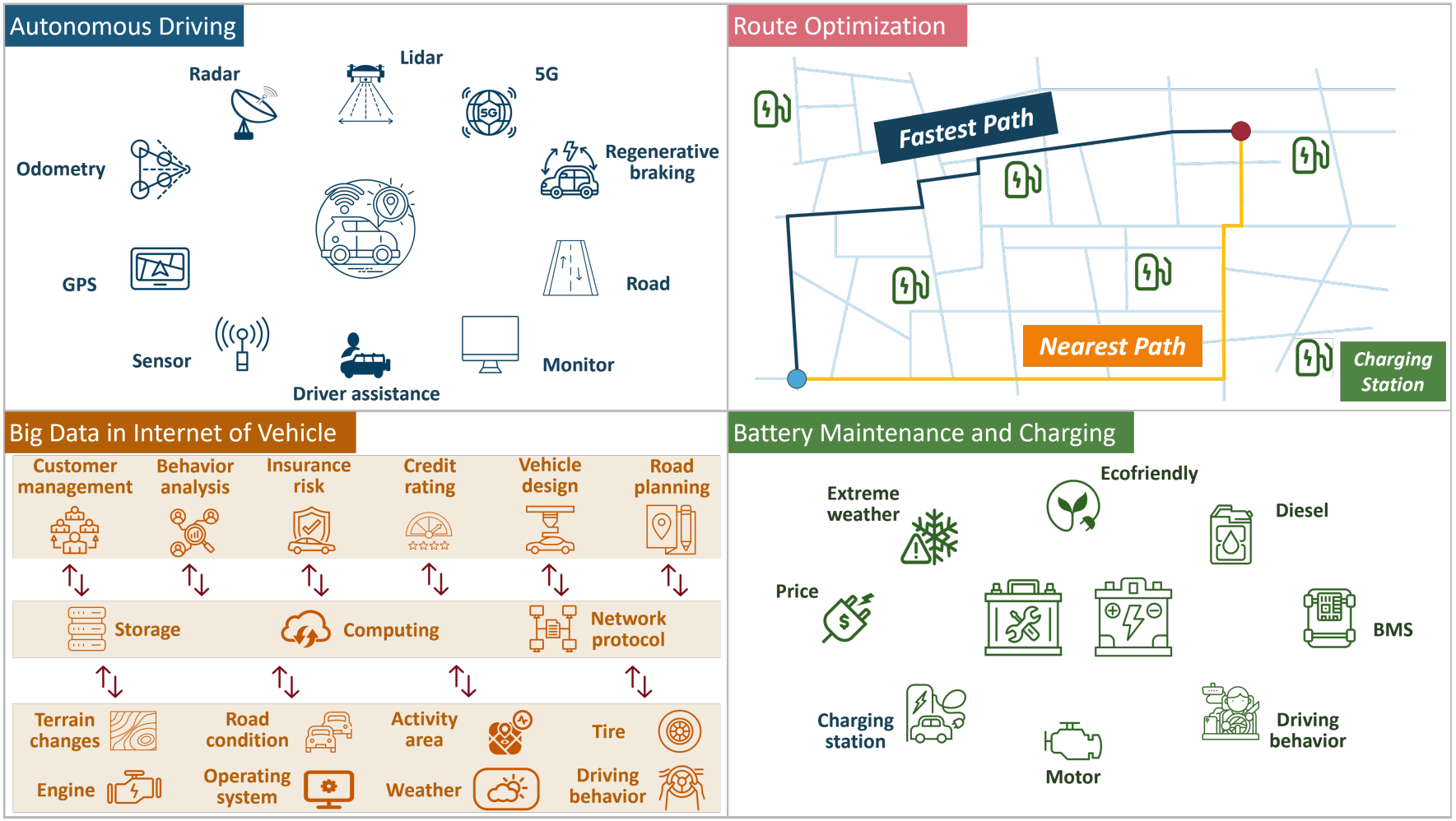


Figure 2.2: **The employment of AI in the internet of vehicles. Applications include autonomous driving, route optimization, bid data, and battery maintenance and charging.**

for operating the vehicle, while the systems can also take over control. Level 3, "Highly Automated Driving," level 4, "Fully Automated Driving," and level 5, "Full Automation" are still in the test phase. Level 1 driver assistance systems are most widespread today. As of March 2022, vehicles with level 3 and higher will make up only a marginal portion of the market. In March 2021, Honda became the first manufacturer to provide a legally approved level 3 vehicle, and Toyota operated a potential level 4 service around the Tokyo 2020 Olympic Village [91, 92]. Uber Eats and Nuro have signed a 10-year partnership to use autonomous vehicles for food delivery [93].

The route optimization problem was first proposed by Dantzig and Ramser [94]. The classical route optimization problem can be described as follows: There is a starting point and multiple customer points, the geographic location and demand of each point are known, subject to various constraints, how can an optimal route be planned so that it can serve each customer point and eventually return to the starting point. The vehicle routing problem plays a great role in practical applications in production and life, such as logistics and distribution, transportation planning, transportation network design, etc. For most gasoline-powered vehicles, conventional navigation algorithms ignore refuelling considerations because gas stations are usually readily available and refuelling times are generally short. Route optimization in electric vehicle networks is used to address mileage anxiety, the fear that a car will run out of power before it reaches a charging station [95]. This concern is so widespread that it is considered one of the barriers to widespread adoption of electric vehicles. Second, charging an electric vehicle's battery is an even more decision-intensive task because charging time can account for a significant portion of total travel time and can vary significantly depending on the charging station, vehicle type, and battery level [96]. In addition, charging times are nonlinear. For example, it takes longer to charge a battery from 80% to 90% than from 10% to 20%. Google has recently developed a routing algorithm that recommends charging stations to EV owners of electric vehicles based on their location, the remaining driving distance of the vehicle, and the plug type of the vehicle [97].

One of the problems in the internet of vehicles is the large amount of data that is transmitted,

including data from cars, roads, users, etc [98, 99]. On the one hand, the internet of vehicles processes the road data coming from the autonomous cars in real time in the cloud and identifies what can be applied through later data processing and updating. On the other hand, the internet of vehicles must determine what data needs to be processed in real time and transmit the appropriately understood data to the electric cars. In the implementation, the architecture of the big data processing technology must realize the autonomous transmission of the road data stored in the cloud and the data of traffic signage data on the road to the terminal for data preparation according to the purpose of autonomous driving and the road conditions in real time; it can also transmit its understanding of objects and various models that will have an impact on autonomous driving to the computing terminal according to the real-time perception data of autonomous driving, such as for buses. In the case of buses, for example, models for understanding the route, arrival, and historical behaviour of the bus can be passed to the terminal.

In every aspect of a vehicle, battery performance is a critical factor. AI helps analyze battery usage and charging data, as well as optimize fast charging behavior, which ensures battery performance and lifecycle management. This benefits the driving range, charging time, and vehicle life. AI-driven hardware has been developed to automate the identification and repair of defects in electric vehicle lithium-ion batteries [100]. Researchers are developing batteries that are safer, recharge faster, and are more sustainable than the current generation of lithium-ion batteries. In addition, charging information from car users will be collected and analyzed by AI to provide a more accurate and faster charging service. Toyota has announced plans to invest 5.6 billion in research and development of new energy battery materials based on artificial intelligence to further improve the performance of current car batteries and fuel cells [101].

With a variety of promising applications such as autonomous driving, intelligent navigation systems, and user behavior monitoring, AI is playing a key role in the EV industry. A broader field is expected to be explored with AI devices.

2.3 Security in Power Management of Smart Grids

Power management security remains a major issue for smart grids, as attackers can both gain economic advantage and cause catastrophic damage by, for example, plunging a city into darkness [102, 103]. A defence mechanism should therefore be able to detect and prevent potential attacks. The cybersecurity of the smart grid is not only about the resilience of the entire smart grid, but also about keeping hackers out and protecting the privacy of personal data. Hundreds of trials have been conducted worldwide to test systems that allow consumers to sell directly to each other using peer-to-peer transactions and smart contracts [104, 105]. In addition, traditional suppliers are looking for more efficient and accurate ways to read electricity meters and send bills. However, none of this would be possible without strong cybersecurity for everything related to electricity management. There are four areas where the grid and electricity management can be made more secure.

- Strong digital identities: All connected devices should have their own unique digital identity that is used to identify each device. If all devices have their own unique identity, only that device is at risk, even if a device is hacked.
- Mutual authentication: This means that two connected devices can only ”talk” to each other if they have successfully answered a digital challenge, the answer to which is known only to those two devices.
- Encryption: Data should always be encrypted when it is transferred between devices and when it is not moving to protect it from tampering.
- Constantly updated security: A secure smart grid should constantly evolve and update its security regularly, with keys and digital challenges for mutual authentication updated every two to three years.

With recent advances in mobile energy storage technologies, electric vehicles (EVs) have become a crucial component of smart grids that support power management. When EVs participate in a demand response program, an optimal EV charge/discharge control strategy can

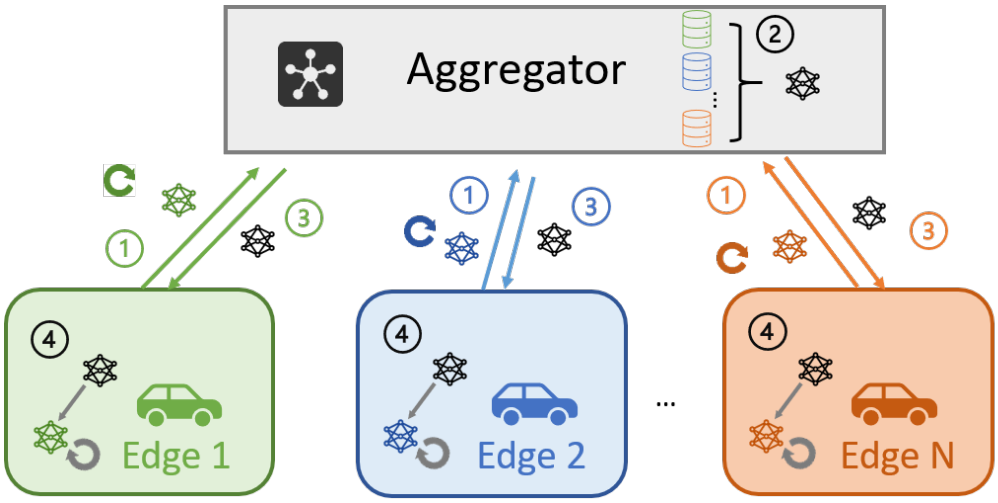


Figure 2.3: **The process of a FL-based scheme. First, each client trains a local model using local data set and then uploads the model to an aggregator. Second, the aggregator generates a global model using the collection of local models. Third, the aggregator distributes the global model to each client. Fourth, the client updates its own model and continues training.**

be performed within a secure collaborative learning scheme based on federated learning and blockchain technology [106, 107].

In recent days, machine learning approaches have centralized the data set on a server center. The standard method requires a lot of data sharing and transmission, leading to potential data leakage and heavy load on the network. Motivated by such challenges, a concept called federated learning is proposed that allows local devices to keep their local data instead of uploading it to the data center [108]. In this paradigm shift, the training tasks are performed locally while the server only works on aggregating the local models, as shown in Fig. 2.3. The original federated learning process named federated average (FedAvg) works as follows.

1. At first, each EV node i trains its local model M^i_{local} using the collected data set D^i_{local}. In each local model, the gradient ∇g^i_L is calculated by the following formula:

$$\nabla g^i_L = \frac{\partial E(W_i)}{\partial W_i} \tag{2.1}$$

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

δ indicates partial derivatives.

2. Each client i uploads the local model M_i to the aggregator.

3. Before aggregating the models, we need to calculate the contribution of each model concerning the corresponding data size:

$$w^i_{local} = \frac{|D^i_{local}|}{\sum_{i=1}^{N} |D^i_{local}|}, i \in N \tag{2.2}$$

4. The local models are aggregated, resuling in a global model with weights and biases:

$$W^r_{global} = \sum_{i=1}^{N} w^i_{local} W^r_i \tag{2.3}$$

$$b^r_{global} = \sum_{i=1}^{N} w^i_{local} W^r_i \tag{2.4}$$

5. Once the edge nodes receive the global model from server site, they update the parameters as follows:

$$W^{r+1}_i = W^r_{global} - \eta \nabla g^i_L \tag{2.5}$$

$$b^{r+1}_i = b^r_{global} - \eta \nabla g^i_L \tag{2.6}$$

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

Traditional transactions are recorded in written ledgers that can be viewed in financial institutions but are only accessible to a certain group of people. The blockchain manages transaction data by removing secrecy [109]. A blockchain is a type of distributed ledger technology (DLT) that offers shared, immutable, and transparent storage of transactions with cryptographic signatures. A blockchain facilitates electronic recording of transactions and tracking of assets in digital format [110]. Each transaction in the distributed ledger is authorized by the owner's dig-

ital signature and cannot be altered, hacked, or tampered with by the system, thus it is protected at a high level of security. Blockchain is a particularly promising and revolutionary technology as it helps reduce security risks, eliminate fraud and provide transparency in a scalable way. As the first and most popular application, Bitcoin is a representative cryptocurrency that relies on blockchain technology for security. Blockchain is a technology that can support a broader range of applications in various industries, including finance, supply chain, and manufacturing [109, 111, 112]. One of the main issues blockchain addresses is trust. Previously, records of data and transactions were kept by third parties. This information is not shared between the recorder and the participants in the transaction. If this information is easily accessible and modified, the entire system fails due to data leakage and loss of trust. The blockchain avoids this problem by excluding third parties, and there are no nodes in the system responsible for data storage. A blockchain has a structure of blocks and chains that record historical transactions. Each block is "chained" to the previous block in a sequence and is recorded immutably on a peer-to-peer network. Each transaction is cryptographically encrypted. All participants maintain an encrypted record of each transaction in a decentralized, highly scalable and resilient recording mechanism that cannot be denied.

When a client creates a new transaction on the blockchain network, the digital signatures are used to validate it Fig. 2.4 illustrates the verification process. First, the client passes the transaction data to the hash function and generates the hash value of the data. Then, the hash value is fed to the signing algorithm with the client's private key, generating an encrypted signed message. Then, the new transaction is sent to all nodes, containing the original transaction information (the signed message and the public key). Each recipient can thus perform a verification. First, one will use the same hash function and generate the hash value of the original message. Since hash mapping always produces the same output, this value is unique and should be identical to the values generated by the creator of the transaction. The signed message is then decrypted using the public key, which should make the resulting value match the previous hash value. If the decrypted hash value matches the recalculated hash value for the same data, the digital signature is proven to be valid. Therefore, this transaction is considered trustworthy and is

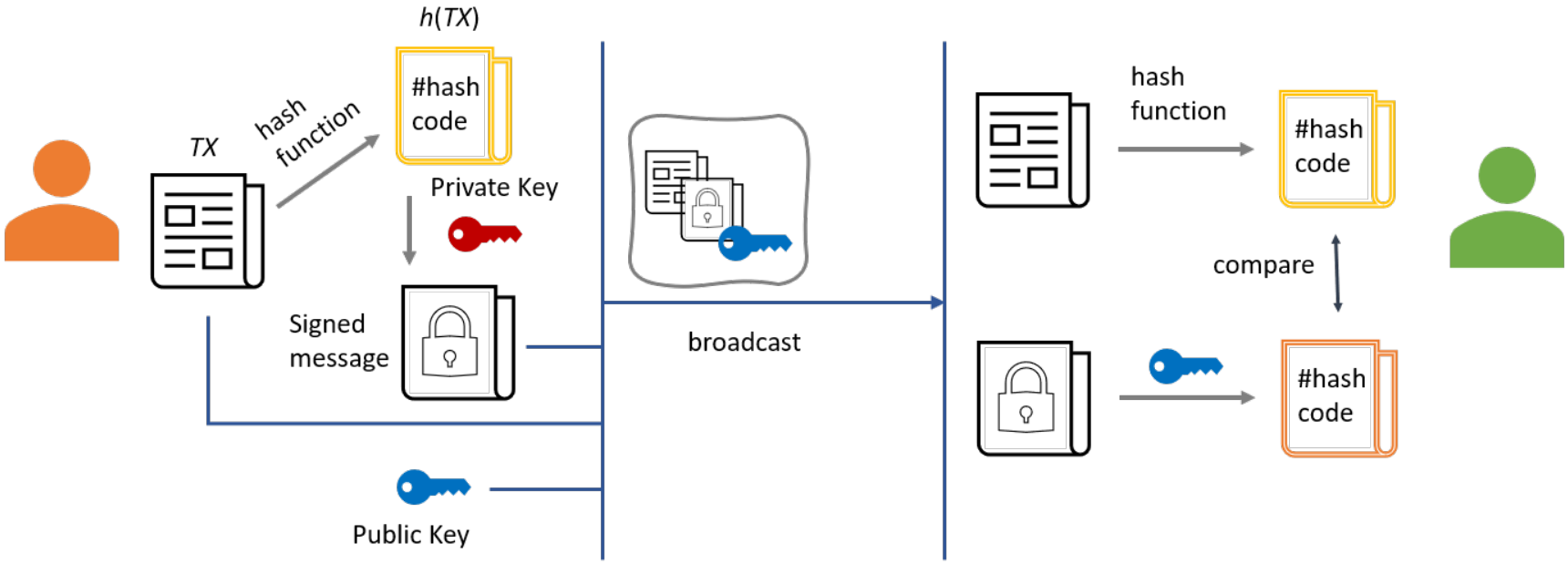


Figure 2.4: **The process of verification using digital signatures. The original transaction is fed to a hash function, which is then encrypted using the signer's private key. The signed message, the original message, and the signer's public key will be broadcasted in the network. A receiver will decode 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.**

added to the transaction pool of each node. Otherwise, the different values of the hash show that the message has been tampered with. In this case, the message is rejected by the recipients.

The transaction pool is where all valid transactions wait to be confirmed by the blockchain network. However, with the increase of unconfirmed transactions, memory consumption and computational efficiency become a challenge. To tackle this problem, the Merkle tree [113] was introduced to significantly reduce memory and computation requirements as shown in Fig. 2.5. Given a sequence of transactions TX_1, TX_2, ..., TX_n, each of them is hashed to form a leaf node of the Merkle tree. The collection of these leaf nodes is denoted by $h(TX_i)_{i\in n}$. Following that, a binary implementation is used to merge every two nodes into a new node belonging to the next layer, as described in equation 2.7.

$$\begin{aligned} h(TX_1 + TX_2) &= h(H(TX_1) + h(TX_2)) \\ h(TX_3 + TX_4) &= h(H(TX_3) + h(TX_4)) \\ &\cdots \\ h(TX_{n-1} + TX_n) &= h(H(TX_{n-1}) + h(TX_n)) \end{aligned} \tag{2.7}$$

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

The entire process of building a Merkle tree results in a single hash value called a 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 other than the nonce by M. Given a pre-determined value n; the goal is to find a nonce that satisfies the requirement shown in equation 6.2.

$$Hash(M + nonce) = \underbrace{0...0}_{n\ bits} x...x \tag{2.8}$$

Once a perfect nonce is found, it is added to the hashed block. The block header is rehashed

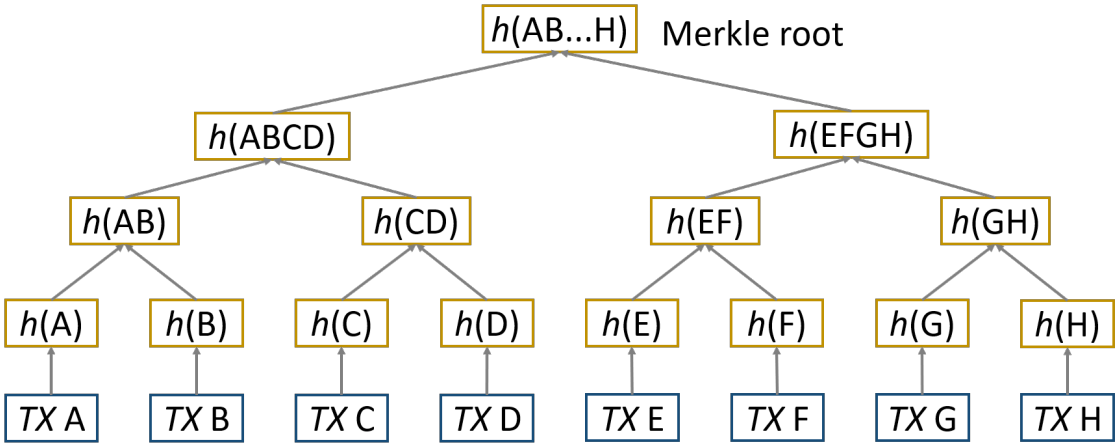


Figure 2.5: **Illustration of Merkle tree structure with eight nodes. Each node represents a transaction that is fed into a hash function, and the hashed transaction is denoted by $h(\mathrm{A})$ to $h(\mathrm{H})$ in the figure. Then a bottom-to-up operation is performed, in which $h(\mathrm{AB}) = h(h(\mathrm{A})) + h(h(\mathrm{B}))$, $h(\mathrm{CD}) = h(h(\mathrm{C})) + h(h(\mathrm{D}))$, etc. The operation ends up with a single hash value, which is referred to Merkle root. In this case, the Merkle root is $h(\mathrm{ABCDEFGH})$.**

along with the successful nonce, then the block, including header and body, is added to the chain. Therefore, the blocks in the chain are shared, immutable, and trusted.

Table 2.1: **Comparison of three data storage methods on the blockchain.**

	Public Blockchain	**Private Blockchain**	**Consortium Blockchain**
Access	• Anyone • Anonymous	• Single organization • Known identities	• Multiple organizations • Known identities
Permission	• Permissionless	• Permissioned	• Permissioned
Concensus	• All the nodes	• Managed by a single node	• Managed by a consortium of participants
Transaction Speed	• Slow	• Fast	• Depend on configuration

There are three types of blockchain technologies: (1) public blockchain; (2) private blockchain; (3) consortium blockchain, as summarized in Table 2.1. A public blockchain allows anyone to freely participate in network activities and serves as the backbone of almost any decentralized solution [114, 115]. In addition, the large number of network participants joining a secure public blockchain makes it immune to privacy breaches, hacking attempts, or other cybersecurity issues. The main disadvantage of a secure public blockchain is the significant energy

consumption required to maintain it. Other problems include the lack of complete privacy and anonymity. A public blockchain allows anyone to view the amount of the transaction and the address involved. If the owner of the address is known, the user loses anonymity.

In a private blockchain, participants can only join the network by invitation, and their identity or other required information is authentic and verified [116]. Authentication is performed by the network operator or by a predefined protocol implemented by the network through a smart contract or other approved method. A private blockchain is not considered decentralized. It is a distributed ledger that operates as a closed database and is secured by the concept of encryption and the requirements of the organization. Only those who have access can operate a full node, perform transactions, or verify/authenticate changes to the blockchain. In this regard, private blockchains are vulnerable to data breaches and other security threats. This is because only a limited number of verifiers are usually able to reach consensus on transactions and data when a consensus mechanism is available.

Consortium blockchains are a hybrid of public and private blockchains [117, 118]. Consortium blockchains allow anyone to join the permission network after the authentication process. The purpose of forming a consortium blockchain is to facilitate collaboration between a group of complementary blockchains that help each other address challenges and develop solutions for the system as a whole. Because a consortium blockchain includes multiple organizations, each organization is involved in the decision-making process, ensuring that the blockchain is not controlled by a single entity. Therefore, collaboration between private organizations in a consortium blockchain offers faster transaction operation while maintaining privacy and scalability. However, there are still some issues with consortium blockchains, such as the more complicated network structure and the effectiveness of protocol updates when a new organization joins the network.

2.4 Chapter Summary

In this chapter, we introduce the architecture of the virtual power plant and how it works. We explain various applications of artificial intelligence in vehicular networks. We also discuss

the security problem and power management solutions for smart power grids. The employment of collaborative learning using federated learning and blockchain technology is described. In the next chapter, we will discuss related works, including optimal operations in virtual power plant, AI deployment in virtual power plant, electric vehicle power consumption prediction, integration of blockchain and federated learning in vehicular networks, and client selection in federated learning.

3

Related Works

3.1 Optimal Operations in Virtual Power Plant

Distributed energy resources and variants of consumer participation are increasingly being integrated into current VPP platforms. The fluctuation of resource generation and unpredictable electricity consumption raises a challenge to the energy balance and economic benefits of VPP. Therefore, related studies have focused on the optimal operation of VPP in conjunction with efficient integration of distributed energy resources and end-user participation

The authors in [16] developed an optimal control and bidding strategy for VPP with renewable energy generations and inelastic demand, formulating the problem as a two-stage stochastic optimization. In [17], a quantile regression forest model was applied to the prediction of wind

and photovoltaic energy generation. In [18–21], information gap decision theory was used to study the uncertainty of wind energy integrated with electricity and natural gas systems. While these studies have mainly focused on power generation and electricity markets, the participation of end consumers was barely investigated.

