Computer Modeling in Engineering & Sciences

Tech Science Press

https://doi.org/10.32604/cmes.2026.081849

ARTICLE

QFedFormer: A Privacy-Preserving Federated Transformer with Blockchain-Anchored Incentives for Dynamic EV Charging Pricing

Lilia Tightiz[1], **L. Minh Dang**[2,3] **and Hyosik Yang**[1,*]

[1]Department of Computer Science and Engineering, Sejong University, 209, Neungdong-ro, Gwangjin-gu, Seoul, Republic of Korea
[2]The Institute of Research and Development, Duy Tan University, Da Nang, Vietnam
[3]Faculty of Information Technology, Duy Tan University, Da Nang, Vietnam

*Corresponding Author: Hyosik Yang. Email: hsyang@sejong.ac.kr

Received: 10 March 2026; Accepted: 14 May 2026; Published: 30 June 2026

ABSTRACT: We present QFedFormer, a federated transformer for dynamic electric vehicle (EV)-charging price prediction that combines quantization-aware training, SHAP-guided explainability, and blockchain-based incentives. The framework trains across distributed charging stations without centralizing user data, and programmable contracts set tariffs from forecasted demand and user-declared flexibility, while token rewards are derived from SHAP-based utility scores and anchored on-chain via Merkle proofs. On a real-world dataset, QFedFormer attains an energy-demand RMSE of 1.82 ± 0.02 kWh and a tariff RMSE of 11.83 ± 0.10 KRW/kWh (MAPE 2.7 ± 0.2%) in the non-private baseline, outperforming FedAvg and Block-FeDL by 14.1% and 9.5%, respectively. Under client-level differential privacy (DP) with ($\sigma = 1.6$, $C = 1$, $p = 0.1$, $\delta_{DP} = 10^{-5}$), QFedFormer achieves ($\varepsilon = 2.0$, $\delta_{DP} = 10^{-5}$) after 50 rounds under a Rényi accountant, with forecast accuracy degrading modestly to 1.95 kWh RMSE (~7.1% relative increase vs. non-private baseline). Blockchain evaluation shows an average audit latency of 58 ms per audit round, while a permissioned Ethereum-compatible deployment sustains more than 500 client updates per minute with gas costs of ~$0.039/client per audit round. These results indicate that QFedFormer enables accurate, privacy-preserving, and auditable coordination of EV–grid interactions, offering both regulators and service providers a practical deployment pathway.

KEYWORDS: Blockchain; federated learning; EV charging; dynamic pricing; differential privacy; SHAP; tokenized incentives

1 Introduction

As EV adoption increases, the need for intelligent and secure EV charging management is increased [1]. Conventional centralized energy management methods are limited by the distributed nature of EV infrastructure, heterogeneous user behavior, and the variability of renewable energy inputs [2]. Key challenges are identified as real-time scalability, data privacy preservation, and demand-side participation without excessive grid burden or loss of user incentives [3]. These challenges are further intensified by the scale and variability of EV charging demand. Additional constraints are introduced for grid coordination and infrastructure efficiency [4].

Further challenges are observed in EV charging pricing compared with other energy applications. Pricing decisions are affected by highly dynamic consumer demand, user behavior, and grid constraints. This condition produces unstable and non-IID data distributions across charging stations.

Privacy concerns arise from the continuous collection of consumer-level charging data, such as user location, charging time, and consumed energy. A potential risk of private data leakage is introduced.

Additional risks are associated with the security of EV charging operations. Communication between charging stations and grid operators is exposed to cyberattacks, such as data injection, spoofing, and MIM attacks. User consumption records and charging price decisions may be altered by attackers. These challenges indicate that an adequate pricing mechanism is required.

To address these issues, decentralized learning and secure communication channels are required. Collaborative model training is enabled by federated leadrning (FL) without direct exchange of raw data, and user anonymity is preserved. Privacy is ensured by DP for transmitted model updates. Price calculation and payment execution are secured by blockchain technology. Pricing rules and reward distribution are executed automatically through smart contracts. Transparent decisions are also supported by explainability tools such as SHAP. Decisions can be verified, and user trust in the pricing policy is improved. Therefore, an EV charging pricing algorithm can be designed with these components to provide security and privacy.

Complementary capabilities are provided by FL and blockchain paradigms in decentralized EV charging systems. Collaborative model training is achieved by FL without data disclosure, and privacy protection is supported in distributed data settings [5]. Coordination integrity, and enforceable execution are guaranteed by blockchain through smart contracts [6].

However, demand prediction and pricing execution are usually considered separately in current methods. Explainability, auditability, and incentive alignment are also not considered jointly in many existing frameworks.

Integration of dynamic pricing with the forecasting process poses several related challenges. First, a challenge arises from the non-IID nature of EV load statistics. Charging demand is influenced by time slots, user routines, regional needs, and charging-point availability. Therefore, existing centralized methods are not considered effective [7]. Second, price changes and rational user expectations are required to be balanced. Real-time pricing may not be accepted by users when fair treatment is not guaranteed. This issue may negatively affect grid balance [8].

Other related problems are also identified. First, infrastructure heterogeneity is required to be considered. Hardware equipment, local policies, renewable integration, and other factors differ across regions. This heterogeneity increases the difficulty of applying a single forecasting or pricing model [9]. In addition, a decentralized solution is required, as discussed in [10]. Moreover, incentive design in FL is underexplored [11]. Most federated methods provide only simple participation-based rewards. Quantifiable utility measures and reward allocation models are not provided.

To address data fragmentation, explainability, and misaligned incentives, SHAP-based model explainability is combined with blockchain-based coordination and incentive schemes. In this study, SHAP is used as a post-hoc model interpretation and compression method. It is not used as a predictive method. In current studies, FL or dynamic pricing is addressed separately. However, a unified approach that considers data fragmentation, model explainability, and incentive design is limited.

Formally, the main problem addressed in this study is the need for a deployable EV charging pricing mechanism. Three goals are required to be satisfied at the same time: (i) accurate forecasting of short-term EV loads, (ii) provable privacy guarantees, and (iii) auditable and explainable tariffs with enforceable incentives. Most existing approaches address only two of these goals. All three goals are not addressed together. Some methods are centralized or privacy-invasive. Explainability for regulators is not provided in some methods. User behavior toward grid-level objectives is not guided by other methods. QFedFormer is designed to

address this gap. FL, SHAP-based explanation techniques, and blockchain technology are combined in the federated setting.

To this end, QFedFormer is proposed as a blockchain-enabled FL approach for EV charging environments. The main contributions are summarized as follows:

- SHAP-Pruned Federated Transformer: A lightweight Transformer-based forecasting model is developed for non-IID and distributed EV charging networks. SHAP-guided attention pruning is applied to improve post-hoc interpretability and edge efficiency.
- Privacy-Preserving Optimization: DP and post-pruning quantization are integrated to protect local model updates while practical utility is preserved.
- Smart Contract Pricing: A blockchain-based pricing engine is designed. Bounded real-time tariff updates are executed based on forecasted demand and user-declared flexibility.
- Utility-Linked Token Incentives: A programmable on-chain reward mechanism is developed. Tokens are issued according to a composite utility score.
- On-Chain Explainability and Auditability: Cryptographic commitments of SHAP-based explanation summaries are anchored on-chain through Merkle roots.

The rest of this paper is organized as follows. Related works in the area of FL, blockchain coordination, and EV charging are discussed in Section 2. The design of our proposed QFedFormer framework is explained in Section 3. These include local prediction, pruning through SHAP values, quantization, federated averaging, and smart contract enforcement. The experimental setting, data set details, and performance metrics are elaborated in Section 4. Experimental results are reported in Section 5. This section includes analysis based on accuracy, privacy, scalability, and incentive compliance behavior. The conclusion of this paper is drawn in Section 6.

2 Related Work

2.1 FL for EV and Smart Grid Forecasting

The FL technology makes it possible to train models using EV charger, smart meter, and grid-edge device data without data transfer between users. It is appropriate for EV and smart grid forecasting tasks due to distributed, privacy-preserving, and non-IID nature of the data sets used.

It is claimed that current achievements in deep learning-based time series forecasting include excellent performance in capturing long-range dependencies in sequences with Transformer models. FedFormer and similar frequency Transformers reduce computational load while ensuring the precision of forecasting. These types of models can be used for energy demand prediction in a distributed manner.

A variety of traditional ML algorithms are explored to predict EV charging in smart grids. Algorithms like DNN, SVM, RF, and LSTM are applied for load forecasting and charging optimization. Demand estimates become more precise, and the grid becomes less vulnerable to power imbalance as a result. However, no focus is paid to privacy protection and decentralization [12]. A rising trend is noted in the application of FL and blockchain in the context of EV charging stations and smart grid systems. The focus areas include privacy, scalability, and distributed intelligence. The FL scheme incorporating blockchain technology has been suggested as Block-FeDL [13] by Danish et al. for load forecasting of EV charging stations. On the same platform of developing secure and decentralized forecasting models, Hameed et al. [14] proposed another model called Block-FeST, which employed sparse transformer models within the federated model. This work still lacks clarity in incentive alignment and model decision-making.

Other FL methods, including FedPT-V2G, an attention architecture proposed by Shang and Li [9], prove the efficacy of attention architectures in facilitating two-way learning in EV and grid realms. Nonetheless,

these algorithms fail to provide techniques to ensure reward traceability, accountability, and feasible pricing implementation. You et al. [15] further extended FL into renewable energy resource contexts using FMGCN, a federated multi-graph convolutional network, to perform wind forecasting on non-IID nodes. Nonetheless, this framework is not developed to address EV demand-side flexibility and pricing-related decision-making. By incorporating Laplace noise into pricing decisions, Hassan et al. [16] proposed a DP demand-side management scheme in the field of data privacy. Although privacy is improved, the method lacks auditability, interpretability, and tokenized incentives.

2.2 Incentive Mechanisms and Learning-Based Pricing

Dynamic pricing in EV charging systems is used to adjust tariffs according to demand, grid constraints, user flexibility, and local operating conditions. However, many pricing methods remain centralized, require global data access, or lack transparent execution, which creates privacy and trust concerns.

Blockchain-based incentive mechanisms have also been explored to ensure transparent and verifiable reward allocation. Pricing regulations and incentivization enforcement are realized through smart contracts. Blockchain is used to ensure auditability and tamper-resistance of the incentive scheme. Incentives are essential for sustained participation in FL systems. Nevertheless, adequate research has not been conducted in the existing studies.

Incentives derived from dynamic pricing in FL have been explored by Ding et al. [17], considering the local cost sensitivity analysis. Nonetheless, the use of decentralized implementation and blockchain support has been excluded. Reward schemes for EV aggregators and user involvement have been designed using Stackelberg games by Chen et al. [18]. However, no provision for smart contract formulation has been considered. Other related works include dynamic pricing frameworks modeled by Zhang et al. [19] using DRL, accounting for the bounded rationality of consumers. Nevertheless, federated mechanisms have not been employed, along with blockchain-based incentive execution.

Additionally, dynamic pricing has been discussed in the literature concerning centralized ML and optimization algorithms [20]. Deep learning on graphs has been employed by Ruan et al. [21] to derive prices for retail electricity consumption under localized demands. Multi-objective optimization methods are proposed by Zhang et al. [22]. Consumer flexibility and price sensitivity are considered. These methods are applied in centralized settings, but privacy-preserving measures are not considered. Similarly, pricing mechanisms are developed by Tang et al. [23] using DRL models, with consideration of grid-based constraints. Regulatory compliance and realistic consumer behavior are examined by Wang et al. [24]. However, validation is limited to centralized settings, or federated predictions with contract-based incentive implementation are not incorporated.