Some works addressed the importance of EV fleet participation [22–28]. In [22], the optimal operations for EV aggregator participation in day-ahead energy and regulation markets were proposed. In [23], the authors proposed optimal scheduling algorithms for V2G energy sales and multiple ancillary services. In [24], the authors studied the tradeoff between energy and reserve markets and proposed optimal operation for uncertain EV battery degradation. In [25], a look-ahead power scheduling algorithm was proposed to manage EV aggregation revenue risk against fluctuating power generation and electricity prices.

However, these studies hardly emphasized the practical power consumption of EVs, which could be predicted based on static and dynamic information (e.g., driver behavior, usage time, and weather conditions). In [27, 28], a solution to quantify preferences based on unknown EV types was investigated. However, since the aggregator has to wait for the interaction until a number of EVs arrive at the parking lot, instead of predicting the electricity consumption of EVs in advance, there is an inevitable delay in energy trading, which also affects the utilization of EVs in car-sharing markets [119].

3.2 AI Deployment in Virtual Power Plant

In recent years, AI technologies have seen a steady increase in various VPP applications. Works in [29, 30] approached economic dispatch using reinforcement learning (RL) or non-dominated genetic sorting algorithms. Variants of intelligent energy management methods based on RL [37–39, 120, 121] and recurrent neural networks (RNN) [33, 34] have also been proposed. Works in [35] employed explainable AI tools and artificial neural networks for photovoltaic power prediction, while [36] proposed an ensemble learning-based model for wind power prediction. In [122], demand-side energy management with price forecasting based on a multilayer perceptron was proposed. The authors in [31] integrated an RL method for an

EV bidding strategy. In [47], a novel centralized learning algorithm for electric vehicle energy demand prediction was presented. Considering that most of these works perform the experiments on a single centralized server, the system faces the following problems. First, there might be a latency and cost bottleneck when the center collects all the distributed data and performs the learning. Second, the stability of the whole system depends heavily on the centralized server. That is, if the server fails, the queries from all the distributed nodes will not be answered.

Moreover, once attackers access the centralized server, the private data is easily fetched or modified. On the contrary, in some edge computing paradigms [47, 120], the computation is moved from data centers to local devices. However, there remain limitations in the storage and speed of the edge nodes.

3.3 Electric Vehicle Power Consumption Prediction

Vatanparvar et al. [123] proposed a novel context-aware methodology for estimating driving behavior with respect to future vehicle speeds for up to 30 seconds. In [124], a speed optimization framework is modeled for both battery life and power consumption of smart electric vehicles during acceleration. Since these works focused only on the acceleration process, they are not suitable for long-trip scenarios. Ferro et al. [125] presented a detailed energy consumption model that considers all aspects affecting vehicle dynamics. Baek et al. [126] presented a general methodology to predict and optimize the operating range of EVs. Zhao et al. [127] proposed a combined machine learning model to predict the remaining range of EVs based on real driving data. A shortcoming of these methods is the complexity of their models. That is, prediction for a single route requires a large amount of vehicle, route, and battery data. In addition, careful and elaborate route-planning for a terrestrial EV involves high time and data storage costs. Features, such as weather conditions and geography were not investigated.

Gomez-Quiles et al. [128] proposed a novel ensemble method to predict the power consumption of electric vehicles by examining the non-stationary time series of consumption. Although the algorithm is used for predictions for the next one to two months, it is unsuitable for specific driving activities.

3.4 Integration of Blockchain and Federated Learning in Vehicular Networks

The work in [129] discussed communication costs, resource allocation, incentive learning, and security and privacy issues. Weng et al. [130] proposed DeepChain, a framework with a value-based incentive mechanism based on blockchain for secure collaborative training. Wang et al. [131] studied two types of Byzantine attacks in a blockchain-empowered decentralized, secure multi-party learning system. Pokhrel et al. [60] proposed a local on-vehicle machine learning (oVML) method in an autonomous blockchain-based FL design. Bao et al. [132] proposed a decentralized FL system that provides incentives and disincentives for collaborative modeling. To analyze the latency performance and robustness of the blockchain system, decentralized architectures named BlockFL and FL-Block, were introduced in [133] and [134] respectively. Despite the consideration of communication and computation costs as well as incentive mechanisms, the increasing number of parties in the blockchain-based FL network poses a significant challenge to the efficiency and applicability of the systems described in the works above.

3.5 Client Selection in Federated Learning

The original FedAvg algorithm in [108] randomly selects a group of clients in each training round, which means that communication quality and delay are difficult to evaluate. The authors in [135] investigated performance degradation due to non-independently and identically distributed (non-IID) data in the FL protocol. The approach focuses on client resource constraints, including data heterogeneity, computation limitation, and communication capability. In [136], the authors proposed a multicriteria-based approach for client selection in FL that aims to group many clients in each round to reduce communication rounds. However, none of these works considered the importance of local data affecting learning performance.
He et al. [137] proposed a different scheme for data selection and resource allocation based on the importance of data in the FL system to improve learning efficiency. The authors in [138] identified a fundamental property of FL, namely the temporal pattern and varying significance

of different learning rounds. They formulated a long-term client selection and bandwidth allocation problem under finite energy constraints and proposed a new Lyapunov-based online optimization algorithm to guarantee the long-term performance. Cho et al. [139] presented a convergence analysis of FL with limited client selection and demonstrated how local losses affect the convergence speed. Zhang et al. [140] proposed a weight-based client selection mechanism to detect the non-IID degrees of local data. However, the above strategies were applied only when the clients' reputation remained unchanged. Considering that an edge node is vulnerable to attacks in any training round, the quality of the model decreases due to tampering. Therefore, a long-term client selection mechanism is required to achieve a robust FL model.

3.6 Chapter Summary

This chapter presents related works on the optimal operation of a virtual power plant, the use of AI in a virtual power plant, the prediction of electric vehicle power consumption, the integration of blockchain and federated learning in vehicular networks, and client selection in federated learning. In addition, the remaining challenges of related works are highlighted. In the next chapter, we present the proposed network of electric vehicles (NoEV) for power management in smart grid and the prediction method for power consumption.

4

Power Consumption Prediction for Electric Vehicles

4.1 Network of Electric Vehicles (NoEV) for Power Management in Smart Grid

This section presents the proposed system and fundamental algorithms for power management in the smart grid. As shown in Fig. 4.1, the network of EV (NoEV) is integrated into a virtual power plant with energy consumers and power grid. The main idea is that the NoEV communicates with the VPP when the consumers need energy, and delivers the energy from the EVs to the consumers through the power grid. The EV battery discharge decision is made

by the EV charging mechanism, which is explained as follows.

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. 4.2 describes the proposed algorithm for calculating the energy that each EV should return to the power grid when needed.

First, we calculate the current remaining power for each vehicle considering the maximum battery capacity and state of charge (SoC). After that, we compare the current remaining power with the expected power consumption based on a fully-connected neural network. We collect information about past trips, including weather data, geographical data, driver data, and power consumption data for model training, as shown in Fig. 4.3(a). When a trained model is available, the future data, including weather data, geographical data, and driver data, are input for model inference, i.e., power consumption prediction, as shown in Fig. 4.3(b). The detailed structure of the neural network model is shown in Fig. 4.4. The neural network model is trained collaboratively under a blockchain architecture, as shown in Fig. 4.5. First, each EV client i trains a local model M^i_{local} using local data D^i_{local}. In each local model M^i_{local}, the gradient ∇g^i_L is calculated according to the following formula:

$$\nabla g^i_L = \frac{\partial E(W_i)}{\partial W_i} \tag{4.1}$$

Here W_i denotes a set of weights, and $E(W_i)$ denotes the loss function with respect to W_i. $E(W_i)$ is used for measuring the model error and finding an optimal solution. Also, ∂ indicates partial derivatives. Each local model is stored as a transaction TX_i and uploads on the blockchain. Each transaction is verified and added into the transaction pool regarding one or more clients, which is then packed into a block. To be added to the blockchain, each block must contain the answer to a complex mathematical problem created using an irreversible cryptographic hash function, as explained in Section 2.3. After mining completed, each client can download the set of local updates. Before aggregating the local models, we need to calculate the

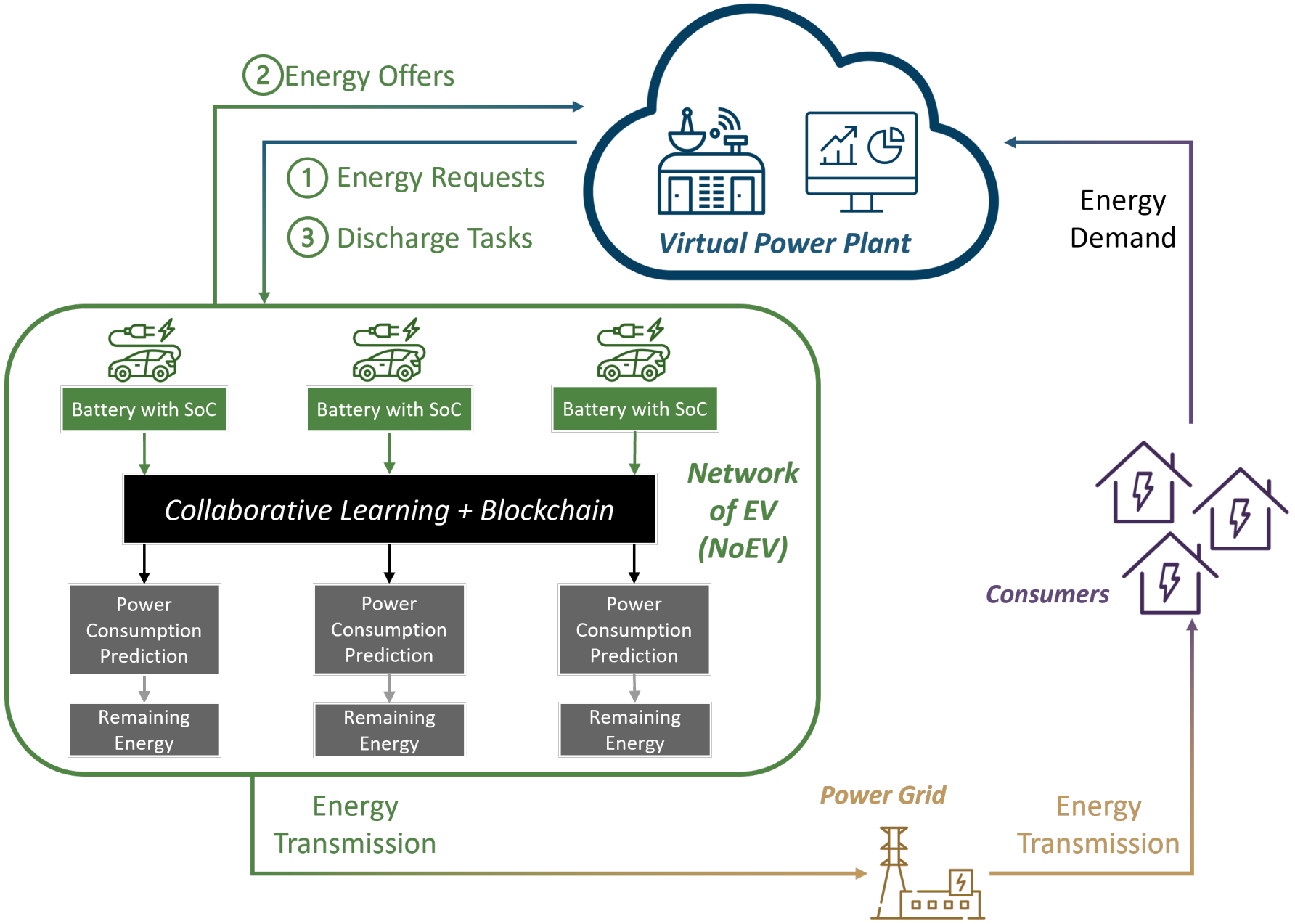


Figure 4.1: **Overview of network of electric vehicles (NoEV) for power management in smart grid.**

contribution of each model concerning the corresponding data size:

$$w_{local}^{i} = \frac{|D_{local}^{i}|}{\sum_{i=1}^{N} |D_{local}^{i}|}, i \in N \tag{4.2}$$

Then the local models are aggregated, resulting in a global model with weights and biases:

$$W_{global}^{r} = \sum_{i=1}^{N} w_{local}^{i} W_{i}^{r} \tag{4.3}$$

$$b_{global}^{r} = \sum_{i=1}^{N} w_{local}^{i} b_{i}^{r} \tag{4.4}$$

The edge node i then updates the parameters as follows:

$$W_{i}^{r+1} = W_{global}^{r} - \eta \nabla g_{L}^{i} \tag{4.5}$$

$$b_{i}^{r+1} = b_{global}^{r} - \eta \nabla g_{L}^{i} \tag{4.6}$$

Where W_{i}^{r} and b_{i}^{r} denote the weights and biases of node i in the r_{th} training round, respectively. η denotes the learning rate. The model is finished training until convergence, when the power consumption is predicted. This AI-enabled blockchain-based electric vehicle integration system (AEBIS) can be built in the controller area network (CAN), as illustrated in Fig. 4.8.

When the remaining power is less than the expected consumption or the electric vehicle is on the road, it cannot supply the power at that time. Therefore, $E^{available}$, which indicates the maximum amount of power that the EV can deliver, is set to zero. Nevertheless, the remaining power is useful information for the next driver to make a reservation. On the other hand, if the vehicle is parked in the charging station and the power remains until it is consumed in the next period, the available power is calculated as follows:

$$E^{available} = RP - ECP \tag{4.7}$$

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 $\left\{E_i^{available}\right\}_{i\in\mathbb{R}^N}$, 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 $\left\{E_i^{available}\right\}_{i\in\mathbb{R}^N}$:

$$E^{supply} = \sum_{i=1}^{N} E_i^{available} \tag{4.8}$$

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 supply will be:

$$\rho = EEL/E^{supply} \tag{4.9}$$

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

$$E_i^{discharge} = \rho \times E_i^{available} \tag{4.10}$$

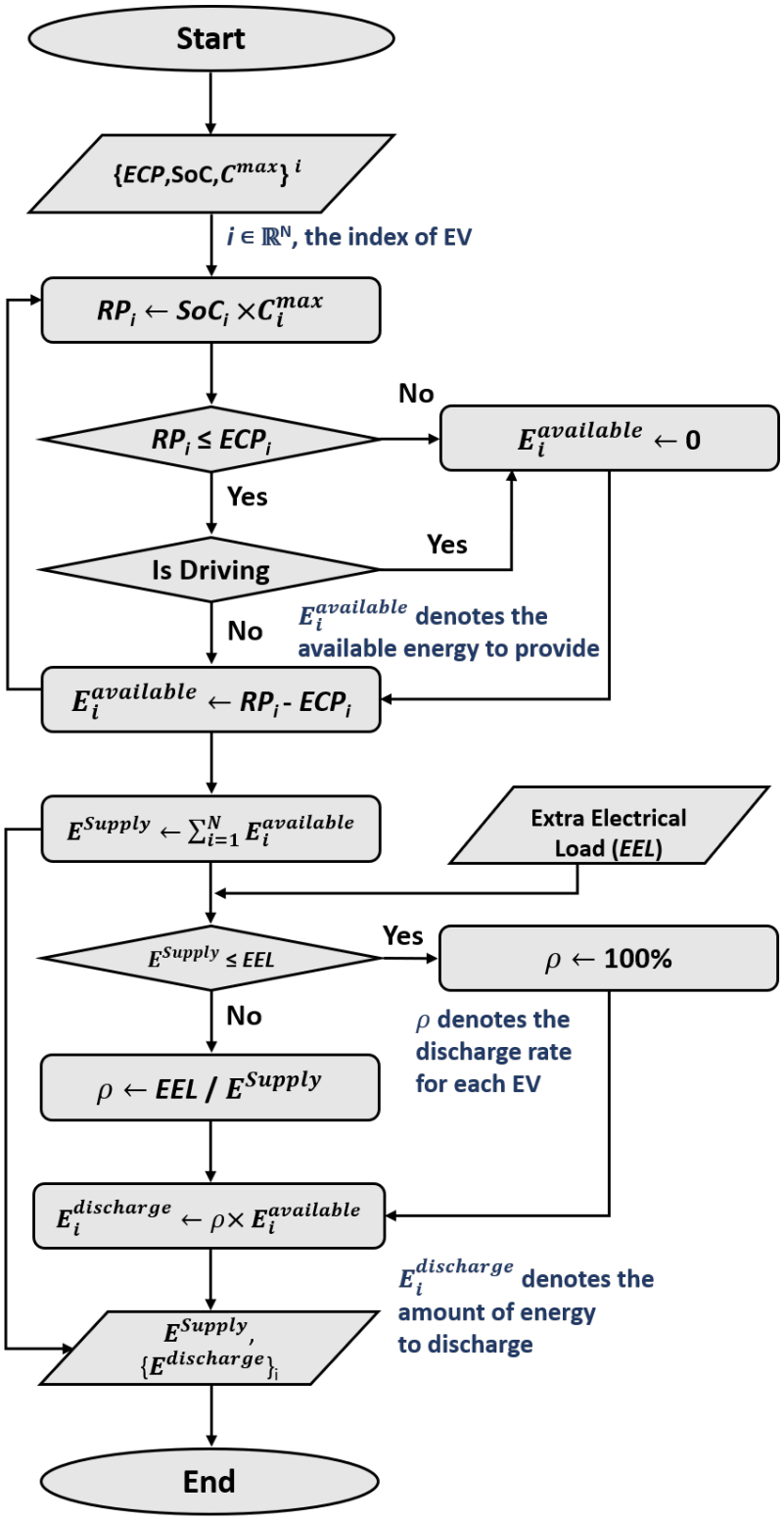


Figure 4.2: **Flowchart for calculating the electrical supply from the EVs to power grid. In stage I, the available electricity of the EVs is output. In Stage II, the electrical supply from each EV is then calculated.**

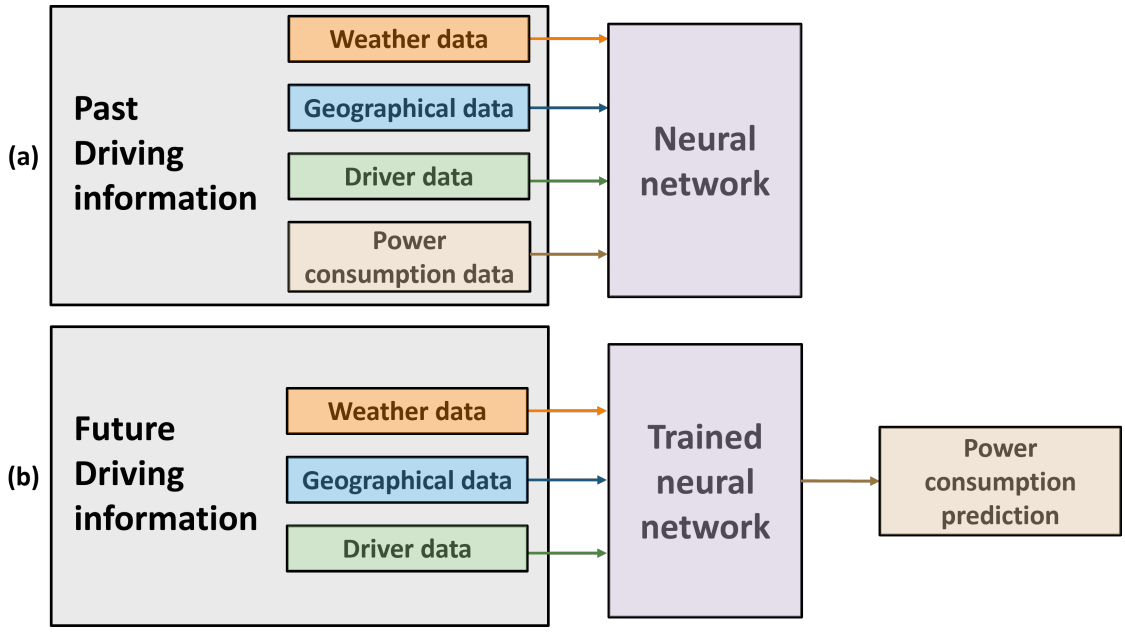


Figure 4.3: **Overview of the data collection, model training and inference for power consumption prediction. (a) Collection of weather data, geographical data, driver data, and power consumption data for model training; (b) Collection of weather data, geographical data, and driver data for model inference.**

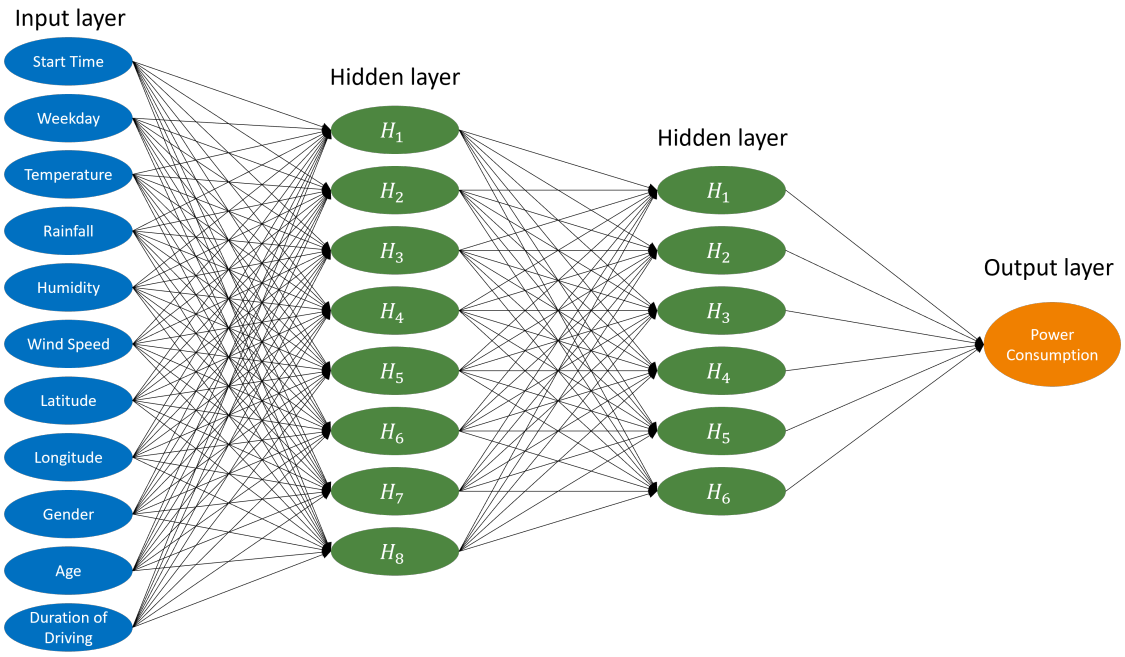


Figure 4.4: **The four-layer neural network. The input layer contains 11 input features. Two hidden layers have eight and six hidden neurons respectively. The output layer has one output neuron, i.e., power consumption prediction.**

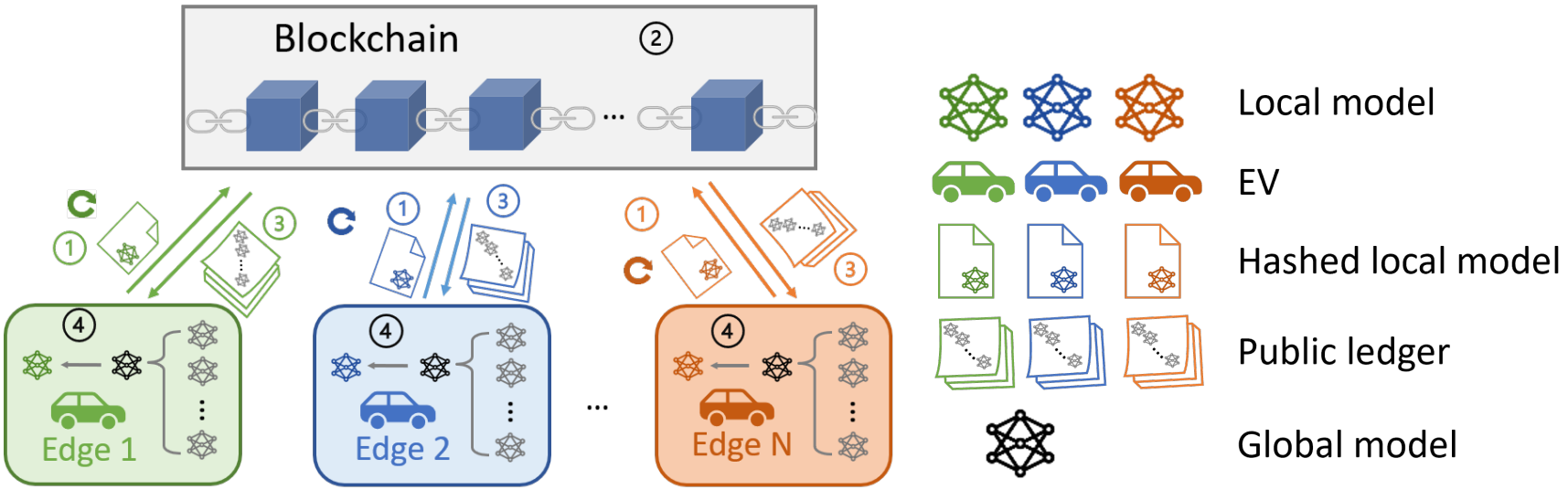


Figure 4.5: **Decentralized FL-based scheme.**

4.2 CAN Bus Communication Protocol

The Controller Area Network was introduced by the Robert Bosch Company in 1986 at the Society of Automotive Engineers conference in Detroit, Michigan, USA. A year later, the first CAN controller chip, the 82526, was produced by Intel [141]. In 1991, Mercedes developed the S-Class W140, which was the first car with a CAN-based vehicle network [142]. In November 1993, ISO officially published the Controller Local Area Network CAN International Standard (ISO 11898), which paved the way for the standardization and promotion of the Controller Local Area Network [143]. The CAN bus enables the electronic control units (ECUs) to communicate with each other over a twisted pair wire, CAN high and CAN low for signal integrity. ECUs are used to control the driving condition of the car and realize its various functions. The main purpose is to use various sensors and collect and exchange bus data to determine the vehicle status and driver's intention, and control the car via actuators. Nowadays, ECUs have become one of the most common components in automobiles and can be divided into different types according to their functions. The most common ones are: Engine control, transmission control, body control, electronic stability program, battery management and vehicle control. A central control node is not required for the CAN standard. When the bus is idle, any node can send messages to the bus. In addition, the node that first sends messages to the bus is granted the right to send messages to the bus. If several nodes send messages to the bus at the same time, the node with the higher priority of the sent messages gets the right to send messages to the bus. The priority of a message is represented by its message ID. In the standard CAN, as shown in Fig. 4.6, the message ID is an 11-bit identifier that sets the priority of the message. The lower the value, the higher the priority. A detailed explanation of the standard CAN and its extended version, extended CAN, can be find in the report [144]. An ECU, for example, the weather ECU as shown in Fig. 4.8, can collect its sensor data and broadcast the message to all other nodes on the CAN bus. Each ECU can decide to receive or discard the message after accepting it.