2.3 Blockchain-Based Coordination and Auditability

Security threats in smart grid communication layers include spoofing, data injection, MITM, replay, and availability attacks. Confidentiality, integrity, and availability may be affected by these threats. Therefore, blockchain-based coordination is relevant for transaction records and for verifiable and tamper-resistant EV charging operations. This layer-wise threat characterization is aligned with recent smart grid security analyses. Cyberattacks are categorized across system and communication layers. Impacts on confidentiality, integrity, and availability are defined, and mitigation strategies are provided [25].

Solutions based on blockchain technology are proposed to support decentralized coordination in energy pricing. A blockchain-enabled zero-sum game-based pricing scheme is proposed by Kakkar et al. [26] for EV charging stations. Smart contracts and IPFS-based transaction storage are incorporated. Price fairness

is improved, and coordination latency is reduced. However, federated demand forecasting and model-level explainability are not included.

A stochastic blockchain-based energy management model for smart cities is proposed by Zhang et al. [27]. V2G and V2S interactions under uncertainty are integrated. Stochastic behavior is captured by an unscented transformation method. Data exchange among energy subsystems is secured by blockchain. Effective coordination between transportation and grid components is demonstrated. However, centralized optimization is used, and federated learning or decentralized model training is not included. In addition, incentive mechanisms and audit-level interpretability of model decisions are not addressed. Applicability in privacy-sensitive and user-participatory environments is limited.

A blockchain-based dynamic energy pricing approach is introduced by Abadi et al. [28]. Machine-learning-based demand prediction is used to support supply-chain resilience. Although their model demonstrates the benefits of blockchain-assisted pricing execution, it operates in a centralized learning setting and does not address privacy-preserving knowledge acquisition, decentralized model coordination, or incentive traceability. To provide a structured view of these security challenges, Table 1 summarizes layer-wise attack types, their impact on confidentiality, integrity, and availability, and the corresponding mitigation mechanisms adopted in this work.

Table 1: Layer-wise attacks and mitigation strategies in FL and blockchain-enabled EV charging systems.

Layer	Attack Type	Impact	Mitigation in This Work
Client Layer	Data poisoning, model manipulation	Incorrect model updates and biased pricing	DP noise addition and SHAP-based pruning
Communication Layer	MITM, spoofing, injection attacks	Data corruption and loss of confidentiality	zk-SNARK verification and secure transmission
Aggregation Layer	Byzantine updates, gradient attacks	Degraded global model accuracy	Weighted aggregation and verification constraints
Blockchain Layer	Transaction tampering, replay attacks	Loss of auditability and trust	Smart contract execution and Merkle commitment
Application Layer	Pricing manipulation, reward exploitation	Unfair pricing and incentive imbalance	Bounded tariff rules and token-based incentives

The table shows that mitigation strategies are aligned with each layer to reduce attack impact and preserve reliable pricing and coordination.

2.4 Summary and Research Gaps

However, in this area of work, there are still several critical areas that are left untouched. First, in most existing work, demand forecasting and tariffs are decoupled in such a way that real-time tariffs are not properly and consistently enforced in real time. Second, without utility-aware scoring or traceable reward distribution, incentive mechanisms are usually static and rely on coarse participation-based rewards. Third, explainability is rarely integrated as a system-level component, and on-chain anchoring of model-derived decision evidence is still mostly unexplored, despite its growing recognition as crucial for user

trust and regulatory oversight. This proposal integrates a privacy-preserving FL solution for decentralized data ownership, a utility-related token economic incentive for participant coordination, a smart contract-based pricing execution service for the enforcement of electricity costs in real-time, a SHAP-based model explanation and approximation service for the interpretability of predictions, and a blockchain-based auditability service for the verification of data integrity. When combined, these elements provide a unified and implementable solution for decentralized EV charging networks.

Table 2 summarizes the identified research gaps and maps them to the corresponding architectural and algorithmic design choices used in this work. The comparison shows that prior approaches address only isolated aspects of decentralized EV charging, such as forecasting, pricing, or blockchain coordination. In contrast, joint integration of FL, explainability, privacy preservation, incentive alignment, and executable smart contracts is achieved in a single deployable design.

Table 2: Technology–challenge justification in decentralized EV charging systems.

System Challenge	Representative Prior Work and Limitations	Design Choice in This Work
Decentralized demand forecasting under non-IID data	Blockchain-assisted FL forecasting without incentive alignment or interpretability (Block-FeDL (2025) [13], Block-FeST (2025) [14], FedPT-V2G (2024) [9])	Federated Transformer forecasting with client-adaptive learning (QFedFormer)
User privacy and regulatory compliance	Centralized or semi-centralized learning and pricing requiring access to raw user data (2024) [20], (2023) [21], (2022) [22]	Client-level DP integrated into federated training
Model interpretability and decision transparency	Black-box forecasting or pricing models without explanation or traceability (2022) [16]	Post-hoc SHAP explanations with structured pruning and on-chain anchoring
Incentive alignment and fair participation	Static or game-theoretic incentives without executable enforcement (2025) [17], (2024) [18], (2024) [19]	Utility-linked token incentives enforced via smart contracts
Executable and auditable pricing mechanisms	Centralized or conceptual pricing schemes lacking real-time enforceability (2024) [20], (2022) [22], (2022) [24]	Smart contract–based bounded tariff execution
Regulator-facing auditability and accountability	Blockchain-based dynamic pricing systems with transaction-level transparency but without learning provenance or explanation traceability (2022) [26], (2024) [28], (2023) [27]	Merkle-root commitments of SHAP explanations for post-hoc verification

3 System Architecture and Methodology

To enable secure, explainable, and scalable dynamic pricing in EV charging networks, we design a five-layer architecture (Fig. 1) centered on QFedFormer, a quantized and SHAP-pruned federated transformer for

edge-level forecasting. The layers cover (i) data acquisition, (ii) local forecasting, (iii) federated aggregation with privacy, (iv) smart contract pricing, and (v) tokenized incentives with auditable explainability.

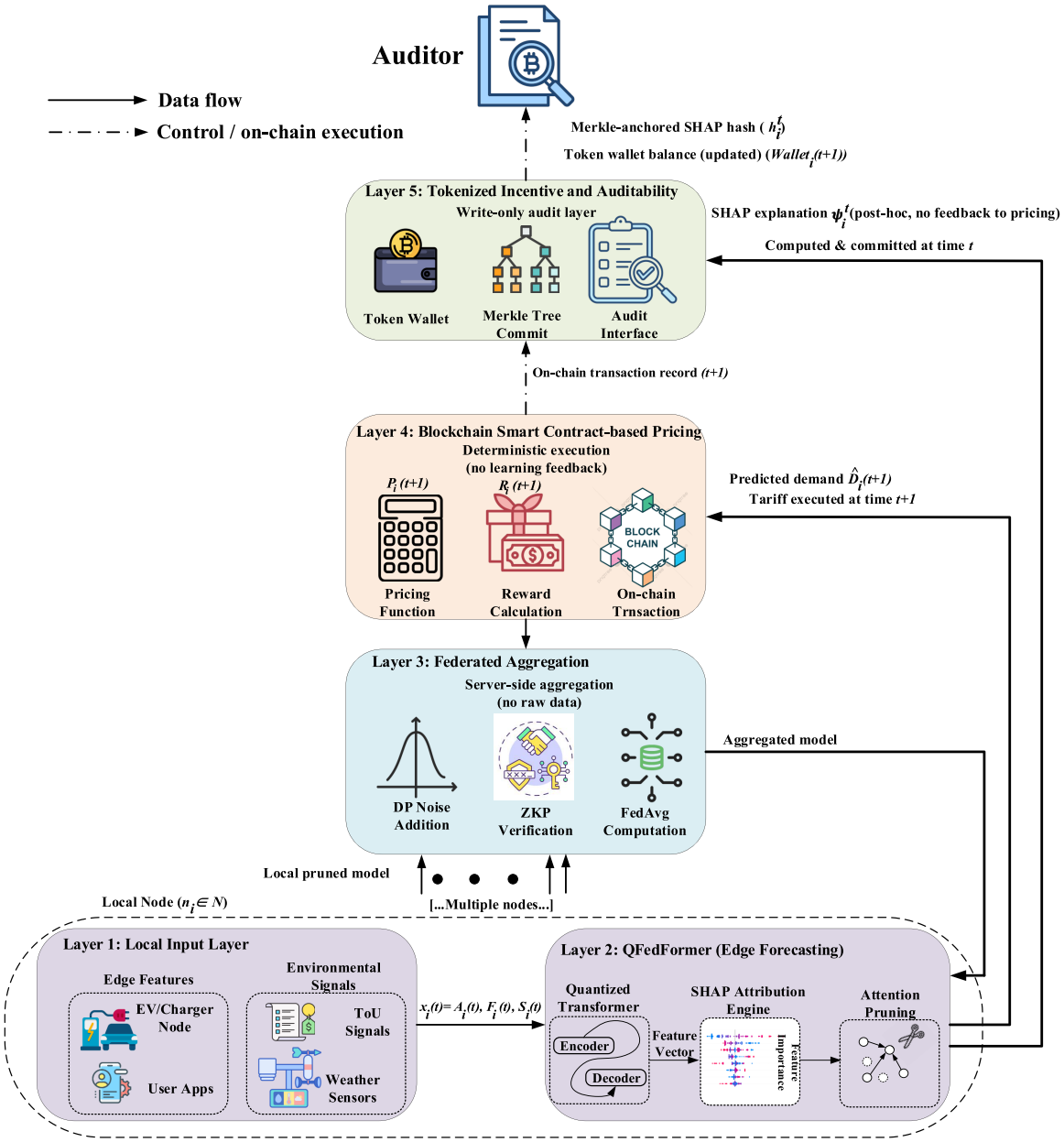


Figure 1: Overview of the QFedFormer architecture. The process follows a bottom-up sequence from local input collection to QFedFormer-based forecasting, SHAP-guided pruning, quantization, DP-protected aggregation, zk-SNARK verification, smart contract pricing, and Merkle-anchored auditability.

For clarity, Fig. 1 is interpreted as a sequential bottom-up process. First, EV charging, tariff, user, and environmental features are collected at each local node. Second, a local demand forecast is produced by QFedFormer. SHAP-guided pruning with quantization is applied to obtain a compressed local model. Third, DP noise is added before the compressed update is transmitted for federated aggregation. Fourth, zk-SNARK verification is used to check whether the submitted update follows the prescribed compression protocol.

Fifth, the verified update is aggregated, and the predicted demand is transferred to the smart contract pricing layer. Sixth, the tariff and reward are computed by the smart contract. Merkle commitments of SHAP summaries are recorded on-chain for audit purposes. The step-wise interpretation is used to clarify the link between quantized local training and blockchain-anchored verification.