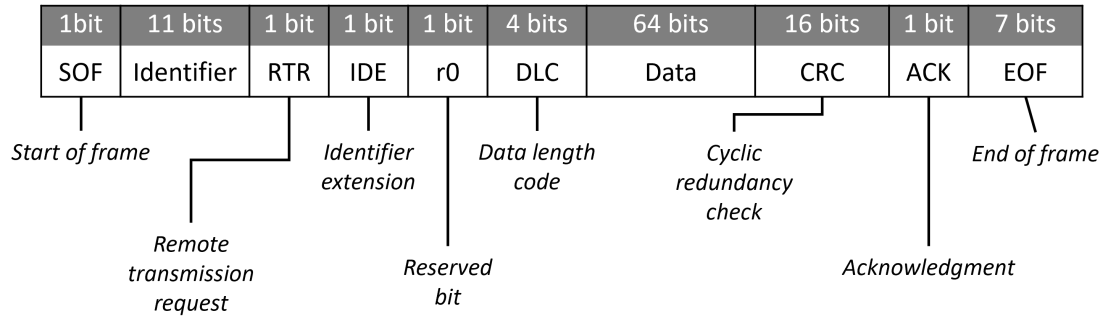


Figure 4.6: **The Standard CAN: 11-Bit Identifier.**

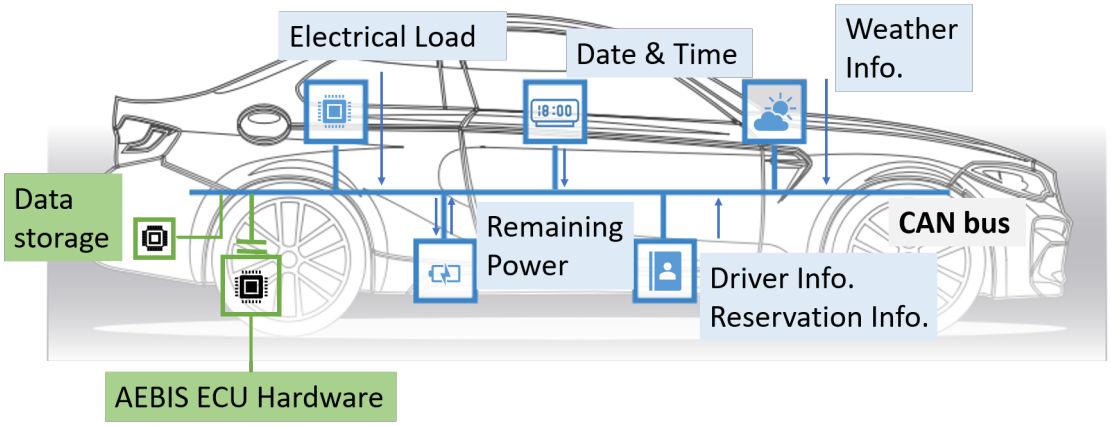


Figure 4.7: **The integration of the proposed AEBIS into the built-in Controller Area Network (CAN) of Electrical Vehicles (EVs). A CAN bus is a robust vehicle interconnect standard allowing microcontrollers and devices to communicate with each other. Each blue box indicates a built-in electronic controller 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.**

4.3 Analysis of Response Time

For an active consumer, the response time is defined as the time from then the consumer submits an energy demand until the start of the energy supply. We consider a group of energy consumers $\{c_i\}, i \in N$, N is the number of consumers. A group of EVs is defined as $\{ev_j\}, j \in M$, M is the number of vehicles. We divide the entire process into six phases: 1) Energy demand reception and organization; 2) Energy request notification; 3) Available energy prediction and energy offer reply; 4) Discharge task allocation; 5) Discharge task notification; 6) Energy transmission.

4.3.1 Energy Demand Reception and Organization

The time at which a consumer c_i sends a demand to the virtual power plant is denoted by t_i^c. The time at which the VPP receives all demands depends on the last consumer, which is formulated as follows:

$$t^c = max(t_1^c, t_2^c, ..., t_n^c) \tag{4.11}$$

We denote the time for organizing the energy demand by t^{org}. Thus, the total time for energy demand reception and organization is:

$$T_1 = t^c + t^{org} \tag{4.12}$$

4.3.2 Energy Request Notification

After collecting and organizing the energy demands, the VPP sends the energy request to the network of EV. The time required for this phase is denoted by T_2.

4.3.3 Available Energy Prediction and Energy Offer Reply

When an EV ev_j receives the energy request, it predicts the available energy it can offer based on the state of charge (SoC) and predicted power consumption, taking the time cost of t_j^p. ev_j then responds to the VPP for its energy offer information. The time required for transmitting the offer is denoted by t_j^o. In addition, a timeout t_{out}^o is specified to control the maximum waiting

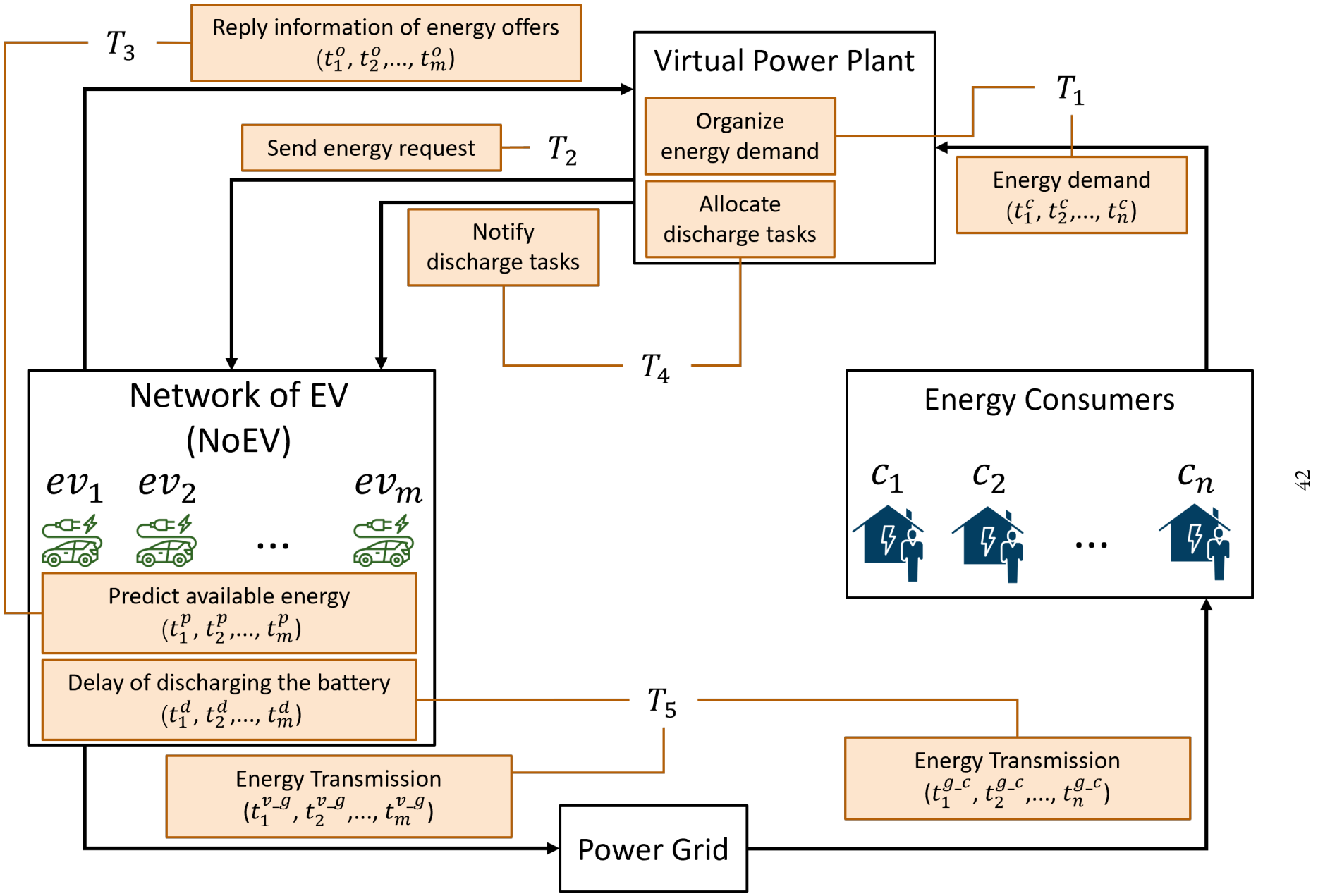


Figure 4.8: **Analysis of response time in NoEV**

time for EV nodes. The time to respond to the virtual power plant depends on the last EV, which is formulated as follows:

$$t^r = max(t_1^p + t_1^o, t_2^p + t_2^o, \cdots, t_n^p + t_n^o) \tag{4.13}$$

We denote the time for organizing the energy demand by t^{org}. Therefore, the total time for energy demand reception and organization is:

$$T_3 = min(max(t_1^p + t_1^o, t_2^p + t_2^o, \cdots, t_n^p + t_n^o), t_{out}) \tag{4.14}$$

4.3.4 Discharge Task Allocation and Notification

After receiving the energy offers from the EV fleet, the VPP starts allocating the discharge tasks based on the demand and offer information, where the time cost is denoted by T^{alloc}. Then, the VPP notifies the EV fleet of the discharge tasks, where the time cost is denoted by T^{notif}. The total time for assigning the discharge tasks and notifying is:

$$T_4 = t^{alloc} + t^{notif} \tag{4.15}$$

4.3.5 Energy Transmission

When an EV ev_j receives the notification, it starts executing the discharge task. The delay in discharging the battery is denoted by t_j^d. The energy transmission from the EV ev_j to the power grid is denoted by $t_j^{v\_g}$ and from the power grid to the consumer c_i is denoted by the value $t_i^{g\_c}$. If the energy delivered to c_i comes from ev_i, the time required for energy transmission regarding c_i is:

$$T_5 = t_j^d + t_j^{v\_g} + t_i^{g\_c} \tag{4.16}$$

In summary, the response time for an energy demand with respect to a consumer c_i is:

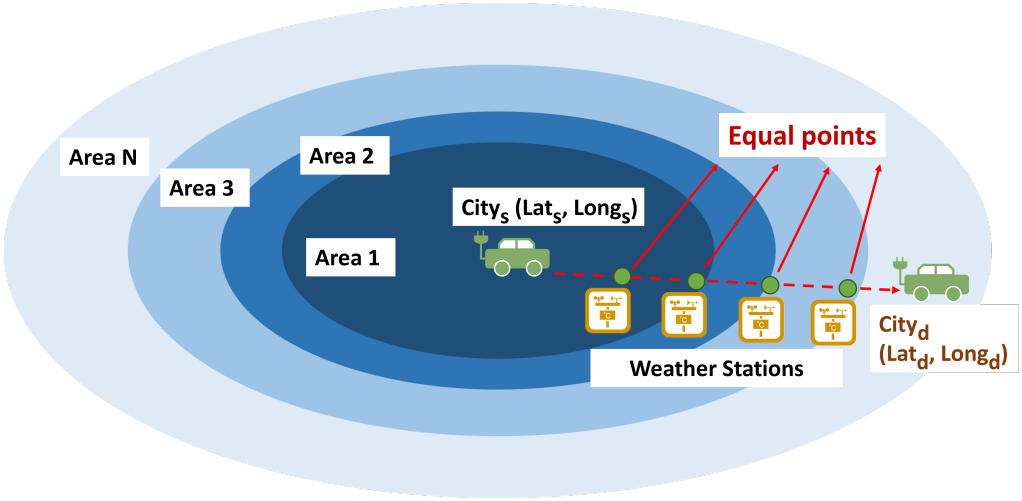
$$\begin{aligned} T^{res} &= T_1 + T_2 + T_3 + T_4 + T_5 \\ &= t^c + t^{org} + T_2 + min(max(t_1^p + t_1^o, t_2^p + t_{12}^o, \cdots, t_n^p + t_n^o), t_{out}) + \\ &\quad t^{alloc} + t^{notif} + t_j^d + t_j^{v\_g} + t_i^{g\_c} \end{aligned} \tag{4.17}$$

4.4 Multi-Stage Power Consumption Prediction Method

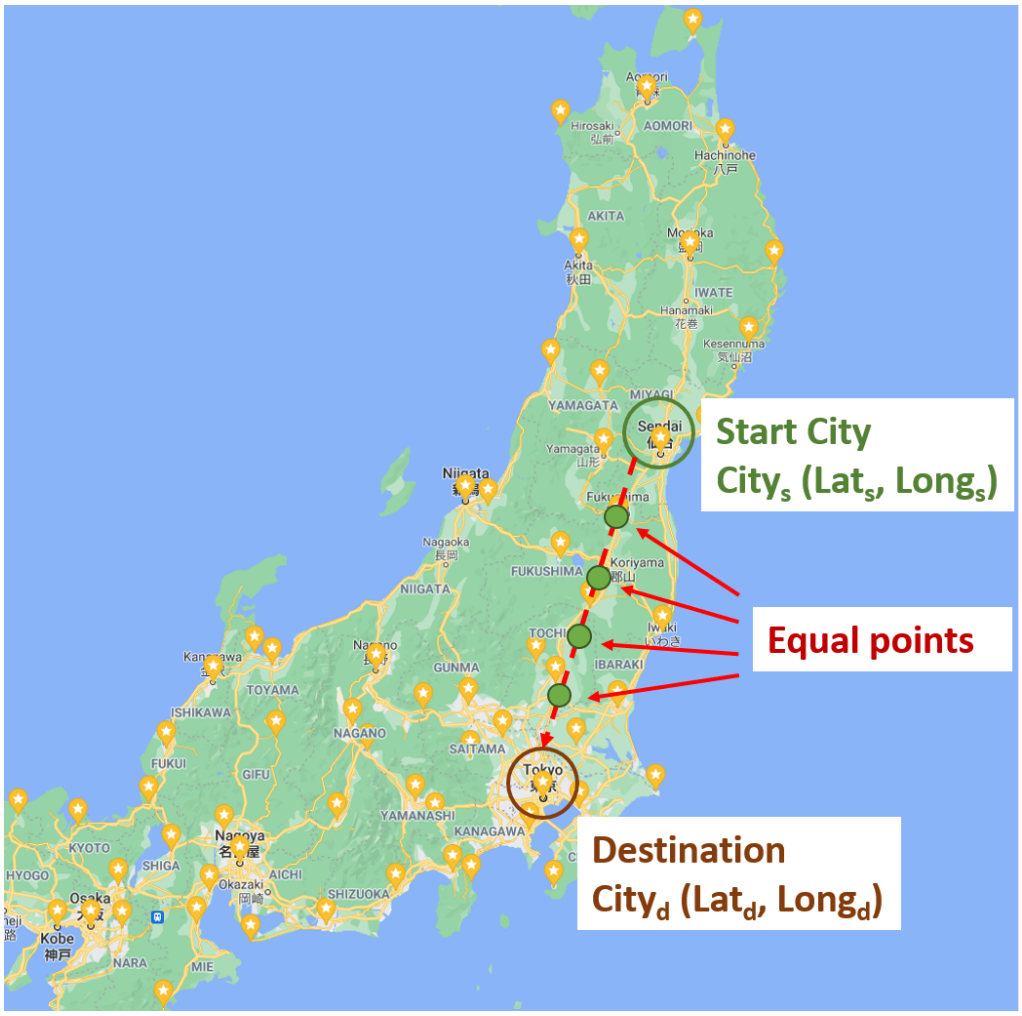
To present the multi-stage power consumption prediction method, we consider a single trip from a start city to a destination, as shown in Fig. 4.9(a). The start city is located in **Area 1** and is denoted by $City_s$. The destination is located in **Area N** and is denoted by $City_d$. Each city is associated with latitude and longitude, e.g. $City_s$ is associated with latitude Lat_s and longitude $Long_s$. The duration of driving is abbreviated as DoD. We assume that DoD takes only integers and ranges from 1 to 12 hours to simplify the problem. The start time is denoted by t_s. We also assume that the EV moves at a constant speed in a straight line. Therefore, we can calculate the position of the EV at each time t, $t \in \{t_s, t_s + 1, ..., t_s + DoD - 1\}$. Each calculated position $City_c$ is called an "equal point" because the distance between two adjacent points is the same. The equal points are marked by green dots, as shown in Fig. 4.9(a). These equal points divide the entire path into multiple sections. We then predict the power consumption for each section and sum up the results. For each section, we need the following features: 1) start time t, 2) weather information at time t, 3) geographic information (latitude and longitude), 4) user information, and 5) duration of driving. For each equal point, we use the weather data from the nearest weather station, which is highlighted in yellow in Fig. 4.9(a). Algorithm 4.1 describes the proposed approach to predict power consumption method in detail.

For ease of understanding, we split the entire algorithm into the following four stages: 1) *Initialization* (Line 1–4), 2) *Intermediate Position Calculation* (Line 5–8), 3) *Practical Position Calculation* (Line 9–23), and 4) *AI Prediction* (Line 24–33).

First, a start city $City_s$ (Lat_s, $Long_s$), a destination $City_d$ (Lat_d, $Long_d$), DoD, and start time t_s are given. Latitude and longitude of all cities are stored in $\{Lat_k\}_{k \in K}$ and $\{Long_k\}_{k \in K}$, respec-



(a) **Illustration of Power Consumption Prediction Method for A Single Trip. The green dots indicate positions that the car will pass through. Icons in yellow denote the nearest weather stations with respect to the green dots.**



(b) **An Example of Power Consumption Prediction Method for A Single Trip from Sendai to Tokyo in Japan. Each yellow star denotes a city associated with an explicit weather record. Created from Google Map [145].**

Figure 4.9: **Illustration of the optimized power consumption prediction.**

Algorithm 4.1: Multi-Stage Power Consumption Prediction Method

Require: Lat_s, Lat_d, $Long_s$, $Long_d$, DoD, t_s, $\{Lat_k\}_{k\in K}$, $\{Long_k\}_{k\in K}$, $\{Weather_{k,t}\}_{k\in K,t\in T}$, $User\_Info$, N_{total}, M

Ensure: Predicted Power Consumption PC_{pred}

Initialize empty arrays Lat_c, $Long_c$, Lat_p and $Long_p$
Initialize $City\_ID$
Initialize temporary variables ED and ED_{min}
Initialize sample S of size 11, which will be fed into model M
for $\forall i \in [0, DoD)$
    $Lat_c[i] = Lat_s + \frac{Lat_d - Lat_s}{DoD} i$
    $Long_c[i] = Long_s + \frac{Long_d - Long_s}{DoD} i$
for $\forall i \in [0, DoD)$
    $ED_{min} = \sqrt{(Lat_c[i] - Lat_0)^2 + (Long_c[i] - Long_0)^2}$
    $Lat_p[i] = Lat_0$
    $Long_p[i] = Long_0$
    $City\_ID[i] = 0$
for $\forall j \in [1, N_{total})$
    $ED = \sqrt{\left(Lat_c[i] - Lat_j\right)^2 + \left(Long_c[i] - Long_j\right)^2}$
If $ED < ED_{min}$ **then**
    $ED_{min} = ED$
    $Lat_p[i] = Lat_j$
    $Long_p[i] = Long_j$
    $City\_ID[i] = j$
    $PC_{pred} = 0$
for $\forall i \in [0, DoD)$
    $S[0], S[1] \leftarrow$ hour, weekday from $t_s + i - 1$
    $S[2], S[3], S[4], S[5] \leftarrow$ temperature, rainfall, humidity, and wind speed from $Weather_{City\_ID[i], t_s+i-1}$
    $S[6] = Lat_p[i], S[7] = Long_p[i]$
    $S[8], S[9] \leftarrow$ gender, age from User_Info
    $S[10] = DoD$
    $PC_{pred} = PC_{pred} + M(S)$
return PC_{pred}

tively, where K denotes the set of city IDs. The weather information is presented by $\{Weather_{k,t}\}_{k\in K,t\in T}$, including temperature, rainfall, humidity, and wind speed, where T is the time period of the weather data and is given by each hour. *User_Info* contains information about the driver's gender and age. N_{total} denotes the total number of cities, and M is the neural network model for power consumption prediction.

In Stage 1, the empty arrays Lat_c, $Long_c$ are initialized for recording equal points. Lat_p, $Long_p$, and *City_ID* are used to record nearest cities to each equal point. As shown in Line 6 and 7, we find coordinates of point that divide the line segment, $City_s City_d$, into multiple equal parts. The length of each array is set to DoD. The temporary variables ED and ED_{min} are initialized for calculation and storage of distance information. An empty sample S is prepared as input for model prediction. In Stage 2, the latitude and longitude of each equal point are calculated, given Lat_s, $Long_s$, Lat_c, $Long_c$ and DoD. In Stage 3, for each equal point, we traverse all practical cities and find the nearest one by Euclidean distance. In Stage 4, we prepare samples with respect to each section and perform prediction. We extract the hour and day of the week from time $t_s + i - 1, i \in [0, DoD)$. We extract gender and age from *User_Info*. Given the weather data at time $t_s + i - 1$ and a city with $City\_ID[i]$, we obtain temperature, rainfall, humidity, and wind speed. We also obtain the latitude Lat_p and the longitude $Long_p$. Finally, we input the sample S into the model M. When the prediction is completed for each driving section, we obtain the final result PC_{pred}.

4.5 Evaluation

4.5.1 EV Charging Algorithm

Evaluation Methodology

We evaluate the performance of the proposed EV charging algorithm in terms of energy fulfillment and mistaken decision. We consider a total demand of 2000 kwh, an EV battery capacity of 40 kwh, and an EV number of 100. The state of an EV is parking or driving. Also, each EV may or may not have future tasks. Energy fulfillment means how much energy the EV fleet can provide to meet the total energy demand. Mistaken decision means an EV makes a wrong

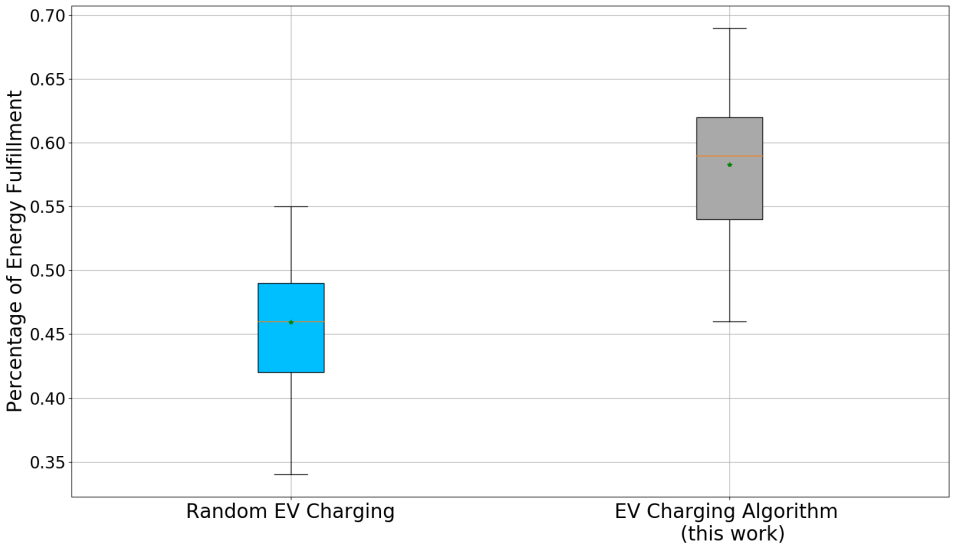


Figure 4.10: **Comparison between random EV charging and the proposed EV charging algorithm in terms of energy fulfillment.**

decision whether to provide energy or not.