3.1 Data Acquisition and Preprocessing Layer

Each edge node $n_i \in \{1, \ldots, N\}$ (e.g., residential or public charger) collects multivariate observations at time t:

$$\mathbf{x}_i(t) = [D_i(t), P_{\text{grid}}(t), A_i(t), u_i(t), \boldsymbol{\xi}_i(t)]^\top \in \mathbb{R}^d \tag{1}$$

where $D_i(t)$ is defined as the observed energy demand (kWh). $P_{\text{grid}}(t)$ is defined as the prevailing utility tariff (KRW/kWh). The operational availability of charger i is denoted by $A_i(t)$. Intrinsic user constraints are represented by $u_i(t)$, including charging urgency and battery state of charge (SoC). Exogenous contextual factors are represented by $\boldsymbol{\xi}_i(t)$, including ambient temperature, photovoltaic (PV) supply, and local congestion conditions.

Charger availability $A_i(t)$ reflects the physical connectivity and operational status of charger i. Therefore, it is included in the forecasting vector $\mathbf{x}_i(t)$.

User flexibility $F_i(t)$ is defined as an explicitly declared incentive variable. It is not used for demand forecasting. It is introduced only at the pricing and reward layers to influence tariff adjustment and token allocation (see Section 3.4).

The EV charging dataset is collected from charging stations in Jiaxing, China. All tariff-related variables are linearly normalized and expressed in KRW/kWh. This representation is used only for interpretability of price deviations. It does not affect learning behavior or relative comparison across models. No exchange-rate-based economic conversion is applied. KRW is used only as a normalized reporting unit and does not represent actual local tariffs.

A sliding window of τ steps is used to form the input tensor:

$$\mathbf{X}_i^{(t)} = [\mathbf{x}_i(t-\tau+1), \ldots, \mathbf{x}_i(t)] \in \mathbb{R}^{d \times \tau}. \tag{2}$$

3.2 QFedFormer: SHAP-Pruned Quantized Transformer Layer

Each node trains a local transformer regressor $\hat{f}_i$ to forecast next-interval demand:

$$\hat{D}_i(t+1) = \hat{f}_i\left(\mathbf{X}_i^{(t)}\right) \in \mathbb{R}_{\geq 0}. \tag{3}$$

The forecasting model is defined as an encoder–decoder Transformer adapted for time-series regression under non-IID edge data distributions.

FedFormer is selected because long-range temporal patterns are required to be captured with reduced computational cost. Quadratic complexity $\mathcal{O}(L^2)$ with respect to sequence length L is observed in standard self-attention. Frequency-domain attention is used by FedFormer, and complexity $\mathcal{O}(L \log L)$ is achieved. The cost of long time-series forecasting is therefore reduced.

Compared with Informer and Autoformer, higher suitability is observed for FedFormer in the EV charging data used in this study. Repeated daily tariff patterns, local demand peaks, and heterogeneous client behavior are contained in the data. A stable representation of trend and seasonal components is

required under non-IID FL. Therefore, FedFormer is adopted as the forecasting backbone of QFedFormer. Long-horizon temporal modeling is supported, and computational burden at edge clients is reduced.

Quantization

To support resource-constrained edge devices and reduce communication overhead, model parameters are quantized to low-precision representations. Given a trained parameter tensor θ, uniform affine quantization with bit-width $b \in \{4, 8\}$ is applied as

$$Q(\theta) = \left\lfloor \frac{\theta - \theta_{\min}}{\Delta} \right\rfloor, \quad \Delta = \frac{\theta_{\max} - \theta_{\min}}{2^b - 1}. \tag{4}$$

Post-hoc Explainability.

Explainability is incorporated strictly as a *post-hoc* mechanism. The forecast $\hat{D}_i(t+1)$ is always produced by the trained regressor $\hat{f}_i(\cdot)$; SHAP is *not* used as a predictive surrogate. Instead, SHAP is employed only to attribute the realized prediction to input features relative to a background distribution:

$$\boldsymbol{\psi}_i = \text{SHAPExplain}\left(\hat{f}_i, \mathbf{X}_i^{(t)}\right), \tag{5}$$

where $\boldsymbol{\psi}_i = [\psi_{i,1}, \ldots, \psi_{i,d}]$ denotes feature-level attributions for client i. These attributions are used exclusively for post-hoc interpretability and model-structure compression, and are not used for prediction, regression, or as a surrogate forecasting model.

SHAP-Guided Attention Pruning

SHAP attributions are leveraged to guide structural pruning of the Transformer after local training. For each attention head $h \in \mathcal{H}$, a head relevance score s_h is computed as the normalized average absolute SHAP attribution of the features predominantly attended by that head:

$$s_h = \frac{1}{|\mathcal{B}_i|} \sum_{(\mathbf{x},y)\in\mathcal{B}_i} \frac{1}{|\mathcal{F}_h|} \sum_{j\in\mathcal{F}_h} |\psi_{i,j}|, \tag{6}$$

where $\mathcal{B}_i$ is the local mini-batch and $\mathcal{F}_h$ denotes the set of input embedding dimensions with non-negligible attention weights for head h.

The set $\mathcal{F}_h$ is constructed from the attention distribution of head h as follows. Let $\alpha_{h,j}(t)$ denote the attention weight assigned by head h to embedding dimension j at time step t. Attention weights are first averaged across tokens within the local input window and across samples in the mini-batch $\mathcal{B}_i$. The set $\mathcal{F}_h$ then consists of all embedding dimensions whose mean attention weight exceeds a small threshold ϵ_a, i.e., $\mathcal{F}_h = \{j \mid \mathbb{E}[\alpha_{h,j}] > \epsilon_a\}$. It is noted that $\epsilon_a > 0$ is a small attention threshold. This construction ensures that only embedding dimensions with consistent and non-trivial contributions to the attention mechanism of head h are considered when computing s_h.

For multi-head self-attention layers, relevance scores are computed independently per layer and per client, with attention contributions averaged across tokens within the local window. Attention heads whose relevance scores fall below a fixed threshold δ are removed:

$$\mathcal{H}_{\text{pruned}} = \{h \in \mathcal{H} \mid s_h < \delta\}. \tag{7}$$

Pruning is restricted to attention-head parameters only; all remaining model weights are preserved.