Evaluation Results

The evaluation results demonstrate that the proposed EV charging algorithm achieves an average energy fulfillment of 0.58 compared to the random EV charging of 0.46. Moreover, less than 10% of EVs make wrong decisions when the proposed algorithm is applied. In comparison, random EV charging results in almost 33% wrong decisions. We conclude that the proposed EV charging algorithm achieves better performance in both energy demand response and local EV management.

4.5.2 Multi-Stage Power Consumption Prediction Method

Evaluation Methodology

As discussed previously, the data set for the power consumption prediction includes weather, geography, and user information. We collected weather data from December 2019 to November 2020 in 63 cities in Japan [146]. The start time of vehicle reservation was set from 0:00 to 23:00 and the duration of driving from 1 to 12 hours. We considered the age of drivers

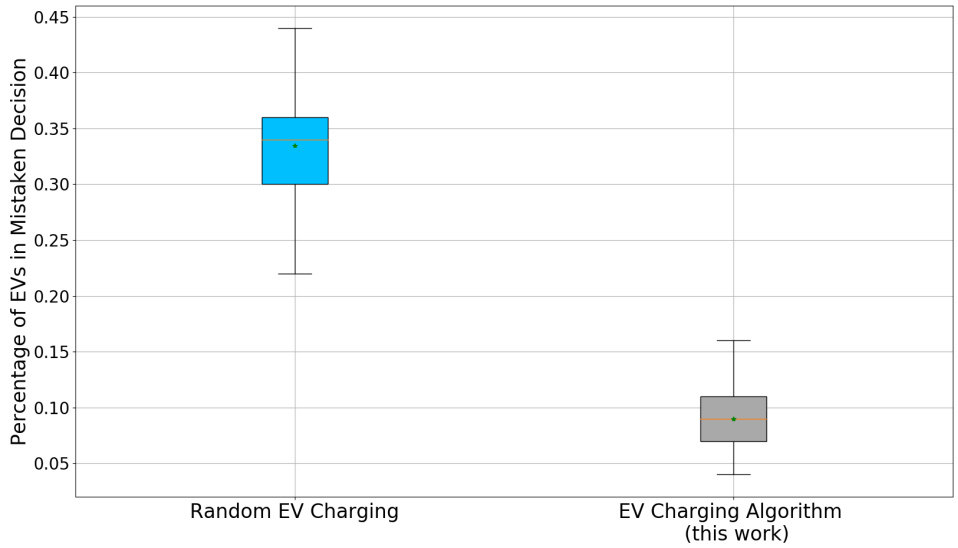


Figure 4.11: **Comparison between random EV charging and the proposed EV charging algorithm in terms of mistaken decision.**

ranging from 21 to 69 years old. The daily power consumption was measured considering the input characteristics and the measurement model [147]. We summarize the detailed information of the data set in Table 4.1. The data set contains a total of 66000 samples. We compared the proposed multi-stage power consumption prediction with the original power consumption prediction (PCP). We investigated the performance of the two methods under different driving activities — (a) short-distance journey, (b) mid-distance journey, and (c) long-distance journey. We summarize our definition of the above three activities in Table 4.2.

Evaluation Results

A comparison between PCP and the proposed multi-stage PCP is illustrated in Fig. 4.12. The overall prediction results are shown in Fig. 4.12(a), where the multi-stage PCP achieves 5.7% lower RMSE compared to PCP. We observed that the multi-stage PCP performs better in scenarios with a short distance. This result is surprising because the original PCP mainly focuses on local driving activities and has achieved decent performance. Our most compelling case is long-distance driving. As can be seen in Fig. 4.12(d), the multi-stage PCP still achieves better results by achieving 14.3% lower RMSE. We also analyzed the performance variance of the two

Table 4.1: **Multi-stage vehicle energy consumption data set.**

Input Feature	**Value**	**Unit and Datatype**
Start Time	0 to 23	-, Int
Weekday	1 to 7	Mon. to Sun., Int
Temperature	-13.6 to 39.5	$°C$, Float
Rainfall	0 to 97.5	mm, Float
Humidity	0.05 to 1	%, Float
Wind Speed	0 to 26.2	m/s, Float
Latitude	34.09 to 41.30	$°N$, Float
Longitude	134.84 to 141.94	$°E$, Float
Gender	0 or 1	Male/Female, Int
Age	21 to 69	Years old, Int
Duration of Driving	1 to 12	Hours, Int
Output	**Value**	**Unit and Datatype**
Power Consumption	5.43 to 139.97	kWh, Float

Table 4.2: **Driving activities.**

Driving Activity	**Duration of Driving**	**Driving Distance**
Short Distance	1 – 2 Hours	< 250 KM
Mid Distance	4 – 6 Hours	250 KM – 500 KM
Long Distance	8 – 12 Hours	> 800 KM

methods in each case. At medium and long distances, the variance of the RMSE of the multi-stage PCP is significantly larger than that of the PCP. The multi-stage approach may explain the reason for this. In the multi-stage PCP, when the distance is long, the trip is first divided into several sections and then the prediction model is run for each section. When the prediction results are summed, the errors caused by each prediction are also accumulated. Therefore, the multi-stage PCP leads to higher variability. On the other hand, for a short trip, e.g., one or two hours, the multi-stage approach has little effect, and therefore the variance of the multi-stage PCP is lower. Moreover, the unit RMSE, i.e., the RMSE per section, decreases from 0.90 to 0.52 as the driving distance increases as shown in Fig. 4.13.

4.6 Chapter Summary

This chapter introduces the proposed system and fundamental algorithms for power management in smart grid. An overview of network of electric vehicles (NoEV) for power management in smart grid is illustrated. The flowchart of charge mechanism for EVs is presented. Besides, we demonstrate the neural network model and multi-stage algorithm for power consumption prediction. The proposed collaborative learning scheme using federated learning and blockchain is also introduced. In the next chapter, a robust federated leaning algorithm for qualified local model selection (FL-QLMS) will be presented.

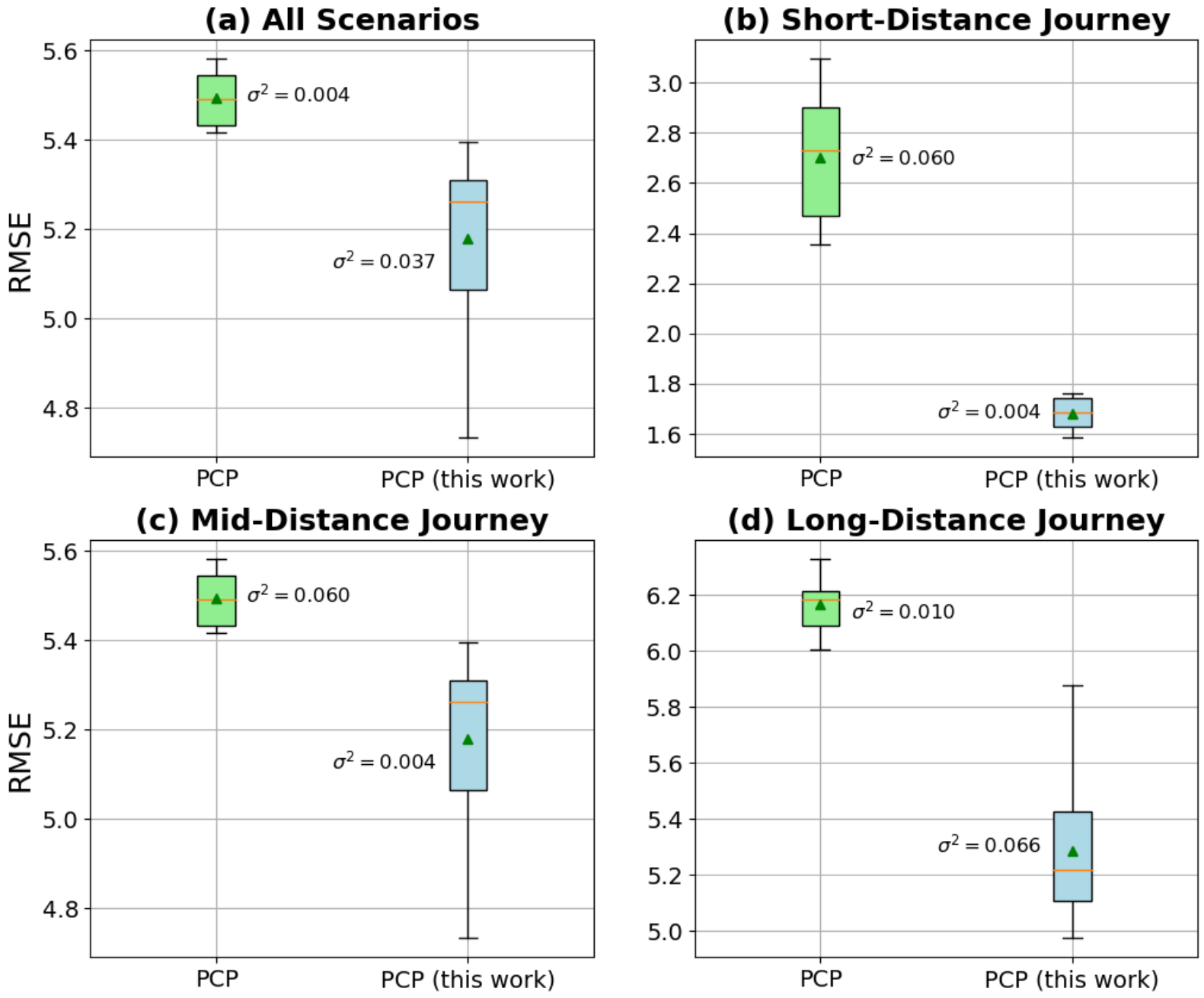


Figure 4.12: **Comparison between PCP and the multi-stage PCP (this work) in different scenarios.**

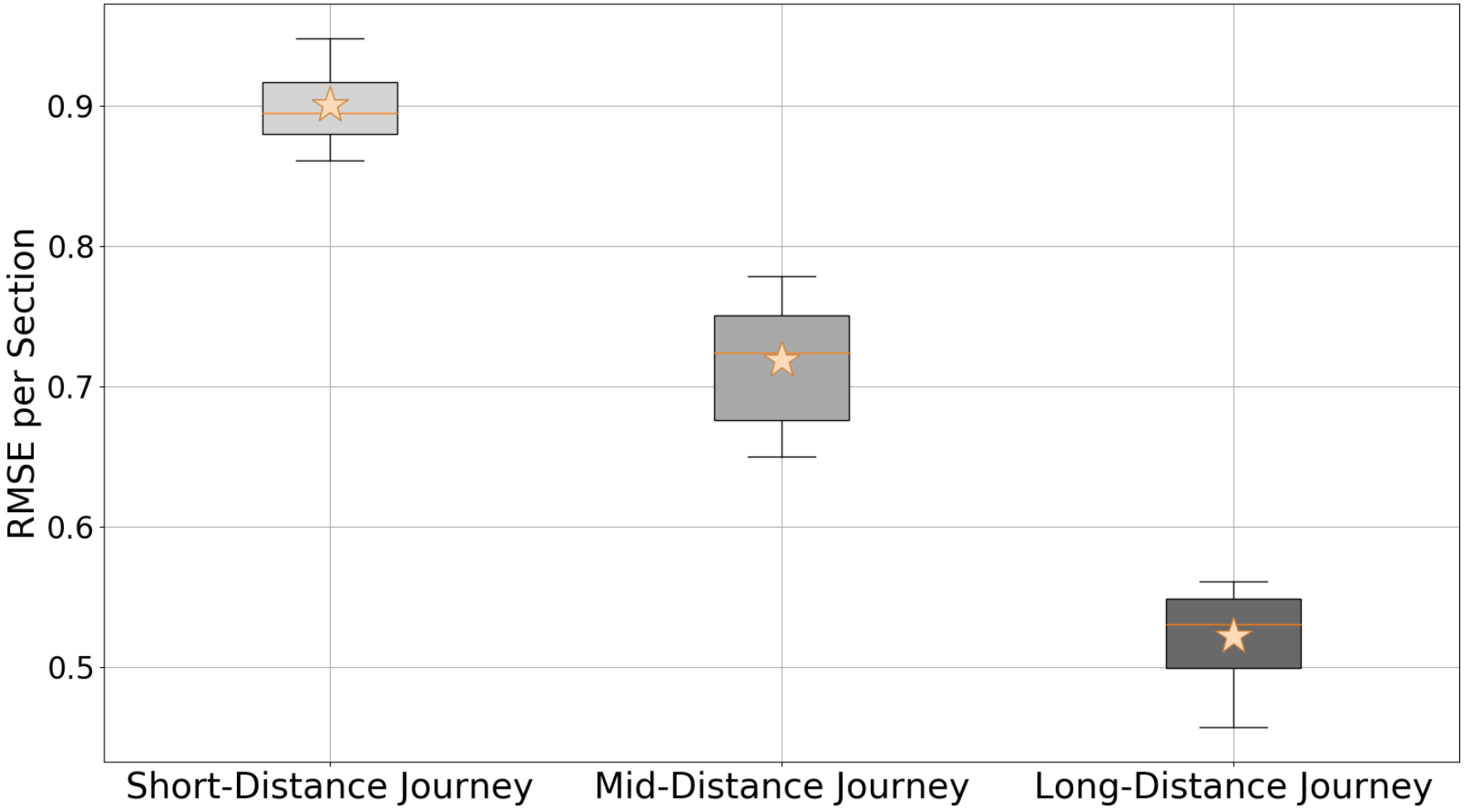


Figure 4.13: **The RMSE per section in different scenarios. The error per unit decreases with increasing distance.**

5

Robust Federated Learning Algorithm

5.1 Poisoning Attacks to Federated Learning

Since federated learning was used to achieve data protection by avoiding the transmission process, malicious local clients or attackers controlling local devices will produce fake local updates without being detected. Depending on the attacker's strategy, poisoning attacks can be classified into data attacks and model attacks [148, 149].

- Data Attacks

  Data attacks refer to the malicious modification of data of any local participants. Data injection can occur either in input features or in the output of a data sample before training begins. In random data poisoning, the targeted data points are replaced with a set of ran-

dom data. In targeted data poisoning, attackers prepare a group of specific data points to inject the desired data samples that will yield the desired prediction.

- Model Attacks

  Model attacks on local models can be carried out after local training or while the model is being transmitted. There is also another scenario where the global model is manipulated. This can happen on the server side or even during the transmisson. One attack strategy is to replace a set of parameter values of the model. The other strategy is model replacement, where a clean model is directly replaced with a prepared malicious model. Compared to data attacks, model attacks are considered more effective [150]. Since a malicious model can be faked without real training or even without a real data set, this is the preferred attack strategy.

5.2 Federated Learning for Qualified Local Model Selection

As we explained in section 3.5, the original FL approaches (i.e., the work in [108]) randomly select a group of clients in each training round, which means that communication quality and delay are difficult to evaluate. Moreover, this approach makes the model vulnerable to client attacks, which eventually leads to a severe degradation in prediction performance (e.g., accuracy in classification or root mean squared error in linear regression). Therefore, to ensure a robust learning environment, it is necessary to always select the *"qualified"* local models for aggregation, where qualified models are considered non-polluted and contribute to the performance of the global model.

In the proposed FL-QLMS algorithm, we focus on selecting a group of *"qualified"* local models for model aggregation. In general, if the distribution of the data is similar, the convergence trend of a local model should also be similar to that of the centralized model [151]. Thus, if the parameters of a local model are similar to those of the centralized model, i.e., if the parameter diversity between the two models is low, the local model is considered to contribute to model aggregation. On the other hand, if a local model is contaminated by a malicious attack, the diversity between the contaminated model and the centralized model should be high. The

diversity between two models can be expressed as follows:

$$DI_{a,b} = \|P_a - P_b\| \tag{5.1}$$

where $DI_{a,b}$ denotes the diversity between model parameters P_a and P_b.

Consider a FL process with N clients, each training round consists of the following six steps:

1. First, each client trains its local model using the collected local data set. In each local model M^i_{local}, the gradient ∇g^i_L is calculated using adaptive moment estimation (Adam) optimizer [153], as shown in the following formula:

$$\nabla g^i_L = \frac{\partial E(W_i)}{\partial W_i} \tag{5.2}$$

   where W_i denotes a set of weights, and $E(W_i)$ represents the loss function with respect to W_i. $E(W_i)$ is used to measure the model error and find an optimal solution. Also, ∂ denotes partial derivatives.

2. Each client uploads the local model M^i_{local} to the aggregator. Besides, the aggregator is informed of the local data size $|D^i_{local}|$ from each client, where D^i_{local} denotes the local data set of the client i, $i \in N$.

3. The aggregator selects a group of uploaded models based on the FL-QLMS algorithm. The number of selected models is determined by the parameter α, i.e., $\alpha\%$ of all models used for aggregation. Given a total set of N models, the number of selected models is $N_{selected} = \lceil \alpha\% \cdot N \rceil$. The list of selected models is denoted by $M_{selected}$.

4. Before aggregating the models, we need to calculate the contribution of each selected model with respect to the corresponding data size [108]:

$$w^i_{local} = \frac{|D^i_{local}|}{\sum_m^{N_{selected}} |D^m_{local}|}, i, m \in N_{selected} \tag{5.3}$$

   where $\sum_m^{N_{selected}} |D^m_{local}|$ is the total data size with respect to the selected models.

5. The selected models are aggregated to produce a global model with weights and biases:

$$W^r_{global} = \sum_{i=1}^{N} w^i_{local} W^r_i \tag{5.4}$$

$$b^r_{global} = \sum_{i=1}^{N} w^i_{local} b^r_i \tag{5.5}$$

6. Once the edge nodes receive the global model from the server side, they update the parameters as follows [153]:

$$W^{r+1}_i = W^r_{global} - \eta \nabla g^i_L \tag{5.6}$$

$$b^{r+1}_i = b^r_{global} - \eta \nabla g^i_L \tag{5.7}$$

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

We present the FL-QLMS algorithm with and without auxiliary model. Algorithm 5.1 describes how FL-QLMS works when an auxiliary data set is available. The auxiliary dataset is prepared on the aggregator side. We denote the auxiliary model as M_{aux}. First, we store all parameters (weights and biases) of M_{aux} as a one-dimensional vector, denoted by P_{aux}. We treat each local model M^i_{local} in the same way and obtain the flattened vector P_i. P_{aux} and P_i have the same size, i.e., $|P_{aux}| = |P_i|$. Then, for each model, we calculate the diversity between P_{aux} and P_i using the Manhattan distance:

$$DI_{aux,i} = \sum_{j}^{|P_{aux}|} \left| p^j_{aux} - p^j_i \right| \tag{5.8}$$

where p^j_{aux} is a parameter of P_{aux}, and p^j_i is a parameter of P_i. Then, $\lceil \alpha \cdot N \rceil$ models with the lowest $DI_{aux,i}$ are selected for aggregation.

Algorithm 5.2 describes how FL-QLMS works when no auxiliary data set is available. For each local model M^i_{local}, we store all parameters (weights and biases) as a one-dimensional vector,

Algorithm 5.1: FL-QLMS with Auxiliary Model

Require: Auxiliary model M_{aux}, local models $\{M^i_{local}\}_{i\in N}$, the total number of clients N, and parameter α
Ensure: List of selected models for aggregation
1: Initialize an empty list $M_{selected}$, which is used to store the selected local models
2: Store all parameters of M_{aux} as a one-dimensional array, denoted by P_{aux}
3: Store all parameters of each M^i_{local} as a one-dimensional array, denoted by P^i_{local}
4: **for each** $i \in N$ **do**
5: Calculate the diversity between P_{aux} and P^i_{local} using the Manhattan distance, denoted by $DI_{aux,i}$
6: **end for**
7: Select $\lceil \alpha\% \cdot N \rceil$ models with lowest $DI_{aux,i}$ and store them to the list $M_{selected}$
8: **return** $M_{selected}$

Algorithm 5.2: FL-QLMS without Auxiliary Model

Require: Local models $\{M^i_{local}\}_{i\in N}$, the total number of clients N, parameter α
Ensure: List of selected models for aggregation
1: Initialize an empty list $M_{selected}$ used to store the selected local models
2: Store all parameters of each M^i_{local} as a one-dimensional array, denoted by P^i_{local}
3: **for each** $i \in N$ **do**
4: **for each** $j \in N$ and $j \neq i$ **do**
5: Calculate the diversity between P_i and P_j using the Manhattan distance, denoted by $DI_{i,j}$
6: **end for**
7: $\bar{DI}_i = \frac{1}{N-1}\sum_{j=1,j\neq i}^{N} DI_{i,j}$ /* Calculate the average diversity between P_i and $\left\{P^j_{local}\right\}_{j\in N, j\neq i}$
8: **end for**
9: Select $\lceil \alpha\% \cdot N \rceil$ models with lowest DI_i and store them to the list $M_{selected}$
10: **return** $M_{selected}$

denoted by P^i_{local}. We then calculate the diversity $DI_{i,j}$ between P^i_{local} and each P^j_{local}, where $j \in N$ and $j \neq i$. Therefore, the average diversity of M^i_{local} can be computed as follows:

$$\bar{DI}_i = \frac{1}{N-1}\sum_{j=1,j\neq i}^{N} DI_{i,j} \tag{5.9}$$

A model with a lower average diversity is considered more representative. In other words, the data set associated with the model is assumed to have a similar distribution to the entire data set. For this purpose, $\lceil \alpha \cdot N \rceil$ models with the lowest $\bar{DI}_i$ are selected for aggregation.

5.3 Evaluation

5.3.1 Conventional vs FL-based approaches

Evaluation Methodology

The data set for the power consumption prediction includes weather, geography, and user information features, as discussed in the previous section. We collected weather data from January 2020 to July 2020 in the Fukushima, Kanagawa, and Tokyo regions of Japan [146]. The start time of vehicle reservation was set from 0:00 to 23:00, and the duration of use was set from 0 to 24 hours. We considered the age range of the driver according to the requirements of Class 2 license [152]. The daily power consumption was measured, given the input features and the measurement model [154]. We summarize the detailed information of the data set in Table 5.1.

Table 5.1: **Vehicle energy consumption data set.**

Input Feature	**Value**	**Unit and Datatype**
Start Time	0 to 23	-, Int
Duration of Use	0 to 24	Hours, Int
Weekday	1 to 7	Mon. to Sun., Int
Temperature	-11.61 to 33.83	$°C$, 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	$°N$, Float
Longitude	139.09 to 139.76	$°E$, 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

We considered the scenario where each client's data is independently and identically distributed (IID). We allocated the entire data set into 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 from different drivers. Therefore, we investigated how the distribution of non-IID

data affects performance. An interesting case is when each EV is reserved at different times of the day, i.e., morning, afternoon, evening, and night. We considered a group of four clients, each of which is associated with the period from 6:00 to 11:59, 12:00 to 17:59, 18:00 to 23:59, and 0:00 to 5:59, respectively. In addition, we are interested in the scenario in which the EVs are reserved by users in a different age group. 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, and 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.

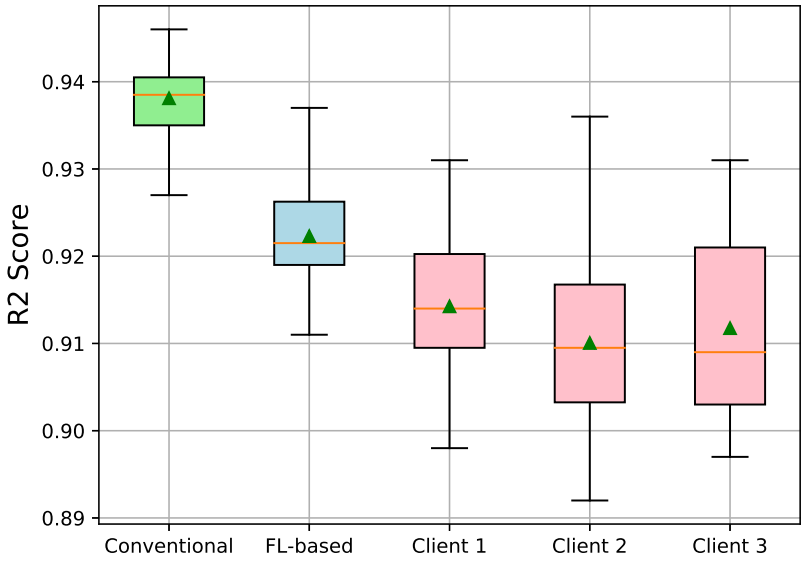


Figure 5.1: **Comparison between the conventional model, individual learning model, and the FL-based model using IID data distribution.**

Evaluation Results

For the first experiment, where the data from each client is independently and identically distributed, the result is shown in Fig. 5.1. We observe that the performance of the FL approach in the R^2 score is 0.922 on average, which is less than the conventional model (0.938). The slightly imbalanced data distribution explains the degradation in accuracy.

Figures 5.2 and 5.3 show experiments of the FL model on non-IID data distributions. In

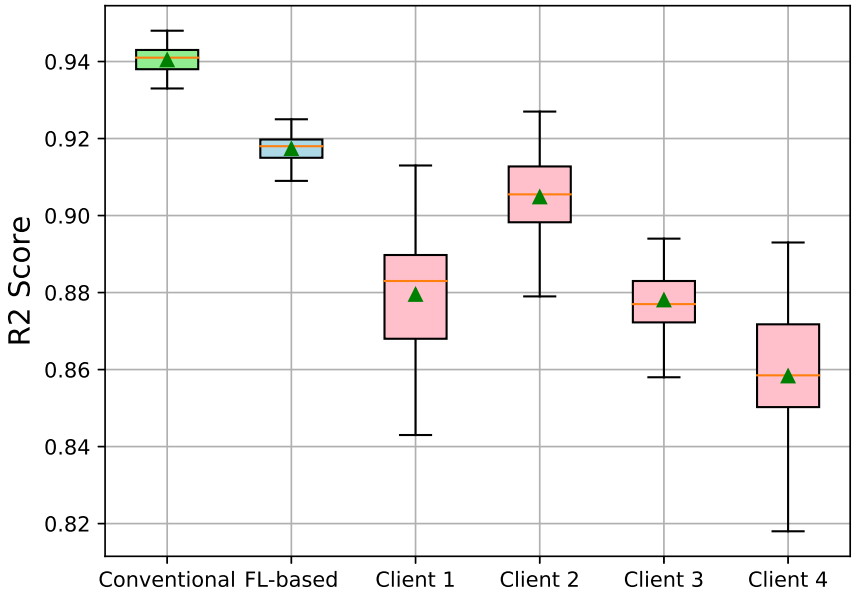


Figure 5.2: **Comparison between the conventional model, individual learning model, and the FL-based model using non-IID data distribution.**

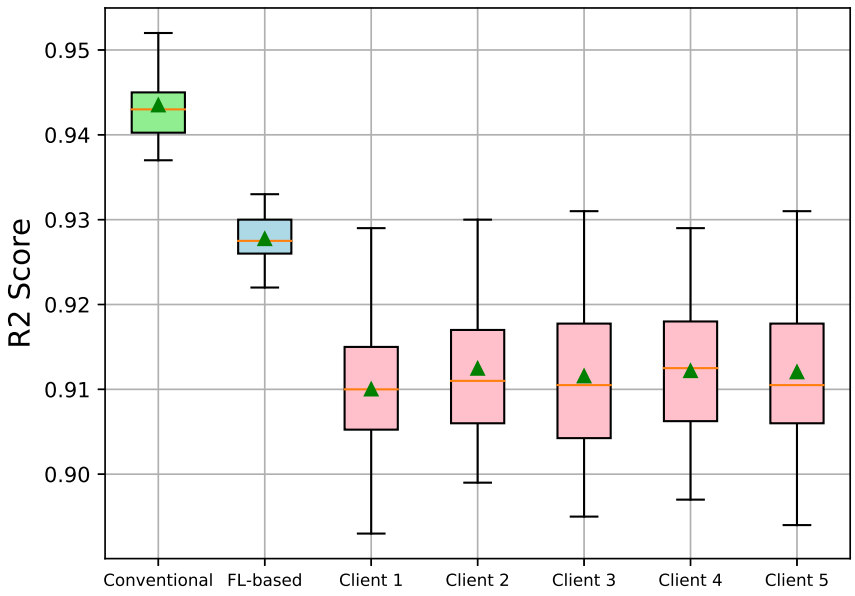


Figure 5.3: **Comparison between the conventional model, individual learning model, and the FL-based model using non-IID data distribution. There are five clients in the experiment; each client is associated with a data set concerning ages ranging from 20 to 69. The FL-based model has proven to be robust in the non-IID setting.**

this evaluation, we observe that the performance of the single client 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, accurate power prediction allows drivers to know the remaining power consumption of EVs in advance, making reservation management more efficient.

5.3.2 Federated Learning for Qualified Local Model Selection (FL-QLMS)

Evaluation Methodology

We considered a set of $N = 63$ clients in the federated learning environment. We used the data set as illustrated in Table 4.1. The data set contains 63000 training samples and 3000 test samples. First, we studied the effects of the model initialization methods — a) global initialization and b) local initialization. We considered an independent and identically distributed (IID) setting and employed the FedAvg (Federated Average) algorithm [108]. Then we considered a scenario where the local data is non-IID. Finally, we went a step further and compared the robustness of the different FL algorithms against client attacks. For each algorithm, the simulation was repeated 20 times. Each simulation included 50 iterations. We used the root mean square error (RMSE) to measure the performance of the model.

Evaluation Results

We considered a set of $N = 63$ clients for the FL schedule. We split the whole data set D into the training set D_{train} of 63000 samples and the test set D_{test} of 3000 samples. First, we evaluated two approaches to model initialization: a) global initialization and b) local initialization. Global initialization means that the aggregator creates an initial model and distributes it to all clients. Local initialization, on the other hand, means that each client creates its own initial model and performs the training task. FedAvg is used for model aggregation. We randomly assigned 1000 samples to each client. Thus, each subset D^i_{iid} follows an independent and identical distribution (IID), where $D_{train} = D^1_{iid} \cup D^2_{iid} \cup \cdots \cup D^N_{iid}$. Fig. 5.4 illustrates the impact of two model initialization options on training performance. The red and blue shaded areas indicate the performance variations for local and global initialization, respectively. While local

initialization leads to slower convergence in the first 20 iterations, it achieves a lower average RMSE of 7.77 than global initialization at the end of training. This shows that it makes more sense to build the initial models on the client side rather than on the server side. Therefore, we implement local initialization in the following FL simulations.

We then considered a scenario where all local data is non-IID We refer to this scenario as **Scenario-I**. We distributed the entire dataset across $N = 63$ clients, each of which is associated with 1 to 5 start cities. Besides, each local data set $D^i_{non-iid}$ contains different reservation times, i.e., morning, afternoon, or evening. For each $D^i_{non-iid}, i \in N$, the data size ranges from 200 to 2000. Similar to the IID scenario, we have $D_{train} = D^1_{non-iid} \cup D^2_{non-iid} \cup \cdots \cup D^N_{non-iid}$. We compared the performance of FedAvg, FCS, and the proposed FL-QLMS with or without auxiliary model M_{aux}. As shown in Fig. 5.5, the FL-QLMS with an additional model has similar performance to FedAvg, while both algorithms cannot keep up with FedCS with an average RMSE of 7.28. The reason is the robustness of FedAvg and FedCS to the non-IID. setting to some extent. Also, compared to FedCS and FL-QLMS, FedCS allows two times as many clients in each training round. We then found that the average RMSE of FL-QLMS without an auxiliary model is higher than the other methods, reflecting the importance of an additional model during training.

We further investigated the impact of hacked clients on various FL algorithms. We refer to this scenario as **Scenario-II**. We assume that $k\%$ of all clients are hacked in each training round. Each hacked client uploads a malicious model where all parameters range from -1 to 1 randomly. Compared to **Scenario-I**, we used the same setting for data distribution and training simulation. From Fig. 5.6 we can see how each method performs against model attacks of varying severity. FL-QLMS (with M_{aux}) is shown to be robust when 10% to 40% of clients are hacked, holding average performance constant. In contrast, FedAvg and FedCS are highly sensitive to attacks, as the training process hardly converges as the number of faked models increases. For FL-QLMS (without M_{aux}), it always leads to convergence, but with slightly worse performance than FL-QLMS (with M_{aux}).

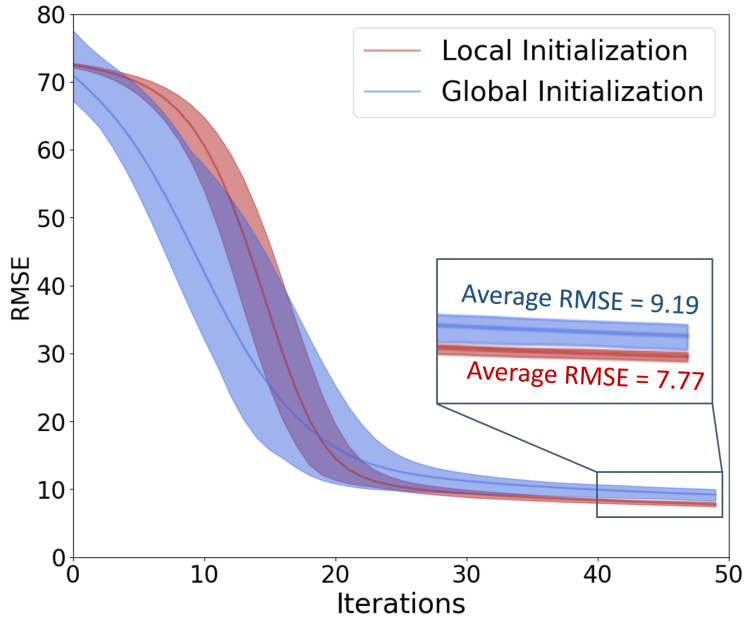


Figure 5.4: **Comparison between two model initialization methods in federated learning. Shaded regions denote the fluctuation of the performance. The meaning of iteration is the number of times that the models were aggregated.**

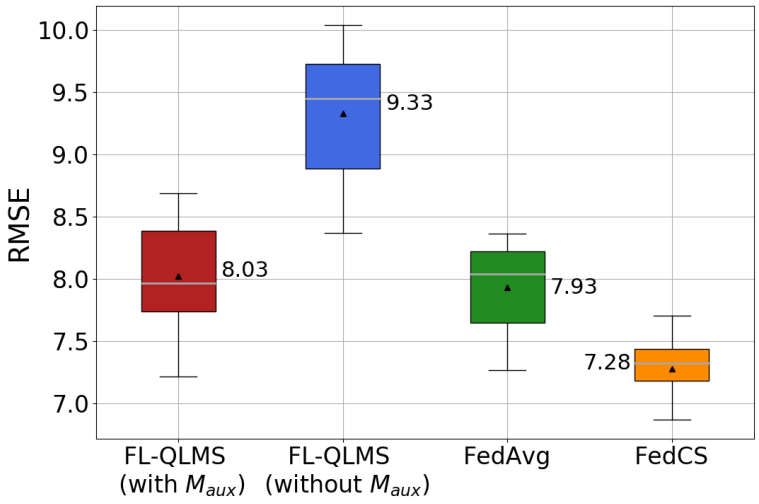


Figure 5.5: **Comparison among FedAvg, FedCS [135], and the proposed FL-QLMS (w/o the auxiliary model). In this experiment, a Non-IID setting is considered. An average RMSE is shown beside each boxplot.**

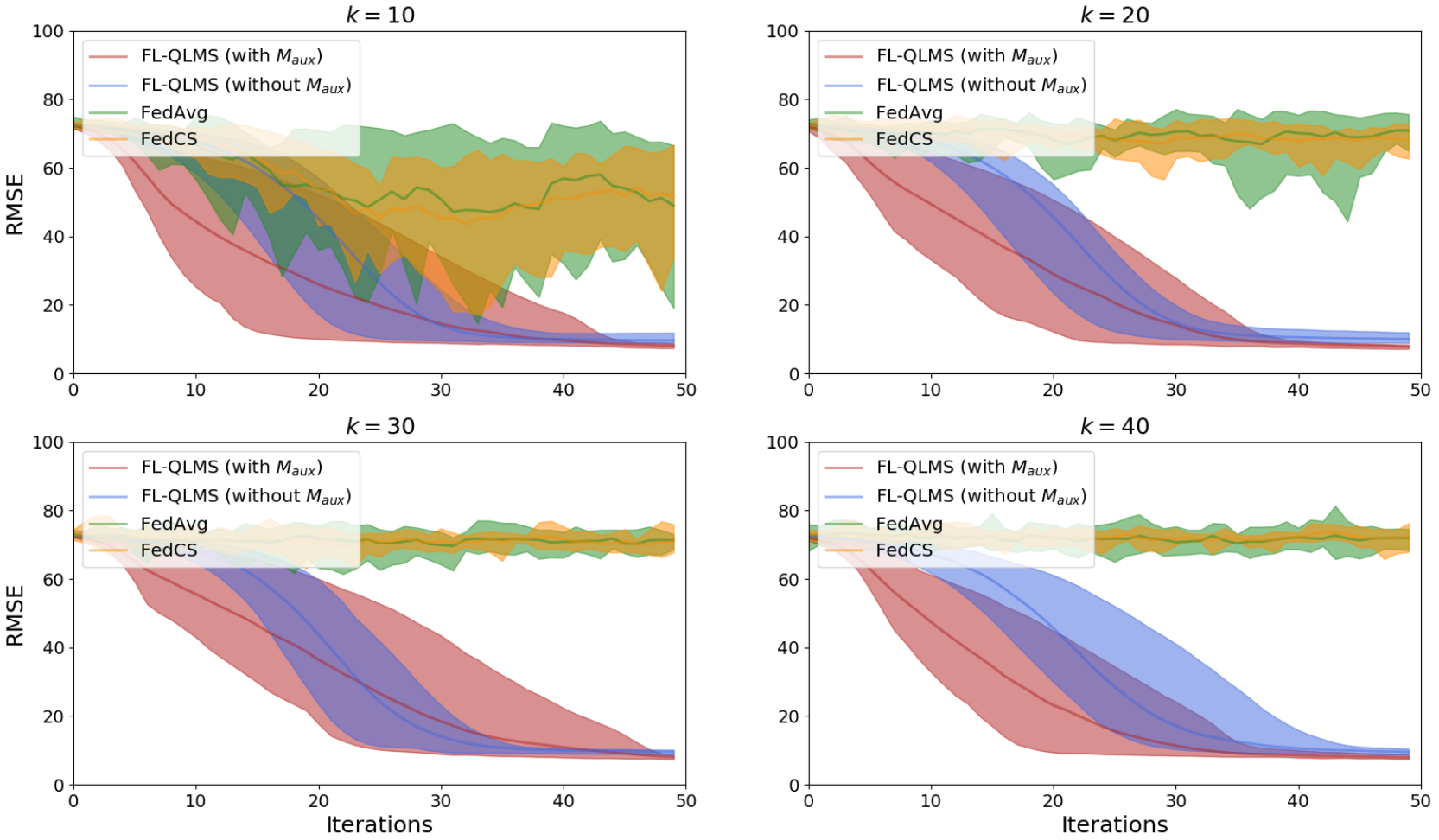


Figure 5.6: **Comparison among FedAvg, FedCS, and the proposed FL-QLMS (w/o the auxiliary model) under client attacks. Different severity of the attack is considered. Shaded regions denote the fluctuation of the performance.**

5.4 Chapter Summary

This chapter presents the robust federated learning algorithm for qualified local model selection (FL-QLMS). There are two versions of FL-QLMS, named FL-QLMS with auxiliary data and FL-QLMS without auxiliary data. The first version uses an auxiliary dataset on the aggregator and trains an auxiliary model. Local models that are more similar to the auxiliary model based on the model parameters are considered more representative. The second version is based on the similarity between local models. The evaluation results demonstrate that the proposed FL-QLMS achieves high robustness when 10% to 40% of the models are attacked. Compared to state-of-the-art methods, FL-QLMS maintains the training performance throughout the learning phase. Next chapter will presents two variants of the proposed blockchain architecture for collaborative learning.

6

Blockchain-Enabled Collaborative Learning

The whole process of the decentralized system is divided into four main steps: (1) The local models are trained and then uploaded to the blockchain network. (2) In the blockchain network, the whole 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 local models. (4) Once an edge node collects the models, a global model is created and replaces the current local model. The whole process is repeated for each node until local convergence is achieved.

Algorithm 6.1: Decentralized FL-based learning scheme based on blockchain and Swarm platform

Require: Real-time data collected from EV
Ensure: Predicted Power Consumption
1: Initialize the aggregator to wait for collecting local models
2: **While** do not converge **do**
3: Train local model M^i_{local} in edge AI system
4: Upload M^i_{local} from AI system to Swarm for storing, obtain hashed value $h_i(addr)$ with respect to the address of stored model
5: 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}
6: Broadcast a request to the whole blockchain network
7: As soon as TX_{ij} is verified, it will be added to the mempool
8: All transactions in a mempool is packaged and then added to the block
9: Mining begins. A successful mined block is added to the public ledger
10: Participants download local models from the public ledger to update the global model as $M_{global} \leftarrow \frac{1}{n}\sum_{i=1}^{n} \nabla M^i_{local}$
11: $M^i_{local} \leftarrow M_{global}$ /* Update local model
12: **End While**
13: **Return** Predicted Power Consumption

6.1 Collaborative Learning based on Blockchain and Swarm Platform

As illustrated in Fig. 4.5, 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 [155] 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. 6.1.

To implement the blockchain network, we start by creating user accounts on Ethernet. Then we initialize the nodes in the Swarm, which is used to allocate memory to each client to store the data, as shown in Fig. 6.1. In the Swarm cluster, discontinuous storage is allocated to each client. Each time a client uploads data, it is partitioned into many segments, which are then

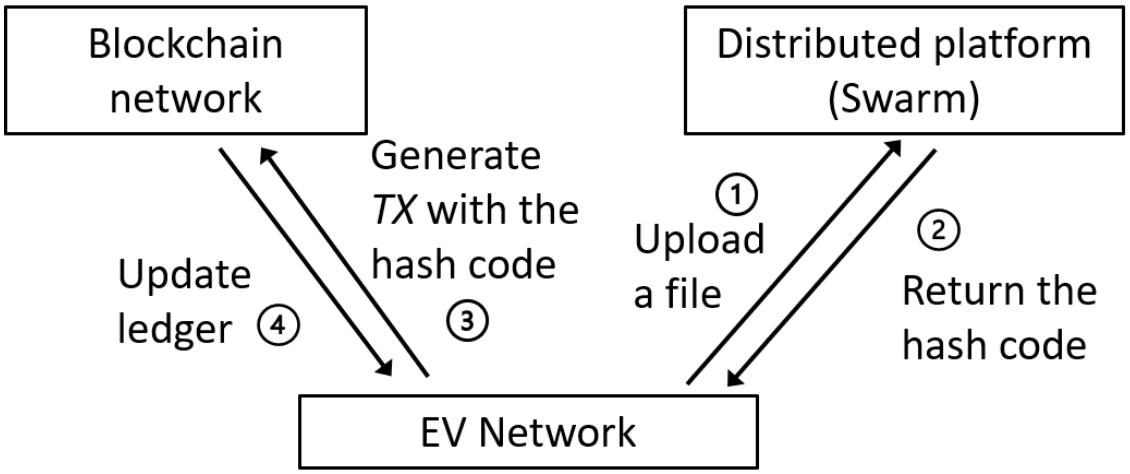


Figure 6.1: **The communication between the blockchain network and the Swarm platform. The distributed storage platform is used to record user information and the updated files.** *TX*: **Transaction.**

stored in different volumes. After that, the cluster generates a hash code representing the address corresponding to the collection of data fragments. Based on the information of the user account and this hash code, access to the complete data is possible. In this work, the data refers to the training model with the parameters of the network.

At the blockchain side, when a client, C_1, uploads a file to Swarm and receives the encrypted hash code, it sends the hash value to another node, C_2, automatically creating a new transaction. However, to ensure that this transaction is trustworthy, digital signatures are used for verification. The transaction pool is where all valid transactions wait to be confirmed by the blockchain network. However, with the increase in unconfirmed transactions, memory consumption and computational efficiency become a challenge. To tackle this problem, the Merkle tree [113] was introduced, which significantly reduces the requirement concerning both memory and computation as shown in Fig. 2.5.

The entire process of building a Merkle tree results 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 goal is to find a nonce that satisfies

Table 6.1: **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)

Table 6.2: **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

the requirement shown in equation 6.2.

$$Hash(M + nonce) = \underbrace{0...0}_{n\ bits} x...x \tag{6.1}$$

Once a perfect nonce is found, it is added to the hashed block. The block header is re-hashed along with the successful nonce, then the block, including header and body, is added to the chain. It is worth noting that in our case a relatively high frequency of information exchange between ledgers is required. Therefore, n is chosen small to pave the way for mining. With the public information in the blockchain, each node can quickly retrieve the data used to access Swarm. This allows them to download the latest 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 6.1.

6.2 Writing Metadata in Transactions

In general, the return operator (OP_RETURN), which is part of the Bitcoin script language, is used to allow metadata to be stored on the blockchain [156]. However, the limit for storing data in an OP_RETURN is a maximum of 83 bytes according to release 0.12.0 [156]. This reveals a significant 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 generates 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 number of parameters is the sum of the number of weights and biases, i.e., $11 \times 8 + 8 \times 6 + 6 \times 1 + (8 + 6 + 1) = 157$. Each parameter in floating point format occupies 4 bytes, so if we extract only the parameters from the model, the data size is $157 \times 4 = 628$ bytes. At least eight transactions are required for each model. With an enormous model size, the increased number of transactions leads to a significant degradation in storage and computation efficiency. An alternative way to store data is to utilize distributed cloud storage instead of Swarm. However, there is still a high risk of data leakage here, although the apparent simplicity offers the advantage. A comparison of the above methods is summarized in Table 6.2.

6.3 Secure Semi-decentralized FL-based Framework

As we explained in the previous section, the proposed system is based on a semi-decentralized architecture. As shown in Fig. 6.2, the solid black lines mean that the local models are uploaded from the clients to the aggregator. This communication does not take place in the blockchain. Other activities indicated by dashed lines in blue belong to the blockchain network. A VPP aggregator, EV fleets, and a group of miners are integrated into the blockchain network. In the

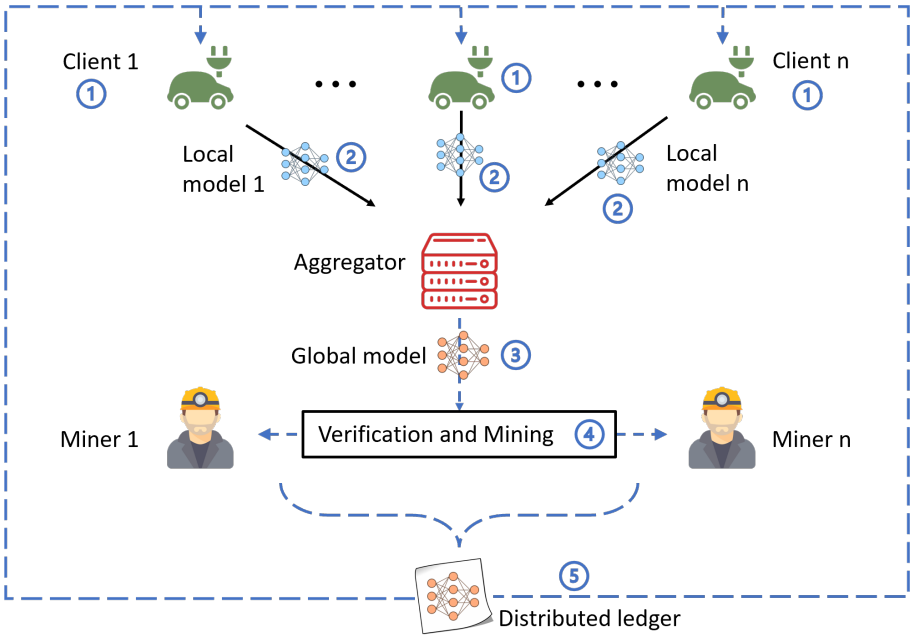


Figure 6.2: **Overview of the proposed secure semi-decentralized FL -based framework. The black solid lines mean that the local models are uploaded from the clients to the aggregator. This communication does not take place in the blockchain. The activities in the blockchain network are indicated by blue dashed lines. A VPP aggregator, EV fleets, and a group of miners are integrated into the blockchain network. The workflow is briefly divided into five steps: 1) Each EV node trains a local model. From the second round of training, each EV node updates the local model until convergence. 2) Each EV node uploads the local model to the aggregator. 3) We apply the FL-QLMS algorithm to select the qualified models for aggregation, resulting in a global model. 4) The aggregator creates and broadcasts a transaction (containing the global model) in the blockchain. After validation and mining, a distributed ledger is created. 5) Each client downloads the global model from the distributed ledger to update the model.**

proposed architecture, the miners are the vehicles themselves, while in Fig. 6.2 EVs and miners are shown separately for explanation. The overall workflow for each training round is described as follows:

1. In the first training round, each EV node initializes and trains a local model M_{local}. From the second training round, each EV node updates the local model until convergence.

2. Each EV node uploads the local model M_{local} to the aggregator.

3. After collecting local models, we apply the FL-QLMS algorithm to the model selection process. Then, the qualified models are selected for aggregation, resulting in a global

model M_{global}.

4. (a) At first, the global model is recorded as metadata in a new transaction TX_0. The aggregator feeds the transaction into a the hash function H and generates a hash value $H(TX_0)$.