Attribution-based pruning is adopted to remove structurally weak components while predictive behavior is preserved. A consistent measure of feature contribution under model-agnostic conditions is

provided by SHAP values. This property ensures that importance scores are comparable across clients with heterogeneous data distributions.

The use of SHAP is motivated by additive consistency and local accuracy. Each attribution value is defined as the marginal contribution of an input feature to the prediction outcome. Low-impact components are identified by this property without reliance on gradient magnitude or weight norm.

Attention-level mapping is applied to align feature attribution with the Transformer structure. Information from specific input dimensions is aggregated by attention heads. Therefore, the relevance of an attention head is estimated from the SHAP values of the features to which it attends.

This approach is suitable under non-IID FL settings. Local data distributions vary across clients, and gradient-based pruning may produce inconsistent patterns. SHAP-based scoring remains stable because it depends on prediction contribution rather than parameter scale.

As a result, pruning decisions are based on contribution to output rather than parameter magnitude. This property improves robustness and preserves the dominant predictive subspace across heterogeneous clients.

The computational overhead of SHAP evaluation is explicitly controlled to ensure feasibility on edge devices. SHAP values are computed on mini-batches instead of the full local dataset, which reduces computational cost. In addition, attribution is aggregated at the attention-head level rather than at the parameter level, which further limits complexity.

SHAP computation is executed after local training and is not required during real-time inference or tariff execution. Therefore, the additional cost does not affect latency-critical operations at charging stations. Feasibility of SHAP-based pruning and incentive evaluation is ensured under limited computational resources.

Compression Ordering

Explainability-driven pruning and quantization are applied in sequence to obtain the compressed local update:

$$\tilde{\theta}_i^{(r)} = \mathcal{Q}\big(\mathcal{P}_{\mathrm{SHAP}}(\theta_i^{(r)})\big). \tag{8}$$

Pruning is applied first to remove low-relevance attention heads. Uniform quantization is then applied to the remaining parameters.

This process is executed independently at each client after local training and before transmission. Client-specific sparsity patterns are produced to represent diverse demand profiles and operational contexts.

At each client, attention heads with relevance scores below the threshold δ are removed by SHAP-guided pruning. Structured sparsity is induced in the self-attention layers. Local inference computation and memory access are reduced. The full quantized model update is transmitted for federated aggregation.

As a result, lightweight and interpretable subnetworks are produced at the edge by QFedFormer. Forecasting accuracy is preserved under non-IID data and limited computational resources.

3.3 Federated Aggregation and Privacy Preservation

Let $\mathcal{D}_i$ denote client i's local dataset. At round r, each client transmits the DP-protected update $\hat{\theta}_i^{(r)}$ for weighted aggregation:

$$\theta^{(r+1)} = \sum_{i=1}^{N} \frac{|\mathcal{D}_i|}{\sum_{j=1}^{N} |\mathcal{D}_j|} \cdot \hat{\theta}_i^{(r)}. \tag{9}$$

DP

Each local update follows a fixed sequence of operations. First, SHAP-based pruning and quantization are applied to obtain $\tilde{\theta}_i^{(r)}$ as defined in Eq. (8). Second, the compressed update is clipped to norm C and is perturbed with Gaussian noise. Third, the resulting update is transmitted for aggregation.

Formally, the perturbed update is defined as:

$$\hat{\theta}_i^{(r)} = \text{clip}\left(\tilde{\theta}_i^{(r)}, C\right) + \mathcal{N}(0, \sigma^2 I). \tag{10}$$

where $C = 1$, sampling rate $p = 0.1$, $R = 50$, and and $\delta_{\text{DP}} = 10^{-5}$. Using the Rényi DP (RDP) accountant [29], noise scale $\sigma = 1.6$ yields $(\varepsilon = 2.0,\ \delta_{\text{DP}} = 10^{-5})$.

The effect of noise on learning accuracy is controlled. A moderate increase in RMSE is observed due to perturbation. The increase is limited to approximately 7.1%. Predictive performance remains stable under the selected privacy budget.

The effect of DP on audit latency is also evaluated. Additional latency is introduced by the verification of perturbed updates and proof validation. This overhead remains bounded and is reported in the evaluation section. Separation between compression and noise addition ensures that DP is applied to the transmitted representation without change to the compression structure.

Only the perturbed updates $\hat{\theta}_i^{(r)}$ are aggregated by the server. No additional modification is applied after transmission. The DP guarantee is preserved for all communicated parameters.

No SHAP attributions, raw features, forecasts, or explanation vectors are transmitted. Only compressed and perturbed model parameters are used for federated averaging.

Robustness to Malicious Updates

The aggregation process is designed to reduce the impact of malicious or corrupted client updates. Poisoning attacks are possible in federated settings when adversarial clients submit manipulated model updates to bias the global model or downstream pricing decisions.

Several design choices are used to mitigate this risk. First, stochastic perturbation is introduced by DP, and the influence of any single client update on the aggregated model is reduced. Second, compression through SHAP-based pruning and quantization limits the degrees of freedom for adversarial manipulation, as only structured and reduced representations are transmitted.

Third, zk-SNARK verification ensures that each submitted update follows the prescribed compression protocol. Invalid updates that do not satisfy pruning and quantization constraints are rejected at the smart contract level. Arbitrary or structurally inconsistent updates are prevented from entry into the aggregation process.

Full Byzantine robustness is not provided by these mechanisms. However, vulnerability to simple poisoning strategies is reduced, and consistency of client contributions is enforced. Robust aggregation rules and adversarial filtering are identified as directions for future extension.