   (b) The aggregator feeds $H(TX_0)$ to a signature algorithm with aggregator's private key, whereby an encrypted message is produced.

   (c) The aggregator then creates a transaction TX that contains the original transaction TX_0, the encrypted message and a public key.

   (d) The transaction will be sent from the aggregator to one of the nodes and then broadcasted to all miners.

   (e) Each miner can start performing validation. One will use the same hash function H and generate the hash value of TX_0. We denote the hash value by H_1. Since the same hash function always produces the same output, H_1 should be identical to $H(TX_0)$. Besides, the encrypted message will then be decrypted using the public key. If the resulted value matches H_1, the digital signature is proven to be valid. Therefore, TX is considered valid and added to each node's transaction pool. Once TX is confirmed by the blockchain network, it is added to the block.

   (f) A block header contains a 32-Byte previous block hash, 32-Byte Merkle root, 4-Byte timestamp, 4-Byte difficulty target, and 4-Byte nonce. A nonce is a 32-bit target that is guessed by miners by solving the following equation:

$$H(nonce) = \underbrace{0\ldots0}_{n\ bits} x_{n+1}\ldots x_{256} \tag{6.2}$$

   where, n is a pre-determined value controlling the mining difficulty.

   (g) Once the nonce is found, the mined block is added to the distributed ledger.

5. Each client downloads the global model from the distributed ledger for model update.

The local model is transmitted and merged without blockchain support. To ensure the robustness of the model aggregation, we introduce a novel algorithm called Federated Learning for Qualified Local Model Selection (FL-QLMS) in Section 5.2. With this, the *fake* models are excluded and thus do not affect the model aggregation. The proposed semi-decentralized FL -based platform drastically reduces blockchain congestion while maintaining a high level of system security. A functionality comparison between the decentralized (i.e., AEBIS) and the proposed semi-decentralized (i.e., NoEV) systems can be found in Table 6.1.

6.4 Evaluation

6.4.1 Blockchain Network on Swarm Platform

Evaluation Methodology

For the blockchain solution, we used Geth [157], 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, creating a genesis file for each node. A genesis file contains the entire configuration of the initial states, with information about several important parameters described in Table 6.3.

Table 6.3: **Key configurations in a genesis file.**

	Description
ChainID	ID of the chain (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

Evaluation Results

For each node, multiple accounts are created with a private and public key pair, with the first account used for mining and associated with the node's address. The address is derived from 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. Then, a local directory with a specific

Swarm ID is automatically created. When both services are set up on blockchain and Swarm, the communication between clients, Swarm and blockchain starts correctly (refer to 6.1).

6.4.2 Semi-decentralized FL-based Framework

Evaluation Methodology

To show the advantage of our proposed system in terms of cost efficiency, we studied the network load in a blockchain system and compared the proposed NoEV with AEBIS, oVML, and DeepChain. We mainly focused on the number of blocks and transactions generated in a given time period. We used an extensible simulation tool *BlockSim* for blockchain systems presented in [158]. The configurations are summarized in Table 6.4. We simulated 63 nodes for AEBIS, oVML, and DeepChain, and 63+1 nodes (1 additional node for the aggregator) for NoEV. We performed ten runs for each simulation, with each run lasting 6000 seconds.

Table 6.4: **Configuration for *BlockSim* simulation.**

Parameters	**Value**	**Description**
TI	30, 60, 120 seconds	Time interval of block generation
B_{size}	1 MB	Block size
B_{delay}	1, 3, 6, 12 seconds	Block propagation delay
T_{size}	650 bytes	Transaction size
Nodes	**Hash Power**	**Description**
N_0	1.587%	In total, 63 nodes (miners) are considered in AEBIS,
N_1	1.587%	oVML and DeepChain.
$\cdots$	$\cdots$	Each miner has the same computing power.
N_{62}	1.587%	For NoEV, the aggregator acts as an
N_{63}	0%	additional node with a hash power of 0.

Note: We collected the data set regarding 63 cities. Therefore, to simply the preparation of data set allocation and federated learning schedule, we simulate 63 EV nodes for model training. Besides, in our blockchain proposal, each EV nodes also acts as a miner, thus we use 63 nodes in this work. To fairly compare the proposed work with other state-of-the-art works, we use 63 nodes for AEBIS, oVML and DeepChain too.

Evaluation Results

The block size was set to 1 megabyte (MB). We considered different combinations of TI and B_{delay}, which represent the average time to generate a new block and the propagation delay of a block, respectively. In [158], the transaction size T_{size} is 572.5 bytes by default. In our ex-

periment, T_{size} is larger because each transaction must additionally store a portion of a model. The total number of parameters for our fully connected network (11-8-6-1) is 157. Each parameter in floating-point format occupies 4 bytes; thus, if we extract the parameters from the model, the total size is $157 \times 4 = 628$ bytes. In general, the return operator (OP_RETURN), which is part of the Bitcoin script language, is used to allow storing metadata on the blockchain with a maximum storage limit of 83 bytes according to release 0.12.0 [156]. Therefore, at least eight transactions are required for each model. The updated transaction size T_{size} is 572.5 bytes + 628/8 bytes = 650 bytes. We implemented 63 nodes (N_0 to N_{62}) for AEBIS, oVML, and DeepChain simulation with respect to a total of 63 EV clients. For simplicity, we consider a simple scenario that each miner has the same hash power. Therefore, given 63 nodes and the total hash power of 1, each of them will have a hash power of approximately 1.587%. For the NoEV simulation, the aggregator is introduced as an additional node N_{63}. Since N_{63} is not assigned any mining task, its hash power is set to 0%. We assume that the number of transactions (T_n) created per second is eight in NoEV. Accordingly, $T_n = 8 \times 63 = 504$ in AEBIS since 63 nodes are considered.

Table 6.5 summarizes the results of AEBIS, NoEV, oVML, and DeepChain on the BlockSim simulator. As the average block interval increases, the total number of blocks decreases accordingly. In addition, as the block propagation delay increases, the number of blocks included in the main chain decreases, while the number of stale blocks increases. The stale blocks have been successfully mined but are not included in the current best chain. Therefore, the overall rate of stale blocks increases. When comparing with other methods, it is observed that NoEV generally requires the fewest transactions, especially for short TI. For example, for a short block interval ($TI = 30$) and short block propagation delay ($B_{delay} = 1$), NoEV requires an average of 25166 transactions, which is 38%, 37%, and 35% less than AEBIS, oVML, and DeepChain, respectively. The significant decrease in NoEV can be explained by the lower number of transactions, since NoEV requires only one global model transmission to the block, while the other methods require frequent local model transmission. DeepChain averaged model updates every 10 to 20 iterations rather than at every iteration to increase communication efficiency, as in AEBIS and

oVML. However, DeepChain and oVML still require the exchange of local models over the blockchain network.

Table 6.5: **The blockchain simulation results of AEBIS, NoEV, oVML, and DeepChain for different combinations of parameters.**

Parameters		AEBIS [147]					NoEV (this work)				
TI	B_{delay}	B_{total}	B_{main}	B_{stale}	r_s	TX	B_{total}	B_{main}	B_{stale}	r_s	TX
30	1	196.4	190.8	5.6	2.9%	40619	200.1	193.9	6.3	3.12%	25166
	3	200.25	182.9	17.4	8.7%	38141	197	180.6	16.4	8.31%	24532
	6	197.5	170.1	27.4	13.9%	33425	209.5	176.3	33.3	15.87%	23299
	12	194.5	148.4	46.1	23.7%	30467	203.9	155.4	48.5	23.79%	20560
60	1	103.4	102	1.4	1.3%	21296	98.5	96.9	1.6	1.7%	17201
	3	104.1	97.5	6.6	6.4%	19851	100.8	95.4	5.4	5.33%	16811
	6	102.1	94.5	7.6	7.5%	19925	100.9	93.3	7.8	7.68%	15782
	12	100.1	84.8	15.4	15.4%	18160	106.8	90	16.8	15.7%	15287
120	1	46.8	46.1	0.6	1.3%	9031	48.4	48.4	0	0.00%	8727
	3	50.2	48.6	1.6	3.2%	10021	51	49.6	1.4	2.7%	9620
	6	52.6	49.9	2.8	5.2%	11673	50	47.1	2.9	5.8%	8768
	12	55.3	50.4	4.9	8.8%	10419	50.4	46.5	3.9	7.7%	9086
Parameters		**oVML [60]**					**DeepChain [130]**				
TI	B_{delay}	B_{total}	B_{main}	B_{stale}	r_s	TX	B_{total}	B_{main}	B_{stale}	r_s	TX
30	1	196.6	191.9	4.8	2.4%	39805	198.3	192.6	5.6	2.8%	38807
	3	192.4	175.6	16.8	8.7%	36892	195.8	180.5	15.3	7.8%	35937
	6	193.4	164.8	28.6	14.8%	37477	197.6	167.9	29.8	15.1%	30397
	12	203.1	152.1	51	25.1%	30204	203	154.5	48.5	23.9%	29601
60	1	101	99.6	1.4	1.4%	21750	102.1	100.1	2	2.0%	21224
	3	101.1	96.4	4.8	4.7%	20652	103.3	98	5.3	5.1%	18598
	6	103.6	93.5	10.1	9.8%	19640	95	86.5	8.5	9.0%	17374
	12	95.6	81.5	14.1	14.8%	16767	97.3	84.9	12.4	12.7%	16510
120	1	50	49.6	0.4	0.8%	8418	49.3	48.8	0.5	1.0%	10375
	3	50.5	49.1	1.4	2.7%	10623	50.6	50	0.6	1.2%	9504
	6	51.1	48.4	2.8	5.4%	9446	52.1	48.9	3.3	6.2%	9741
	12	48.8	44.1	4.6	9.5%	8522	48.3	44.4	3.9	8.0%	9284

B_{total}: The total amount of blocks generated.
B_{main}: The number of blocks included in the main chain.
B_{stale}: Blocks that were successfully mined but not included in the current best chain.
r_s: Stale block rate.
TX: Transactions.

6.5 Chapter Summary

In this chapter, we propose two architectures for federated learning based on blockchain. The first proposal leverages the conventional blockchain with the Swarm platform, where the

hash values of the local models are stored on the blockchain instead of the models themselves. The second proposal introduces an aggregator to the conventional blockchain, where the local models are only transmitted to the aggregator and not to the blockchain. Both of these works help reduce the heavy load on the blockchain and improve the efficiency of the system without compromising security. The next chapter summarizes the thesis and discusses the remaining issues and future research directions.

7

Thesis Summary and Discussion

We conclude this dissertation with a summary chapter where we summarize the main contributions of this research. We discuss the results of the conducted simulation. Finally, we conclude this dissertation with a discussion of how this work can be improved, as well as other considerations not addressed in this dissertation.

7.1 Contributions Summary

In this thesis, we propose a trustworthy AI-based system and algorithms for power management in network of electric vehicles.

We establish a novel communication mechanism between the aggregator and each EV nodes using an AI system based on reconfigurable hardware (FPGA) to predict the amount of avail-

able power that 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 a vehicle's controller area network (CAN) bus.

The proposed EV charging mechanism incorporates a new EV battery power consumption prediction algorithm based on a fully-connected neural network model. The prediction of power consumption is performed by dividing a long trip into multiple sections. Each small section is associated with a list of features that are used as inputs to the network.

Taking a step further, to guarantee the model learning in an efficient and secure way, we introduce a robust collaborative learning scheme that integrates federated learning and blockchain technology. We proposed an algorithm called federated learning for qualified learning model selection (FL-QLMS) that is robust to both data and model attacks. The FL-QLMS is performed in each training round to find a group of the best local models and filter out the malicious or disqualified models. In addition, the novel blockchain architecture consists of a VPP aggregator and an EV fleet, and only global models are transmitted to the blockchain. The local models are collected on the aggregator side in an off-chain manner.

7.2 Results Summary

In this research, we focused on the performance of the proposed prediction of electric vehicle power consumption. We also focused on the performance of the proposed Fl algorithm and blockchain architecture.

The proposed multi-stage PCP shows better performance in scenarios with short-distance journey. Besides, the multi-stage PCP achieves a greater advantage in long-distance travel scenarios. We also analyzed the performance variation of the two methods in each case. For medium and long distances, the variance of the RMSE of the multi-stage PCP is significantly larger than that of the PCP. The multi-stage approach may explain the reason for this. In the multi-stage PCP, when the distance is long, the trip is first divided into several sections and then the prediction model is run for each section. When the prediction results are summed, the er-

rors caused by each prediction are also accumulated. Therefore, the multi-stage PCP leads to higher variability. On the other hand, for a short trip, e.g., one or two hours, the multi-stage approach has little effect, so the variance of the multi-stage PCP is lower.

We also studied the impact of hacked clients on various FL algorithms with non-IID data. FL-QLMS (with M_{aux}) proves robust when 10% to 40% of clients are hacked, holding average performance constant. In contrast, FedAvg and FedCS are highly sensitive to attacks, as the training process hardly converges as the number of faked models increases. For FL-QLMS (without M_{aux}), it always leads to convergence, but with slightly worse performance than FL-QLMS (with M_{aux}).

We summarize the results of AEBIS, NoEV, oVML, and DeepChain on the BlockSim simulator. As the average block interval increases, the total number of blocks decreases accordingly. Moreover, as the block propagation delay increases, the number of blocks included in the main chain decreases, while the number of stale blocks increases. The stale blocks have been successfully mined but are not included in the current best chain. Therefore, the overall rate of stale blocks increases. When compared to other methods, it is observed that NoEV generally requires the fewest transactions, especially for short *TI*. The significant decrease in NoEV can be explained by the fewer number of transactions, since NoEV requires only one global model transmission on the block, while the other methods require frequent local model transmission. DeepChain averaged model updates every 10 to 20 iterations rather than at every iteration to increase communication efficiency, as in AEBIS and oVML. However, DeepChain and oVML still require the exchange of local models over the blockchain network. The results demonstrate that the proposed NoEV blockchain architecture allows the entire system to maintain a high level of security while significantly increasing the efficiency of the blockchain network.

7.3 Discussion

The presented network of electric vehicles considers the same type of electric vehicles. Since different types of electric vehicles may have different impacts due to different battery capacities, charging speeds, driving behaviors, etc., the discharge allocation mechanism needs to be

redesigned in the face of such a complicated scenario. Based on the proposed multi-stage power consumption method, the use of the global positioning system will help contribute to more realistic route planning. In addition, different road conditions in different locations may affect power consumption. Air resistance, surface resistance, and high or low battery temperatures also significantly affect battery performance. Therefore, these variables should be considered in future research. It is expected that the proposed method can be applied to other types of engines, vehicles, trams, and trains given a specific driving task.

Moreover, the efficient division of the whole trip into several sections remains a problem to be optimized. In addition, there are a number of factors that have not yet been considered in our research that can have a large impact on energy consumption, such as the vehicle model, the age of the vehicles, the driving style of the driver, etc. Since any change will affect the prediction, we would like to regularize the data set and propose a new learning scheme that is compatible with new data features.

Although the proposed FL and blockchain-based architecture shows great potential for efficient and robust collaborative learning, there are still some challenges that need to be pointed out. First, generic neural networks have a large scale that cannot be stored in a single transaction in practice. Given the theoretical limit of 4 MB for current transactions, dealing with large models remains a problematic issue. One possible solution is to first apply knowledge distillation, a model compression method in which a small model is trained to mimic a pre-trained larger model, and then the model is divided into a group of small segments. Each segment is fit into a single transaction, and the model segments are finally reconstructed by local devices. Second, current studies, including this work, use a large, single blockchain that integrates a large portion of local devices. Such architectures have the problem of high latency, high maintenance cost, vulnerability to single point of failure attacks, etc. To solve this problem, we are aiming for a multi-blockchain architecture in the future. The new architecture is to divide the original computing society into a group of clusters. Each cluster is responsible for a part of the work and communicates with the others. In this way, we could achieve a system with higher efficiency and scalability. Third, an incentive mechanism is preferable in vehicular networks, especially

when private vehicles are considered in the edge computing scenario. To attract more participants for data sharing, model training, and block mining, a series of reward mechanisms can be proposed to provide incentives for private car owners. In addition, based on the incentive mechanism, we will propose a novel trading system to enable smooth energy trading for the V2G network.

Besides, qualified local model selection is essential to ensure the robustness of federated learning. The FL-QLMS algorithm demonstrates robustness to model attacks during the federated process. However, the performance of the current FL-QLMS algorithm is highly dependent on a prepared auxiliary data set, which raises two critical issues. First, the supplemental data should ideally have the same distribution, as the entirety of the data is not guaranteed. Moreover, since the client-side data is updated daily, the auxiliary information is unreliable for local model selection. Second, edge nodes must not pass raw data to the server for privacy and security reasons.

References

[1] *Renewable Capacity Statistics 2022.* (accessed February 3, 2023). [Online]. Available: https://www.irena.org/publications/2022/Jul/Renewable-Energy-Statistics-2022

[2] *Road transport: Reducing CO2 emissions from vehicles.* (accessed February 3, 2023). [Online]. Available: https://ec.europa.eu/clima/eu-action/transport-emissions/road-transport-reducing-co2-emissions-vehicles/co2-emission-performance-standards-cars-and-vans_en

[3] Y. Lin, M. Yang, C. Wan, J. Wang, and Y. Song, "A multi-model combination approach for probabilistic wind power forecasting," *IEEE Transactions on Sustainable Energy*, vol. 10, no. 1, pp. 226–237, 2018, doi: 10.1109/TSTE.2018.2831238.

[4] H. D. Tafti, A. I. Maswood, G. Konstantinou, J. Pou, and F. Blaabjerg, "A general constant power generation algorithm for photovoltaic systems," *IEEE Transactions on Power Electronics*, vol. 33, no. 5, pp. 4088–4101, 2017, doi: 10.1109/TPEL.2017.2724544.

[5] M. Chazarra, J. I. Pérez-Díaz, and J. Garcia-Gonzalez, "Optimal join energy and secondary regulation reserve hourly scheduling of variable speed pumped storage hydropower plants," *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 103–115, 2017, doi: 10.1109/TPWRS.2017.2699920.

[6] 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 transactions on smart grid*, vol. 9, no. 6, pp. 6806–6814, 2017, doi: 10.1109/TSG.2017.2723845.

[7] A. B. Abdallah, M. Hisada, "Virtual Power Platform Control System," Japanese Patent Application No. 2020-033678 (2020.02.28), Patent Nbr. 6804072.

[8] D. Koraki and K. Strunz, "Wind and solar power integration in electricity markets and distribution networks through service-centric virtual power plants," *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 473–485, 2017, doi: 10.1109/TPWRS.2017.2710481.

[9] N. Ruiz, I. Cobelo, and J. Oyarzabal, "A direct load control model for virtual power plant management," *IEEE Transactions on Power Systems*, vol. 24, no. 2, pp. 959–966, 2009, doi: 10.1109/TPWRS.2009.2016607.

[10] E. Dall'Anese, S. S. Guggilam, A. Simonetto, Y. C. Chen, and S. V. Dhople, "Optimal regulation of virtual power plants," *IEEE Transactions on Power Systems*, vol. 33, no. 2, pp. 1868–1881, 2017, doi: 10.1109/TPWRS.2017.2741920.

[11] G. Zhang, C. Jiang, and X. Wang, "Comprehensive review on structure and operation of virtual power plant in electrical system," *IET Generation, Transmission & Distribution*, vol. 13, no. 2, pp. 145–156, 2018, doi: 10.1049/iet-gtd.2018.5880.

[12] *Limejump.* (accessed February 3, 2022). [Online]. Available: https://theenergyst.com/limejump-aggregators-will-need-an-electricity-supply-licence-to-survive/

[13] 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," *Automation of Electric Power Systems*, vol. 36, no. 21, pp. 1–6, 2012.

[14] M. Wache and D. Murray, "Application of synchrophasor measurements for distribution networks," in 2011 IEEE Power and Energy Society General Meeting, Detroit, MI, USA, 24-28 July 2011; pp. 1–4, doi: 10.1109/PES.2011.6039337.

[15] Masayuki Hisada, Abderazek Ben Abdallah, "Virtual Power Platform Control System," Japanese Patent Application Laid-Open No 2019-58007.

[16] 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 Transactions on Sustainable Energy*, vol. 7, no. 2, pp. 562–575, 2015, doi: 10.1109/TSTE.2015.2504561.

[17] S. Camal, A. Michiorri, and G. Kariniotakis, "Optimal offer of automatic frequency restoration reserve from a combined pv/wind virtual power plant," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6155–6170, 2018, doi: 10.1109/TPWRS.2018.2847239.

[18] M. A. Mirzaei, M. N. 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," *Applied Thermal Engineering*, pp. 115413, 2020, doi: 10.1016/j.applthermaleng.2020.115413.

[19] 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," *International Journal of Electrical Power & Energy Systems*, vol. 123, pp. 106293, 2020, doi: 10.1016/j.ijepes.2020.106293.

[20] M. A. Mirzaei, A. Sadeghi-Yazdankhah, B. Mohammadi-Ivatloo, M. Marzband, M. Shafie-khah, and J. P. Catalão, "Integration of emerging resources in igdt-based ro-

bust scheduling of combined power and natural gas systems considering flexible ramping products," *Energy*, vol. 189, pp. 116195, 2019, doi: 10.1016/j.energy.2019.116195.

[21] M. Jadidbonab, B. Mohammadi-Ivatloo, M. Marzband, and P. Siano, "Short-term self-scheduling of virtual energy hub plant within thermal energy market," *IEEE Transactions on Industrial Electronics*, pp. 1-1, 2020, doi: 10.1109/TIE.2020.2978707.

[22] M. G. Vayá and G. Andersson, "Self scheduling of plug-in electric vehicle aggregator to provide balancing services for wind power," *IEEE Transactions on Sustainable Energy*, vol. 7, no. 2, pp. 886–899, 2015, doi: 10.1109/TSTE.2015.2498521.

[23] 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 Transactions on Power Systems*, vol. 34, no. 3, pp. 1881–1894, 2018, doi: 10.1109/TPWRS.2018.2883753.

[24] 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 Transactions on Power Systems*, vol. 31, no. 5, pp. 3506–3515, 2015, doi: 10.1109/TPWRS.2015.2496551.

[25] M. Shamshirband, J. Salehi, and F. S. Gazijahani, "Look-ahead risk-averse power scheduling of heterogeneous electric vehicles aggregations enabling v2g and g2v systems based on information gap decision theory," *Electric Power Systems Research*, vol. 173, pp. 56–70, 2019, doi: 10.1016/j.epsr.2019.04.018.

[26] W. Wang, P. Chen, D. Zeng, and J. Liu, "Electric vehicle fleet integration in a virtual power plant with large-scale wind power," *IEEE Transactions on Industry Applications*, vol. 56, no. 5, pp. 5924–5931, 2020, doi: 10.1109/TIA.2020.2993529.

[27] 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*

Transactions on Systems, Man, and Cybernetics: Systems, vol. 50, no. 1, pp. 43–57, 2019, doi: 10.1109/TSMC.2019.2896323.

[28] Z. Zhou, B. Wang, Y. Guo, and Y. Zhang, "Blockchain and computational intelligence inspired incentive-compatible demand response in internet of electric vehicles," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 3, no. 3, pp. 205–216, 2019, doi: 10.1109/TSMC.2019.2896323.

[29] 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 of Things Journal*, vol. 7, no. 7, pp. 6288–6301, 2020, doi: 10.1109/JIOT.2020.2966232.

[30] F. Li, J. Qin, and W. X. Zheng, "Distributed q-learning-based online optimization algorithm for unit commitment and dispatch in smart grid," *IEEE transactions on cybernetics*, vol. 50, no. 9, pp. 4146-4156, 2019, doi: 10.1109/TCYB.2019.2921475.

[31] S. Najafi, M. Shafie-khah, P. Siano, W. Wei, and J. P. Catalão, "Reinforcement learning method for plug-in electric vehicle bidding," *IET Smart Grid*, vol. 2, no. 4, pp. 529–536, 2019, doi: 10.1049/iet-stg.2018.0297.

[32] Y. Ye, D. Qiu, M. Sun, D. Papadaskalopoulos, and G. Strbac, "Deep reinforcement learning for strategic bidding in electricity markets," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1343–1355, 2019, doi: 10.1109/TSG.2019.2936142.

[33] H. Shi, M. Xu, and R. Li, "Deep learning for household load forecasting—a novel pooling deep rnn," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 5271–5280, 2017, doi: 10.1109/TSG.2017.2686012.

[34] A. Rosato, M. Panella, R. Araneo, and A. Andreotti, "A neural network based prediction system of distributed generation for the management of microgrids," *IEEE*

Transactions on Industry Applications, vol. 55, no. 6, pp. 7092–7102, 2019, doi: 10.1109/TIA.2019.2916758.

[35] M. Kuzlu, U. Cali, V. Sharma, and Ö. Güler, "Gaining insight into solar photovoltaic power generation forecasting utilizing explainable artificial intelligence tools," *IEEE Access*, vol. 8, pp. 187 814–187 823, 2020, doi: 10.1109/ACCESS.2020.3031477.

[36] J. Lee, W. Wang, F. Harrou, and Y. Sun, "Wind power prediction using ensemble learning-based models," *IEEE Access*, vol. 8, pp. 61 517–61 527, 2020, doi: 10.1109/ACCESS.2020.2983234.

[37] 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 Transactions on Smart Grid*, vol. 11, no. 4, pp. 3201-3211, 2020, doi: 10.1109/TSG.2020.2971427.

[38] R. Lu, S. H. Hong, and M. Yu, "Demand response for home energy management using reinforcement learning and artificial neural network," *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6629–6639, 2019, doi: 10.1109/TSG.2019.2909266.

[39] Z. Wan, H. Li, H. He, and D. Prokhorov, "Model-free real-time ev charging scheduling based on deep reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5246–5257, 2018, doi: 10.1109/TSG.2018.2879572.

[40] J. Zhao, Q. Li, Y. Gong, and K. Zhang, "Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7944–7956, 2019, doi: 10.1109/TVT.2019.2917890.

[41] C. Sonmez, C. Tunca, A. Ozgovde, and C. Ersoy, "Machine learning-based workload orchestrator for vehicular edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2239–2251, 2020, doi: 10.1109/TITS.2020.3024233.

[42] V. Chamola, A. Sancheti, S. Chakravarty, N. Kumar, and M. Guizani, "An IoT and edge computing based framework for charge scheduling and EV selection in V2G systems," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 10 569–10 580, 2020, doi: 10.1109/TVT.2020.3013198.

[43] Y. Huang, Y. Lu, F. Wang, X. Fan, J. Liu, and V. C. Leung, "An edge computing framework for real-time monitoring in smart grid," in 2018 IEEE International Conference on Industrial Internet (ICII), Seattle, WA, USA, 21-23 October 2018; pp. 99–108, doi: 10.1109/ICII.2018.00019.

[44] H. Ko, S. Pack, and V. C. Leung, "Mobility-aware vehicle-to-grid control algorithm in microgrids," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2165–2174, 2018, doi: 10.1109/TITS.2018.2816935.

[45] G. Qiao, S. Leng, K. Zhang, and Y. He, "Collaborative task offloading in vehicular edge multi-access networks," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 48–54, 2018, doi: 10.1109/MCOM.2018.1701130.

[46] *Toshiba Energy Systems & Solutions Corporation.* (accessed February 3, 2023). [Online]. Available: https://www.toshiba-energy.com/en/renewable-energy/product/vpp.htm

[47] Y. M. Saputra, D. T. Hoang, D. N. Nguyen, E. Dutkiewicz, M. D. Mueck, and S. Srikanteswara, "Energy demand prediction with federated learning for electric vehicle networks," in 2019 IEEE Global Communications Conference (GLOBECOM), Hawaii, USA, 9-13 December 2019; pp. 1–6, doi: 10.1109/GLOBECOM38437.2019.9013587.

[48] *RX 32-Bit Performance / Efficiency MCUs.* (accessed February 3, 2023). [Online]. Available: https://www.renesas.com/us/en/products/microcontrollers-microprocessors/rx-32-bit-performance-efficiency-mcus

[49] S. Fan, Q. Ai, and L. Piao, "Fuzzy day-ahead scheduling of virtual power plant with optimal confidence level," *IET Generation, Transmission & Distribution*, vol. 10, no. 1, pp. 205–212, 2016, doi: 10.1049/iet-gtd.2015.0651.

[50] G. N. Ericsson, "Cyber security and power system communication—essential parts of a smart grid infrastructure," *IEEE Transactions on Power Delivery*, vol. 25, no. 3, pp. 1501–1507, 2010, doi: 10.1109/TPWRD.2010.2046654.

[51] H. Khurana, M. Hadley, N. Lu, and D. A. Frincke, "Smart-grid security issues," *IEEE Security & Privacy*, vol. 8, no. 1, pp. 81–85, 2010, doi: 10.1109/MSP.2010.49.

[52] P. Li, Y. Liu, H. Xin, and X. Jiang, "A robust distributed economic dispatch strategy of virtual power plant under cyber-attacks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4343–4352, 2018, doi: 10.1109/TII.2017.2788868.

[53] Y. Liu, H. Xin, Z. Qu, and D. Gan, "An attack-resilient cooperative control strategy of multiple distributed generators in distribution networks," *IEEE Transactions on Smart Grid*, vol. 7, no. 6, pp. 2923–2932, 2016, doi: 10.1109/TSG.2016.2542111.

[54] W. Zeng, Y. Zhang, and M.-Y. Chow, "Resilient distributed energy management subject to unexpected misbehaving generation units," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 1, pp. 208–216, 2015, doi: 10.1109/TII.2015.2496228.

[55] M. A. Razzaque, S. M. Cheraghi *et al.*, "Security and privacy in vehicular ad-hoc networks: survey and the road ahead," in *Wireless Networks and Security*. Springer, 2013, pp. 107–132, doi: 10.1007/978-3-642-36169-2_4.

[56] J. Cui, L. Wei, H. Zhong, J. Zhang, Y. Xu, and L. Liu, "Edge computing in VANETs-an efficient and privacy-preserving cooperative downloading scheme," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1191–1204, 2020, doi: 10.1109/JSAC.2020.2986617.

[57] J. A. Onieva, R. Rios, R. Roman, and J. Lopez, "Edge-assisted vehicular networks security," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8038–8045, 2019, doi: 10.1109/JIOT.2019.2904323.

[58] G. Luo, H. Zhou, N. Cheng, Q. Yuan, J. Li, F. Yang, and X. S. Shen, "Software defined cooperative data sharing in edge computing assisted 5G-VANET," *IEEE Transactions on Mobile Computing*, vol. 20, no. 3, pp. 1212–1229, 2019, doi: 10.1109/TMC.2019.2953163.

[59] A. M. Elbir, B. Soner, and S. Coleri, "Federated learning in vehicular networks," *arXiv preprint arXiv:2006.01412*, 2020.

[60] S. R. Pokhrel and J. Choi, "Federated learning with blockchain for autonomous vehicles: Analysis and design challenges," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4734–4746, 2020, doi: 10.1109/TCOMM.2020.2990686.

[61] M. S. Alvim, M. E. Andrés, K. Chatzikokolakis, P. Degano, and C. Palamidessi, "Differential privacy: on the trade-off between utility and information leakage," in International Workshop on Formal Aspects in Security and Trust (FAST 2011), Leuven, Belgium, September 12-14, 2011; pp. 39–54, doi: 10.1007/978-3-642-29420-4_3.

[62] A. Sheikh, V. Kamuni, A. Urooj, S. Wagh, N. Singh, and D. Patel, "Secured energy trading using byzantine-based blockchain consensus," *IEEE Access*, vol. 8, pp. 8554–8571, 2019, doi: 10.1109/ACCESS.2019.2963325.

[63] Y. Li and B. Hu, "A consortium blockchain-enabled secure and privacy-preserving optimized charging and discharging trading scheme for electric vehicles," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 1968–1977, 2020, doi: 10.1109/TII.2020.2990732.

[64] J. Kang, R. Yu, X. Huang, S. Maharjan, Y. Zhang, and E. Hossain, "Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium

blockchains," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3154–3164, 2017, doi: 10.1109/TII.2017.2709784.

[65] L. P. Qian, Y. Wu, X. Xu, B. Ji, Z. Shi, and W. Jia, "Distributed charging-record management for electric vehicle networks via blockchain," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2150–2162, 2020, doi: 10.1109/JIOT.2020.3027482.

[66] V. Hassija, V. Chamola, S. Garg, D. N. G. Krishna, G. Kaddoum, and D. N. K. Jayakody, "A blockchain-based framework for lightweight data sharing and energy trading in V2G network," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 5799–5812, 2020, doi: 10.1109/TVT.2020.2967052.

[67] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1495–1505, 2018, doi: 10.1109/JIOT.2018.2836144.

[68] U. Javaid, M. N. Aman, and B. Sikdar, "A scalable protocol for driving trust management in internet of vehicles with blockchain," *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11 815–11 829, 2020, doi: 10.1109/JIOT.2020.3002711.

[69] D. Gabay, K. Akkaya, and M. Cebe, "Privacy-preserving authentication scheme for connected electric vehicles using blockchain and zero knowledge proofs," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 5760–5772, 2020, doi: 10.1109/TVT.2020.2977361.

[70] L. Li, J. Liu, L. Cheng, S. Qiu, W. Wang, X. Zhang, and Z. Zhang, "CreditCoin: A privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2204–2220, 2018, doi: 10.1109/TITS.2017.2777990.

[71] Y. Wang, Z. Su, and N. Zhang, "BSIS: Blockchain-based secure incentive scheme for energy delivery in vehicular energy network," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3620–3631, 2019. doi: 10.1109/TII.2019.2908497.

[72] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019, doi: 10.1145/3298981.

[73] T. H. Vu, O. M. Ikechukwu, and A. B. Abdallah, "Fault-tolerant spike routing algorithm and architecture for three dimensional noc-based neuromorphic systems," *IEEE Access*, vol. 7, pp. 90 436–90 452, 2019, doi: 10.1109/ACCESS.2019.2925085.

[74] 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 Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 15, no. 4, pp. 1–28, 2019, doi: 10.1145/3340963.

[75] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Cryptography Mailing list at https://metzdowd.com*, 03 2009.

[76] M. F. Akorede, H. Hizam, and E. Pouresmaeil, "Distributed energy resources and benefits to the environment," *Renewable and sustainable energy reviews*, vol. 14, no. 2, pp. 724–734, 2010, doi: 10.1016/j.rser.2009.10.025.

[77] H. Dong, S. Li, H. Dong, Z. Tian, and S. Hillmansen, "Coordinated scheduling strategy for distributed generation considering uncertainties in smart grids," *IEEE Access*, vol. 8, pp. 86 171–86 179, 2020, doi: 10.1109/ACCESS.2020.2992342.

[78] E. A. Martínez Ceseña, T. Capuder, and P. Mancarella, "Flexible distributed multienergy generation system expansion planning under uncertainty," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 348–357, 2016, doi: 10.1109/TSG.2015.2411392.

[79] M. Gough, S. F. Santos, M. Lotfi, M. S. Javadi, G. J. Osório, P. Ashraf, R. Castro, and J. P. Catalão, "Operation of a technical virtual power plant considering diverse distributed energy resources," *IEEE Transactions on Industry Applications*, vol. 58, no. 2, pp. 2547–2558, 2022, doi: 10.1109/TIA.2022.3143479.

[80] B. Li and M. Ghiasi, "A new strategy for economic virtual power plant utilization in electricity market considering energy storage effects and ancillary services," *Journal of Electrical Engineering & Technology*, vol. 16, no. 6, pp. 2863–2874, 2021.

[81] T.-T. Ku, C.-H. Lin, C.-T. Hsu, C.-S. Chen, Z.-Y. Liao, S.-D. Wang, and F.-F. Chen, "Enhancement of power system operation by renewable ancillary service," *IEEE Transactions on Industry Applications*, vol. 56, no. 6, pp. 6150–6157, 2020, doi: 10.1109/TIA.2020.3020782.

[82] E. Mashhour and S. M. Moghaddas-Tafreshi, "Bidding strategy of virtual power plant for participating in energy and spinning reserve markets—part i: Problem formulation," *IEEE Transactions on Power Systems*, vol. 26, no. 2, pp. 949–956, 2010, doi: 10.1109/T-PWRS.2010.2070884.

[83] E. Mashhour and S. M. Moghaddas-Tafreshi, "Bidding strategy of virtual power plant for participating in energy and spinning reserve markets—part ii: Numerical analysis," *IEEE Transactions on Power Systems*, vol. 26, no. 2, pp. 957–964, 2010, doi: 10.1109/T-PWRS.2010.2070883.

[84] *Nissan's Blue Switch.* (accessed February 3, 2023). [Online]. Available: https://global.nissanstories.com/en/releases/nissan-blue-switch

[85] *Mitsubishi Motors i-MiEV.* (accessed February 3, 2023). [Online]. Available: https://www.mitsubishi-motors.com/en/newsrelease/2019/detail1191.html

[86] *The Future of EV Charging is Bidirectional, If You Can Afford It.* (accessed February 3, 2023). [Online]. Available: https://www.wired.co.uk/article/the-future-of-electric-vehicle-charging-is-bidirectional-if-you-can-afford-it

[87] W. Tong, A. Hussain, W. X. Bo, and S. Maharjan, "Artificial intelligence for vehicle-to-everything: A survey," *IEEE Access*, vol. 7, pp. 10 823–10 843, 2019, doi: 10.1109/ACCESS.2019.2891073.

[88] T. Kanade, C. Thorpe, and W. Whittaker, "Autonomous land vehicle project at cmu," in Proceedings of the 1986 ACM fourteenth annual conference on Computer science, Cincinnati, Ohio, USA, 4-6 February 1986; pp. 71–80, doi: 10.1145/324634.325197.

[89] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1826–1848, 2019, doi: 10.1109/TITS.2019.2913998.

[90] *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles.* (accessed February 3, 2023). [Online]. Available: https://www.sae.org/standards/content/j3016_202104/

[91] *Honda Level 3 Automated Driving.* (accessed February 3, 2023). [Online]. Available: https://hondanews.eu/eu/et/cars/media/pressreleases/318975/honda-receives-type-designation-for-level-3-automated-driving

[92] *Toyota Potential Level 4 Automated Mobility.* (accessed February 3, 2023). [Online]. Available: https://global.toyota/en/newsroom/corporate/29933371.html

[93] *Uber Eats and Nuro's Autonomous Food Deliveries.* (accessed February 3, 2023). [Online]. Available: https://www.engadget.com/uber-eats-nuro-autonomous-food-deliveries-texas-california-045758421.html

[94] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management science*, vol. 6, no. 1, pp. 80–91, 1959, doi: 10.1287/mnsc.6.1.80.

[95] E. Bulut and M. C. Kisacikoglu, "Mitigating range anxiety via vehicle-to-vehicle social charging system," in 2017 IEEE 85th Vehicular Technology Conference (VTC Spring), Sydney, Australia, 4-7 June 2017; pp. 1–5, doi: 10.1109/VTCSpring.2017.8108288.

[96] C.-S. Wang, O. H. Stielau, and G. A. Covic, "Design considerations for a contactless electric vehicle battery charger," *IEEE Transactions on industrial electronics*, vol. 52, no. 5, pp. 1308–1314, 2005, doi: 10.1109/TIE.2005.855672.

[97] *Addressing Range Anxiety with Smart Electric Vehicle Routing*. (accessed February 3, 2023). [Online]. Available: https://ai.googleblog.com/2021/01/addressing-range-anxiety-with-smart.html

[98] G. S. Bauer, C. Zheng, S. Shaheen, and D. M. Kammen, "Leveraging big data and coordinated charging for effective taxi fleet electrification: The 100% ev conversion of shenzhen, china," *IEEE Transactions on Intelligent Transportation Systems*, 2021, doi: 10.1109/TITS.2021.3092276.

[99] J. C. Perafan-Villota, O. H. Mondragon, and W. M. Mayor-Toro, "Fast and precise: parallel processing of vehicle traffic videos using big data analytics," *IEEE Transactions on Intelligent Transportation Systems*, 2021, doi: 10.1109/TITS.2021.3109625.

[100] K. Schwenk, S. Meisenbacher, B. Briegel, T. Harr, V. Hagenmeyer, and R. Mikut, "Integrating battery aging in the optimization for bidirectional charging of electric vehicles," *IEEE Transactions on Smart Grid*, vol. 12, no. 6, pp. 5135–5145, 2021, doi: 10.1109/TSG.2021.3099206.

[101] *Toyota Battery Production Investment*. (accessed February 3, 2023). [Online]. Available: https://global.toyota/en/newsroom/corporate/37964997.html

[102] J. H. Park, M. Kim, and D. Kwon, "Security weakness in the smart grid key distribution scheme proposed by xia and wang," *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1613–1614, 2013, doi: 10.1109/TSG.2013.2258823.

[103] M. Amin, F. F. El-Sousy, G. A. A. Aziz, K. Gaber, and O. A. Mohammed, "Cps attacks mitigation approaches on power electronic systems with security challenges for smart grid applications: a review," *IEEE Access*, vol. 9, pp. 38 571–38 601, 2021, doi: 10.1109/ACCESS.2021.3063229.

[104] S.-V. Oprea, A. Bâra, and A. I. Andreescu, "Two novel blockchain-based market settlement mechanisms embedded into smart contracts for securely trading renewable energy," *IEEE Access*, vol. 8, pp. 212 548–212 556, 2020, doi: 10.1109/ACCESS.2020.3040764.

[105] H. Ji, J. Jian, H. Yu, J. Ji, M. Wei, X. Zhang, P. Li, J. Yan, and C. Wang, "Peer-to-peer electricity trading of interconnected flexible distribution networks based on distributed ledger," *IEEE Transactions on Industrial Informatics*, 2021, doi: 10.1109/TII.2021.3137220.

[106] H. Chai, S. Leng, Y. Chen, and K. Zhang, "A hierarchical blockchain-enabled federated learning algorithm for knowledge sharing in internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 3975–3986, 2020, doi: 10.1109/TITS.2020.3002712.

[107] H. Liu, S. Zhang, P. Zhang, X. Zhou, X. Shao, G. Pu, and Y. Zhang, "Blockchain and federated learning for collaborative intrusion detection in vehicular edge computing," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 6073–6084, 2021, doi: 10.1109/TVT.2021.3076780.

[108] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in Artificial Intelligence and Statistics, PMLR, Fort Lauderdale, FL, USA, 20-22 April 2017; pp. 1273–1282.

[109] J. Leng, M. Zhou, J. L. Zhao, Y. Huang, and Y. Bian, "Blockchain security: A survey of techniques and research directions," *IEEE Transactions on Services Computing*, vol. 15, no. 4, pp. 2490-2510, 2020, doi: 10.1109/TSC.2020.3038641.

[110] A. A. Yavuz and M. O. Ozmen, "Ultra lightweight multiple-time digital signature for the internet of things devices," *IEEE Transactions on Services Computing*, vol. 15, no. 1, pp. 215-227, 2019, doi: 10.1109/TSC.2019.2928303.

[111] M. Guerar, A. Merlo, M. Migliardi, F. Palmieri, and L. Verderame, "A fraud-resilient blockchain-based solution for invoice financing," *IEEE Transactions on Engineering Management*, vol. 67, no. 4, pp. 1086–1098, 2020, doi: 10.1109/TEM.2020.2971865.

[112] Z.-H. Sun, Z. Chen, S. Cao, and X. Ming, "Potential requirements and opportunities of blockchain-based industrial iot in supply chain: A survey," *IEEE Transactions on Computational Social Systems*, vol. 9, no. 5, pp. 1469-1483, 2021, doi: 10.1109/TCSS.2021.3129259.

[113] H. Li, R. Lu, L. Zhou, B. Yang, and X. Shen, "An efficient merkle-tree-based authentication scheme for smart grid," *IEEE Systems Journal*, vol. 8, no. 2, pp. 655–663, 2013, doi: 10.1109/JSYST.2013.2271537.

[114] M. Wazid, B. Bera, A. K. Das, S. P. Mohanty, and M. Jo, "Fortifying smart transportation security through public blockchain," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 16532-16545, 2022, doi: 10.1109/JIOT.2022.3150842.

[115] Y. Jiao, P. Wang, D. Niyato, and K. Suankaewmanee, "Auction mechanisms in cloud/-fog computing resource allocation for public blockchain networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 9, pp. 1975–1989, 2019, doi: 10.1109/TPDS.2019.2900238.

[116] M. J. Baucas, S. A. Gadsden, and P. Spachos, "Iot-based smart home device monitor using private blockchain technology and localization," *IEEE Networking Letters*, vol. 3, no. 2, pp. 52–55, 2021, doi: 10.1109/LNET.2021.3070270.

[117] S. Aggarwal and N. Kumar, "A consortium blockchain-based energy trading for demand response management in vehicle-to-grid," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 9480–9494, 2021, doi: 10.1109/TVT.2021.3100681.

[118] A. Boualouache, H. Sedjelmaci, and T. Engel, "Consortium blockchain for cooperative location privacy preservation in 5g-enabled vehicular fog computing," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 7, pp. 7087–7102, 2021, doi: 10.1109/TVT.2021.3083477.

[119] *Japan Car Sharing Association.* (accessed February 3, 2023). [Online]. Available: https://www.japan-csa.org/

[120] F. L. Da Silva, C. E. Nishida, D. M. Roijers, and A. H. R. Costa, "Coordination of electric vehicle charging through multiagent reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 11, no. 3, pp. 2347–2356, 2019, doi: 10.1109/TSG.2019.2952331.

[121] Y. Du and F. Li, "Intelligent multi-microgrid energy management based on deep neural network and model-free reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1066–1076, 2019, doi: 10.1109/TSG.2019.2930299.

[122] 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, 2020, doi: 10.1049/iet-stg.2019.0258.

[123] K. Vatanparvar, S. Faezi, I. Burago, M. Levorato, and M. A. Al Faruque, "Extended range electric vehicle with driving behavior estimation in energy management," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 2959–2968, 2018, doi: 10.1109/TSG.2018.2815689.

[124] B. Gao, L. Guo, Q. Zheng, B. Huang, and H. Chen, "Acceleration speed optimization of intelligent EVs in consideration of battery aging," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8009–8018, 2018, doi: 10.1109/TVT.2018.2840531.

[125] G. Ferro, M. Paolucci, and M. Robba, "Optimal charging and routing of electric vehicles with power constraints and time-of-use energy prices," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14 436–14 447, 2020, doi: 10.1109/TVT.2020.3038049.

[126] D. Baek, Y. Chen, A. Bocca, L. Bottaccioli, S. Di Cataldo, V. Gatteschi, D. J. Pagliari, E. Patti, G. Urgese, N. Chang *et al.*, "Battery-aware operation range estimation for terrestrial and aerial electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5471–5482, 2019, doi: 10.1109/TVT.2019.2910452.

[127] L. Zhao, W. Yao, Y. Wang, and J. Hu, "Machine learning-based method for remaining range prediction of electric vehicles," *IEEE Access*, vol. 8, pp. 212 423–212 441, 2020, doi: 10.1109/ACCESS.2020.3039815.

[128] C. Gomez-Quiles, G. Asencio-Cortes, A. Gastalver-Rubio, F. Martinez-Alvarez, A. Troncoso, J. Manresa, J. C. Riquelme, and J. M. Riquelme-Santos, "A novel ensemble method for electric vehicle power consumption forecasting: Application to the spanish system," *IEEE Access*, vol. 7, pp. 120 840–120 856, 2019, doi: 10.1109/ACCESS.2019.2936478.

[129] D. C. Nguyen, M. Ding, Q.-V. Pham, P. N. Pathirana, L. B. Le, A. Seneviratne, J. Li, D. Niyato, and H. V. Poor, "Federated learning meets blockchain in edge computing: Opportunities and challenges," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12 806–12 825, 2021, doi: 10.1109/JIOT.2021.3072611.

[130] J. Weng, J. Weng, J. Zhang, M. Li, Y. Zhang, and W. Luo, "Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive," *IEEE Transac-*

tions on Dependable and Secure Computing, vol. 18, no. 5, pp. 2438–2455, 2019, doi: 10.1109/TDSC.2019.2952332.

[131] Q. Wang, Y. Guo, X. Wang, T. Ji, L. Yu, and P. Li, "Ai at the edge: Blockchain-empowered secure multiparty learning with heterogeneous models," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9600–9610, 2020, doi: 10.1109/JIOT.2020.2987843.

[132] X. Bao, C. Su, Y. Xiong, W. Huang, and Y. Hu, "Flchain: A blockchain for auditable federated learning with trust and incentive," in 2019 5th International Conference on Big Data Computing and Communications (BIGCOM), Qingdao, China, 9-11 August 2019; pp. 151–159, doi: 10.1109/BIGCOM.2019.00030.

[133] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Blockchained on-device federated learning," *IEEE Communications Letters*, vol. 24, no. 6, pp. 1279–1283, 2019, doi: 10.1109/L-COMM.2019.2921755.

[134] Y. Qu, L. Gao, T. H. Luan, Y. Xiang, S. Yu, B. Li, and G. Zheng, "Decentralized privacy using blockchain-enabled federated learning in fog computing," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5171–5183, 2020, doi: 10.1109/JIOT.2020.2977383.

[135] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in ICC 2019-2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20-24 May 2019; pp. 1–7, doi: 10.1109/ICC.2019.8761315.

[136] S. AbdulRahman, H. Tout, A. Mourad, and C. Talhi, "FedMCCS: multicriteria client selection model for optimal IoT federated learning," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4723–4735, 2020, doi: 10.1109/JIOT.2020.3028742.

[137] Y. He, J. Ren, G. Yu, and J. Yuan, "Importance-aware data selection and resource allocation in federated edge learning system," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13 593–13 605, 2020, doi: 10.1109/TVT.2020.3015268.