Zero-Knowledge Proofs (ZKPs)

Each client generates zk-SNARKs π_i proving that pruning and quantization were correctly applied:

$$\texttt{Verify}(\pi_i) = \begin{cases} \texttt{True}, & \tilde{\theta}_i^{(r)} \text{ valid}, \\ \texttt{False}, & \text{otherwise}. \end{cases} \tag{11}$$

Smart contracts verify π_i on-chain, ensuring protocol compliance and resilience against malicious updates.

The proofs attest only to the correct execution of the compression protocol and do not reveal model parameters, SHAP values, or client data, nor do they certify forecast optimality. To characterize the circuit structure and computational complexity, the zk-SNARK statement enforces that the client-transmitted update is the result of the prescribed local compression pipeline: (i) reshaping the trained model and masking attention heads whose SHAP relevance scores fall below the threshold δ, (ii) applying affine post-training quantization to 8-bit precision with a fixed scale and zero-point, and (iii) computing a cryptographic commitment over the resulting compressed update and associated metadata. Accordingly, the arithmetic circuit is composed primarily of comparison and masking constraints for attention-head pruning, together with linear constraints enforcing affine quantization consistency. The total number of constraints scales linearly with the number of transmitted parameters, i.e., $\mathcal{O}(|\theta|)$, since each parameter contributes a constant number of constraints for quantization validity and commitment binding. Proof generation and on-chain verification times are reflected in the reported audit latency metrics in Section 5.5.

3.4 Smart Contract-Based Pricing Execution

Dynamic tariffs are computed by a blockchain-deployed contract consuming one-step-ahead demand forecasts $\hat{D}_i(t+1)$.

The pricing formulation is selected based on three design criteria: interpretability, bounded response, and deterministic execution under smart contract constraints. A linear structure is adopted. Each input variable is allowed to contribute in a transparent and monotone manner to the final tariff. Direct verification of pricing behavior is enabled. Ambiguity in decision rules is avoided.

The formulation is aligned with demand–supply balance in EV charging networks. The congestion term $S_i(t)$ is defined as the ratio between predicted demand and local capacity. The tariff is adjusted in proportion to the expected load level. Prices are increased under high demand. Prices are reduced when capacity is sufficient. Load balance at the feeder level is supported.

Real-time pricing adaptation is achieved through one-step-ahead forecasts $\hat{D}_i(t+1)$. The tariff at time $t+1$ is determined by the predicted demand at time t. A response to short-term variation is ensured without iterative updates. Stable and timely responses are achieved under non-IID demand conditions.

User behavior is incorporated through the flexibility variable $F_i(t)$. Lower tariffs are assigned to flexible users. Demand shift is encouraged, and peak load is reduced. A direct and controllable link between user participation and pricing incentives is established. Compatibility with smart grid operation is maintained. The pricing rule is based only on local measurements and predicted demand. Integration with distributed grid control is enabled without centralized coordination.

Existing studies in EV charging pricing often rely on reinforcement learning or game-theoretic optimization. Pricing decisions are determined through iterative updates or equilibrium computation. Multiple interaction steps are required. Stable outcomes are not guaranteed under non-IID demand conditions. In contrast, a single-step deterministic output is produced by the proposed formulation. Suitability for real-time on-chain execution is ensured.

A bounded structure is required for safe deployment. Extreme values may be produced by adaptive pricing rules under demand spikes when explicit bounds are not applied. The clipping operator is used to enforce operational limits. Tariffs are ensured to remain within acceptable ranges.

The additive form is used to separate congestion, availability, and flexibility effects. Direct control of pricing sensitivity is enabled through parameters (α, β, η). Consistent behavior is supported across heterogeneous charging nodes.

The contract implements a single bounded tariff update rule:

$$P_i(t+1) = \mathrm{clip}_{[P_{\min}, P_{\max}]}\left(P_0 + \alpha \cdot S_i(t) - \beta_p \cdot A_i(t) - \eta \cdot F_i(t)\right) \tag{12}$$

where P_0 (KRW/kWh) is the baseline tariff. A dimensionless congestion stress score is denoted by $S_i(t)$. Charger availability is denoted by $A_i(t) \in [0, 1]$. A user-declared flexibility indicator is denoted by $F_i(t) \in [0, 1]$. Parameters $\alpha, \beta_p, \eta > 0$ control surge pricing, availability discounts, and flexibility incentives.

Tariff safety limits are enforced by the clipping bounds $[P_{\min}, P_{\max}]$. A lower tariff is assigned to users with higher flexibility. The reduction in tariff represents the economic benefit associated with flexible behavior. The flexibility variable $F_i(t)$ influences token rewards only through its contribution to the utility score defined in Section 3.6. No direct reward is assigned at the pricing layer.

The stress score is defined as:

$$S_i(t) = \frac{\hat{D}_i(t+1)}{D_{\mathrm{cap},i}} \cdot \omega_i(t), \tag{13}$$

where $D_{\mathrm{cap},i}$ is the rated capacity of charger i, and $\omega_i(t) \in [0, 1]$ is a normalized feeder-level congestion weight.

The tariff is monotone increasing in $S_i(t)$ and monotone decreasing in $A_i(t)$ and $F_i(t)$. Predictable incentives are ensured under congestion. Pricing depends only on bounded one-step-ahead forecasts and local context. Inter-temporal arbitrage and oscillatory behavior are avoided. Tariffs are executed deterministically at time $t+1$ based only on forecasts computed at time t.

The operational transfer process from federated demand forecasting to smart-contract pricing and blockchain auditability is illustrated in Fig. 2.