[138] J. Xu and H. Wang, "Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective," *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 1188–1200, 2020, doi: 10.1109/TWC.2020.3031503.

[139] Y. J. Cho, J. Wang, and G. Joshi, "Client selection in federated learning: Convergence analysis and power-of-choice selection strategies," *arXiv preprint arXiv:2010.01243*, 2020.

[140] W. Zhang, X. Wang, P. Zhou, W. Wu, and X. Zhang, "Client selection for federated learning with non-IID data in mobile edge computing," *IEEE Access*, vol. 9, pp. 24 462–24 474, 2021, doi: 10.1109/ACCESS.2021.3056919.

[141] *History of CAN technology.* (accessed February 3, 2023). [Online]. Available: https://www.can-cia.org/can-knowledge/can/can-history/

[142] *Mercedes W140: First car with CAN.* (accessed February 3, 2023). [Online]. Available: https://can-newsletter.org/engineering/applications/160322_25th-anniversary-mercedes-w140-first-car-with-can/

[143] *ISO 11898:1993 Road vehicles - Interchange of digital information - Controller area network (CAN) for high-speed communication.* (accessed February 3, 2023). [Online]. Available: https://www.iso.org/standard/20380.html

[144] *Introduction to the Controller Area Network (CAN).* (accessed February 3, 2023). [Online]. Available: https://www.ti.com/lit/an/sloa101b/sloa101b.pdf

[145] *Google Maps. Cities of Japan.* (accessed February 3, 2023). [Online]. Available: https://www.google.com/maps

[146] *Japan Meteorological Agency.* (accessed February 3, 2023). [Online]. Available: http://www.data.jma.go.jp/gmd/risk/obsdl/index.php

[147] Z. Wang, M. Ogbodo, H. Huang, C. Qiu, M. Hisada, and A. B. Abdallah, "AEBIS: AI-enabled blockchain-based electric vehicle integration system for power management in

smart grid platform," *IEEE Access*, vol. 8, pp. 226 409–226 421, 2020, doi: 10.1109/ACCESS.2020.3044612.

[148] L. Burkhalter, H. Lycklama, A. Viand, N. Küchler, and A. Hithnawi, "Rofl: Attestable robustness for secure federated learning," *arXiv preprint arXiv:2107.03311*, 2021.

[149] L. Lyu, H. Yu, J. Zhao, and Q. Yang, "Threats to federated learning," in *Federated Learning*. Springer, 2020, pp. 3–16, doi: 10.1007/978-3-030-63076-8_1.

[150] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in International Conference on Machine Learning, PMLR, Long Beach, California, USA, 9-15 June 2019; pp. 634–643.

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

[152] *My license.* (accessed February 3, 2023). [Online]. Available: https://www.mylicense.co.jp/search/license/whatnormal2.php

[153] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[154] *REXEV.* (accessed February 3, 2023). [Online]. Available: https://rexev.co.jp/

[155] *Swarm.* (accessed February 3, 2023). [Online]. Available: https://www.ethswarm.org/

[156] *Bitcoin release 0.12.0*, 2016 (accessed February 3, 2023). [Online]. Available: https://bitcoin.org/en/release/v0.12.0

[157] *Go Ethereum.* (accessed February 3, 2023). [Online]. Available: https://geth.ethereum.org/

[158] M. Alharby and A. van Moorsel, "Blocksim: An extensible simulation tool for blockchain systems," *arXiv preprint arXiv:2004.13438*, 2020.

List of Publications

Refereed Journals

1. **Z. Wang**, M. Ogbodo, H. Huang, C. Qiu, M. Hisada and A. B. Abdallah, "AEBIS: AI-Enabled Blockchain-Based Electric Vehicle Integration System for Power Management in Smart Grid Platform," in IEEE Access, vol. 8, pp. 226409-226421, 2020, doi: 10.1109/ACCESS.2020.3044612.

2. **Z. Wang** and A. Ben Abdallah, "A Robust Multi-Stage Power Consumption Prediction Method in a Semi-Decentralized Network of Electric Vehicles," in IEEE Access, vol. 10, pp. 37082-37096, 2022, doi: 10.1109/ACCESS.2022.3163455.

3. Y. Liang, **Z. Wang** and A. B. Abdallah, "V2GNet: Robust Blockchain-Based Energy Trading Method and Implementation in Vehicle-to-Grid Network," in IEEE Access, vol. 10, pp. 131442-131455, 2022, doi: 10.1109/ACCESS.2022.3229432.

Refereed International Conferences

1. H. Huang, M. Ogbodo, **Z. Wang**, C. Qiu, M. Hisada and A. B. Abdallah, "Smart Energy Management System based on Reconfigurable AI Chip and Electrical Vehicles," 2021 IEEE International Conference on Big Data and Smart Computing (BigComp), 2021, pp. 233-238, doi: 10.1109/BigComp51126.2021.00051.

2. S. Phea, **Z. Wang**, J. Wang and A. B. Abdallah (2021), "Optimization and Implementation of a Collaborative Learning Algorithm for an AI-Enabled Real-time Biomedical System." In SHS Web of Conferences (Vol. 102, p. 04017).

Patents

1. Abderazek Ben Abdallah, **Wang Zhishang**, Masayuki Hisada, "An electricity trading system and an electricity trading method", 特願 2022-022472

2. Abderazek Ben Abdallah, **Wang Zhishang**, Khanh N. Dang, Masayuki Hisada, "EV Power Consumption Prediction Method and System for Power Management in Smart Grid", 特願 2022 (US)

A

Distance measures

This appendix provides the distance measures commonly used to calculate the distance between two vectors. We analyze whether the following distance measures are appropriate for calculating the diversity between two models.

Consider the calculation of diversity between two models M_i and M_j with corresponding parameter vectors P_i and P_j. The length of P_i and P_j is n. The parameters of P_i are denoted by p_i^1 to p_i^n. We start introducing the following distance measures.

- *Manhattan distance*
  The Manhattan distance, also refers to as L1 distance, is used to indicate the sum of the absolute axis distances at two points in a standard coordinate system. The Manhattan distance of P_1 and P_2 is calculated as:

$$D_{manhattan}(P_1, P_2) = \sum_{k=1}^{n} \left| p_i^k - p_j^k \right| \tag{A.1}$$

  One advantage of the Manttanan distance is its fast computational speed. Although it seems less intuitive compared to Euclidean distance, it is a useful distance measure for high-dimensional vectors.

- *Euclidean distance*
  The Euclidean distance, also refers to as L2 distance, is the true distance between two points in n-dimensional space. The Euclidean distance formula uses the Pythagorean theorem to calculate the distance from the Cartesian coordinates of these points. The Euclidean distance of P_1 and P_2 is calculated as follows:

$$D_{euclidean}(P_1, P_2) = \sqrt{\sum_{k=1}^{n} (p_i^k - p_j^k)^2} \tag{A.2}$$

Euclidean distance is not scale invariant, it works if the metric of each component of the vector is uniform. For calculating diversity, the Euclidean distance measure is considered appropriate because the magnitude of the parameters is the same. Nevertheless, a normalized Euclidean distance can be used as a more general measure:

$$D_{euclidean}^{normalized}(P_1, P_2) = \sqrt{(P_1 - P_2)^T S^{-1}(P_1 - P_2)} \tag{A.3}$$

where S is the covariance matrix.

- *Chebyshev distance*
  The Chebyshev distance between two vectors is defined as the maximum of the absolute value of the difference between the values of their coordinates. The Chebyshev distance of P_1 and P_2 is calculated as follows:

$$D_{chebyshev}(P_1, P_2) = max(\left|p_i^1 - p_j^1\right|, \left|p_i^2 - p_j^2\right|, ..., \left|p_i^k - p_j^k\right|) \tag{A.4}$$

  The Chebyshev distance is often used for special applications, which complicates its use as a general distance measure like the Manhattan or Euclidean distance. In this work, the Chebyshev distance is inappropriate for calculating model diversity because the parameter vector is high dimensional and diversity is unlikely to depend on a single parameter.

- *Cosine distance*
  Cosine similarity is the calculation of the cosine of the angle between two vectors. The cosine distance is the cosine similarity obtained by subtracting this value from 1. The cosine distance of P_1 and P_2 is calculated as follows:

$$D_{cosine}(P_1, P_2) = 1 - \frac{P_1 \cdot P_2}{\|P_1\| \cdot \|P_1\|} \tag{A.5}$$

  The cosine distance focuses more on the differences between the dimensions and less on the numerical differences. While cosine similarity does not account for the difference in rating scale between different vectors, it is inappropriate for two vectors with similar directions but very different magnitudes.

- *Minkowski distance*
  Minkowski distances are a set of definitions of distances, which are generalized expressions of multiple distance metric formulas. The Minkowski distance of P_1 and P_2 is calculated as:

$$D_{minkowski}(P_1, P_2) = \left(\sum_{i=1}^{n}(x_i - y_i)^r\right)^{1/r} \tag{A.6}$$

  If $r = 1$, the Minkowski distance is converted to a Manhattan distance; if $r = 2$, the Minkowski distance is converted to a Euclidean distance; if $r = \infty$, the Minkowski distance is converted to a Chebyshev distance.

B

Example of the FL-QLMS algorithm using Manhattan distance

This appendix demonstrates examples of the FL-QLMS algorithm using Manhattan distance with and without auxiliary models. In the example, we consider five local models M_1 to M_5 with a size of 157 (11-8-6-1). A local model is to be selected. An auxiliary model M_{aux} with the same size is provided. The detailed parameter vectors of the models are shown in Table B.1, B.2 and B.3. We begin by demonstrating how FL-QLMS is performed with an auxiliary model to select a qualified model from M_1 to M_5. We calculate the model diversity as follows:

- $DI(M_1, M_{aux}) = \sum_{k=1}^{157} \left| p_1^k - p_{aux}^k \right| = 15.14$
- $DI(M_2, M_{aux}) = \sum_{k=1}^{157} \left| p_2^k - p_{aux}^k \right| = 15.23$
- $DI(M_3, M_{aux}) = \sum_{k=1}^{157} \left| p_3^k - p_{aux}^k \right| = 15.62$
- $DI(M_4, M_{aux}) = \sum_{k=1}^{157} \left| p_4^k - p_{aux}^k \right| = 16.23$
- $DI(M_5, M_{aux}) = \sum_{k=1}^{157} \left| p_5^k - p_{aux}^k \right| = 15.44$

Therefore, M_1 with lowest diversity of 15.14 is selected as the qualified model in this case.

Then we show how FL-QLMS without the auxiliary model is performed. We calculate the model diversity for each local model as follows:

- For M_1:
  $DI(M_1, M_2) = \sum_{k=1}^{157} \left| p_1^k - p_2^k \right| = 20.87$
  $DI(M_1, M_3) = \sum_{k=1}^{157} \left| p_1^k - p_3^k \right| = 20.10$
  $DI(M_1, M_4) = \sum_{k=1}^{157} \left| p_1^k - p_4^k \right| = 20.77$
  $DI(M_1, M_5) = \sum_{k=1}^{157} \left| p_1^k - p_5^k \right| = 20.51$

- For M_2:
  $DI(M_2, M_1) = DI(M_1, M_2) = 20.87$
  $DI(M_2, M_3) = \sum_{k=1}^{157} \left| p_2^k - p_3^k \right| = 21.15$
  $DI(M_2, M_4) = \sum_{k=1}^{157} \left| p_2^k - p_4^k \right| = 20.06$
  $DI(M_2, M_5) = \sum_{k=1}^{157} \left| p_2^k - p_5^k \right| = 21.07$

- For M_3:
  $DI(M_3, M_1) = DI(M_1, M_3) = 20.10$
  $DI(M_3, M_2) = DI(M_2, M_3) = 21.15$
  $DI(M_3, M_4) = \sum_{k=1}^{157} \left| p_3^k - p_4^k \right| = 22.23$
  $DI(M_3, M_5) = \sum_{k=1}^{157} \left| p_3^k - p_5^k \right| = 20.70$

- For M_4:
  $DI(M_4, M_1) = DI(M_1, M_4) = 20.77$
  $DI(M_4, M_2) = DI(M_2, M_4) = 20.06$
  $DI(M_4, M_3) = DI(M_3, M_4) = 22.23$
  $DI(M_4, M_5) = \sum_{k=1}^{157} \left| p_4^k - p_5^k \right| = 21.69$

- For M_5:
  $DI(M_5, M_1) = DI(M_1, M_5) = 20.52$
  $DI(M_5, M_2) = DI(M_2, M_5) = 21.07$
  $DI(M_5, M_3) = DI(M_3, M_5) = 20.70$
  $DI(M_5, M_4) = DI(M_4, M_5) = 21.69$

We obtain the average diversity for each model subsequently as follows:

- $\bar{DI}_1 = \frac{1}{4}(DI(M_1, M_2) + DI(M_1, M_3) + DI(M_1, M_4) + DI(M_1, M_5)) = 20.565$
- $\bar{DI}_2 = \frac{1}{4}(DI(M_2, M_1) + DI(M_2, M_3) + DI(M_2, M_4) + DI(M_2, M_5)) = 20.788$
- $\bar{DI}_3 = \frac{1}{4}(DI(M_3, M_1) + DI(M_3, M_2) + DI(M_3, M_4) + DI(M_3, M_5)) = 21.045$
- $\bar{DI}_4 = \frac{1}{4}(DI(M_4, M_1) + DI(M_4, M_2) + DI(M_4, M_3) + DI(M_4, M_5)) = 21.188$
- $\bar{DI}_5 = \frac{1}{4}(DI(M_5, M_1) + DI(M_5, M_2) + DI(M_5, M_3) + DI(M_5, M_4)) = 20.995$

Therefore, M_1 with lowest average diversity of 20.565 is selected as the qualified model in this case.

Table B.1: **The parameter vector of models M_1 and M_2.**

Local model M_1																		
Weights of Layer 1																		
0.72	-0.79	0.55	-1	0.96	0.86	0.69	-0.19	-0.76	0.15	0.84	-0.51	-0.41	0.0	-0.43	-0.82	0.95	0.14	0.26	0.11
-0.49	-0.77	0.3	-0.48	0.59	0.3	0.79	0.72	-0.51	0.07	0.6	0.01	0.74	-0.42	-0.53	-0.69	-0.43	0.87	0.02	0.91
-0.5	0.13	0.21	-0.29	-0.76	-0.4	0.58	1	-0.35	-0.13	0.14	-0.24	0.45	-0.3	-0.06	0.84	0.15	0.42	-0.36	0.57
-0.86	0.07	-0.94	0.21	-0.68	-0.49	0.94	0.43	1	-0.22	-0.62	-0.51	-0.1	-0.6	0.3	-0.21	-0.87	0.7	0.61	1
-0.56	-0.75	-0.68	-0.87	0.41	0.27	-0.48	0.31												
Biases of Layer 1																		
0.58	0.75	0.69	0.91	0.04	0.73	0.82	-0.83											
Weights of Layer 2																		
-1.0	-0.67	-0.14	-0.75	0.75	-0.04	0.44	-1	0.55	-0.93	0.2	1	0.06	-0.3	0.85	0.02	0.22	0.44	-0.94	0.73
0.37	0.51	-0.51	0.87	-0.02	-0.33	-0.55	0.53	-0.27	-0.09	-0.44	0.35	-0.66	-0.19	-0.56	-0.85	0.73	0.35	-0.29	-0.55
-0.68	-0.69	0.14	-0.49	0.13	-0.63	-0.76	-0.35												
Biases of Layer 2							Weights of Layer 3							Biases of Layer 3					
0.68	0.51	-0.05	-0.65	0.6	0.93		0.21	0.74	-0.26	-0.36	-0.02	-0.65		0.51					
Local model M_2																		
Weights of Layer 1																		
0.9	-0.85	0.83	-0.89	1	0.79	0.82	-0.31	-0.89	0.08	0.63	-0.27	-0.66	-0.3	-0.67	-0.84	0.89	0.2	0.0	0.23
-0.52	-0.85	0.17	-0.37	0.7	0.14	0.76	0.57	-0.66	-0.19	0.48	0.11	0.8	-0.46	-0.55	-0.5	-0.52	0.74	0.08	1
-0.39	0.18	0.11	-0.5	-1	-0.6	0.61	0.64	-0.61	-0.46	0.09	-0.16	0.2	-0.3	-0.09	0.57	0.21	0.41	-0.48	0.29
-0.59	0.41	-0.62	0.03	-0.51	-0.73	1	0.51	0.99	-0.34	-0.55	-0.52	-0.02	-0.78	-0.03	-0.26	-0.65	0.66	0.43	1
-0.61	-0.56	-0.73	-0.63	0.2	0.4	-0.29	0.15												
Biases of Layer 1																		
0.51	0.56	0.98	1	-0.02	0.76	0.86	-0.69											
Weights of Layer 2																		
-0.85	-0.74	-0.03	-0.83	0.9	0.1	0.56	-1	0.79	-0.79	0.31	0.75	-0.07	-0.28	0.73	0.21	0.25	0.5	-0.81	0.5
0.43	0.67	-0.57	0.99	-0.25	-0.51	-0.6	0.71	-0.21	-0.08	-0.54	0.31	-0.94	-0.4	-0.57	-0.93	0.91	0.29	-0.25	-0.26
-0.86	-0.61	0.01	-0.63	0.45	-0.68	-0.74	-0.37												
Biases of Layer 2							Weights of Layer 3							Biases of Layer 3					
0.49	0.53	0.02	-0.83	0.69	0.65		-0.05	0.96	-0.14	-0.1	0.05	-0.56		0.54					

Table B.2: **The parameter vector of models M_3 and M_4.**

Local model M_3																		
Weights of Layer 1																		
0.66	-0.97	0.64	-1	0.86	1.0	0.92	-0.3	-0.86	0.25	0.94	-0.49	-0.7	-0.2	-0.44	-0.95	0.93	0.09	0.05	0.32
-0.46	-0.92	0.17	-0.65	0.41	0.33	0.84	0.86	-0.56	-0.13	0.52	0.06	0.98	-0.43	-0.53	-0.47	-0.41	0.87	0.22	0.84
-0.47	-0.07	0.21	-0.59	-0.76	-0.24	0.56	0.64	-0.54	-0.22	0.2	-0.22	0.35	-0.22	0.17	0.68	0.19	0.55	-0.19	0.55
-0.57	0.42	-0.81	0.19	-0.45	-0.62	0.76	0.37	0.76	-0.22	-0.61	-0.48	-0.11	-0.59	0.28	-0.28	-0.65	0.95	0.46	0.81
-0.65	-0.51	-0.63	-0.71	0.12	0.33	-0.57	0.46												
Biases of Layer 1																		
0.57	0.88	0.95	0.95	-0.06	0.58	0.77	-0.69											
Weights of Layer 2																		
-0.96	-0.51	0.16	-0.74	0.67	-0.04	0.41	-0.85	0.6	-0.88	0.13	0.87	0.18	-0.23	0.93	0.37	0.4	0.31	-0.62	0.82
0.11	0.65	-0.53	0.74	-0.23	-0.37	-0.44	0.68	-0.42	-0.31	-0.21	0.05	-0.63	-0.38	-0.43	-0.8	0.86	0.19	-0.14	-0.25
-0.95	-0.47	0.29	-0.64	0.34	-0.9	-0.94	-0.53												
Biases of Layer 2							Weights of Layer 3							Biases of Layer 3					
0.58	0.63	0.07	-0.92	0.52	0.8		-0.17	0.73	-0.19	-0.26	0.05	-0.83		0.51					
Local model M_4																		
Weights of Layer 1																		
0.97	-0.6	0.59	-0.81	0.94	0.91	0.66	-0.34	-1	0.22	0.63	-0.15	-0.66	-0.32	-0.45	-0.64	1	-0.0	-0.02	0.15
-0.82	-1	0.28	-0.42	0.48	0.25	0.96	0.78	-0.49	-0.12	0.56	0.03	0.79	-0.35	-0.44	-0.61	-0.44	0.89	0.01	1
-0.26	0.13	0.13	-0.32	-1	-0.26	0.36	0.91	-0.44	-0.22	0.08	-0.55	0.3	-0.44	0.01	0.61	0.49	0.65	-0.47	0.34
-0.63	0.17	-0.91	0.28	-0.65	-0.6	0.82	0.59	0.8	-0.4	-0.38	-0.8	-0.36	-0.78	0.01	-0.39	-0.55	0.83	0.56	1
-0.57	-0.73	-0.79	-0.85	0.25	0.31	-0.58	0.11												
Biases of Layer 1																		
0.41	0.7	0.62	0.91	0.13	0.86	0.66	-0.75											
Weights of Layer 2																		
-0.7	-0.55	-0.17	-1	0.65	-0.04	0.62	-1	0.76	-1	0.34	0.78	0.13	-0.35	0.64	0.32	0.28	0.39	-0.86	0.65
0.44	0.65	-0.77	0.99	-0.1	-0.41	-0.74	0.46	-0.29	-0.15	-0.4	0.13	-0.63	-0.23	-0.25	-0.81	0.89	0.13	-0.19	-0.26
-0.8	-0.56	0.24	-0.39	0.36	-0.96	-0.69	-0.67												
Biases of Layer 2							Weights of Layer 3							Biases of Layer 3					
0.43	0.5	0.3	-0.78	0.55	0.65		-0.11	0.73	-0.08	-0.22	-0.09	-0.63		0.48					

Table B.3: **The parameter vector of models M_5 and M_{aux}.**

Local model M_5																		
Weights of Layer 1																		
0.84	-0.92	0.68	-1	0.96	1	0.96	-0.24	-0.97	0.09	0.72	-0.26	-0.47	-0.02	-0.49	-0.71	0.81	-0.09	0.17	0.28
-0.53	-0.74	0.16	-0.69	0.36	0.39	1	0.74	-0.57	0.01	0.62	-0.08	0.72	-0.43	-0.36	-0.74	-0.61	0.85	0.3	0.96
-0.44	0.17	0.29	-0.67	-0.94	-0.45	0.52	0.66	-0.66	-0.22	0.3	-0.5	0.54	-0.28	-0.04	0.6	0.43	0.34	-0.18	0.27
-0.54	0.11	-0.78	0.04	-0.4	-0.45	0.98	0.52	0.89	-0.11	-0.37	-0.77	-0.27	-0.52	0.25	-0.17	-0.87	0.99	0.71	0.79
-0.73	-0.76	-0.72	-0.69	0.4	0.37	-0.52	0.22												
Biases of Layer 1																		
0.51	0.56	0.87	0.98	-0.08	0.66	0.94	-0.57											
Weights of Layer 2																		
-0.89	-0.59	-0.19	-1	0.81	0.08	0.42	-1	0.62	-1	0.49	1	0.18	-0.2	0.9	0.21	0.51	0.6	-0.96	0.45
0.07	0.61	-0.58	0.67	-0.17	-0.53	-0.77	0.37	-0.26	-0.08	-0.49	0.34	-0.96	-0.31	-0.49	-1	0.62	0.09	-0.41	-0.25
-0.87	-0.47	0.28	-0.47	0.29	-0.99	-0.79	-0.45												
Biases of Layer 2							Weights of Layer 3							Biases of Layer 3					
0.54	0.45	0.31	-0.73	0.49	0.73		-0.09	1	-0.34	-0.36	0.24	-0.71		0.57					
Auxiliary model M_{aux}																		
Weights of Layer 1																		
0.83	-0.78	0.73	-0.94	0.97	0.92	0.82	-0.2	-0.88	0.06	0.8	-0.31	-0.53	-0.13	-0.59	-0.81	0.87	0.01	0.08	0.27
-0.64	-0.86	0.36	-0.53	0.53	0.29	0.94	0.74	-0.46	-0.1	0.53	-0.07	0.89	-0.43	-0.48	-0.59	-0.55	0.9	0.17	0.9
-0.33	0.06	0.24	-0.48	-0.9	-0.42	0.45	0.83	-0.54	-0.29	0.2	-0.36	0.39	-0.33	0.05	0.66	0.35	0.48	-0.29	0.37
-0.7	0.27	-0.79	0.1	-0.48	-0.55	0.95	0.46	0.94	-0.23	-0.43	-0.61	-0.17	-0.7	0.12	-0.2	-0.71	0.81	0.57	0.91
-0.58	-0.58	-0.79	-0.72	0.31	0.23	-0.49	0.3												
Biases of Layer 1																		
0.57	0.74	0.81	0.97	0.05	0.69	0.79	-0.68											
Weights of Layer 2																		
-0.85	-0.55	0.01	-0.91	0.7	0.13	0.54	-0.97	0.61	-0.98	0.31	0.95	0.09	-0.39	0.8	0.18	0.34	0.48	-0.82	0.64
0.26	0.6	-0.6	0.81	-0.2	-0.45	-0.59	0.56	-0.27	-0.22	-0.35	0.2	-0.83	-0.37	-0.43	-0.85	0.77	0.25	-0.3	-0.42
-0.84	-0.62	0.12	-0.59	0.33	-0.81	-0.78	-0.54												
Biases of Layer 2							Weights of Layer 3							Biases of Layer 3					
0.56	0.56	0.12	-0.85	0.62	0.84		0.02	0.91	-0.18	-0.2	0.07	-0.71		0.47					