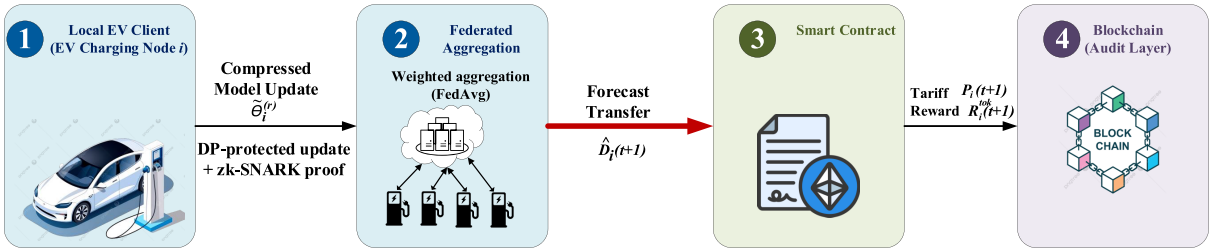


Figure 2: Transfer architecture of QFedFormer from local EV charging nodes to federated aggregation, smart-contract pricing execution, and blockchain-based auditability.

Compared to reinforcement learning-based pricing, policy training or exploration is not required by the proposed rule. Non-stationary pricing outputs are produced by RL-based methods, and dependence on reward design is observed. Instability may be introduced by this dependence in decentralized environments. In contrast, consistent responses are produced for identical inputs by the present formulation.

Game-theoretic pricing models assume rational agents and equilibrium convergence. These assumptions are difficult to satisfy in EV charging scenarios with heterogeneous user behavior. Computational overhead is introduced by equilibrium computation, which is not compatible with real-time smart contract execution.

Auction-based mechanisms require iterative bidding and communication among participants. Latency and communication cost are increased under large-scale deployment. In contrast, only local inputs and one-step-ahead forecasts are required by the proposed pricing rule. Low computational cost and deterministic execution are ensured. Stable tariff updates are obtained without iterative optimization or multi-round interaction.

Parameter Calibration

Pricing parameters (α, β, η) and bounds $(P_{\min}, P_{\max})$ are calibrated offline using historical time-of-use (ToU) tariffs and grid-constrained simulations to ensure realistic price ranges and compliance with operational limits. Sensitivity analyses with respect to these parameters are reported in the evaluation section.

Billing records are stored as signed tuples $\mathcal{T}_i^{(t+1)} = (\mathrm{TxID}_i, P_i(t+1), \hat{D}_i(t+1), t+1)$, anchored on-chain for auditability.

3.5 Stability and Theoretical Considerations

Closed-Loop Stability

The proposed design operates in a receding-horizon manner. Demand forecasts $\hat{D}_i(t+1)$ are computed at time t. Tariffs for the next interval are determined without intra-interval feedback. The pricing function in Eq. (12) is bounded and monotone. Dependence is limited to one-step-ahead forecasts and locally bounded context variables. Smooth evolution of the demand–price interaction over time is ensured under bounded user price elasticity. Rapid oscillations are mitigated, and arbitrage across consecutive pricing intervals is prevented.

Effect of SHAP-Guided Pruning

Attention heads with consistently low attribution magnitude are removed by SHAP-guided pruning. All high-contribution pathways in the trained Transformer are preserved. Low-SHAP heads are assumed to correspond to redundant or weakly informative representations. The dominant predictive subspace of the model is preserved by pruning. The fixed point of the forecast–pricing interaction is unchanged up to a bounded approximation error. This behavior is validated in the evaluation section.

3.6 Tokenized Incentives and Auditability

Each user's incentive utility is defined as a combination of flexibility, participation, and data quality:

$$U_i^{\mathrm{inc}}(t) = \beta_1^{\mathrm{inc}} F_i(t) + \beta_2^{\mathrm{inc}} \rho_i(t) + \beta_3^{\mathrm{inc}} Q_i(t). \tag{14}$$

The weights β_1^{inc}, β_2^{inc}, and β_3^{inc} are tuned via grid search to align token distribution with performance and fairness KPIs. Token rewards are issued as

$$R_i^{\mathrm{tok}}(t) = T_{\max} \cdot \frac{U_i^{\mathrm{inc}}(t)}{U_{\max} + \epsilon_{\mathrm{stab}}}, \tag{15}$$

where $\epsilon_{\mathrm{stab}} > 0$ is a numerical stabilizer. $U_{\max} = \max_i U_i^{\mathrm{inc}}(t)$ is the maximum utility value across participating clients at time t. For explainability, each prediction $\hat{D}_i(t+1)$ is paired with SHAP attributions $\boldsymbol{\psi}_i$. These values are hashed as:

$$h_i^{(t)} = \mathrm{SHA256}(\hat{D}_i(t+1) \| \boldsymbol{\psi}_i \| \mathcal{M}_i), \tag{16}$$

where $\mathcal{M}_i$ is metadata. SHAP attributions are used only for post-hoc auditability and incentive verification. Feedback to the pricing function or the learning process is not provided.

Merkle roots of these hashes are published once per round. Tamper-evident auditability is ensured. Auditors can request proofs to verify explanations without access to raw data.

Blockchain-Anchored Incentive Design

The incentive mechanism is implemented through smart contract execution with deterministic pricing and reward allocation rules. On-chain functions are triggered at each audit round. Tariffs are computed, and rewards are distributed based on verified client contributions.

Incentive distribution is based on the utility score defined in Eq. (14). Flexibility, participation, and data quality are combined. Tokenization is used to represent rewards. Tokens are assigned to each client in proportion to its validated contribution under predefined bounds. Trust and transparency are ensured through on-chain execution and immutable storage of transaction records. Verifiable audit trails are provided by Merkle-root commitments of SHAP summaries without exposure of sensitive information.

Security and attack resistance are supported through multiple mechanisms. The influence of individual updates is limited by differential privacy. Correct execution of the local compression protocol is enforced by zk-SNARK verification. Invalid or inconsistent updates are prevented from affecting pricing and reward allocation by smart contract constraints. A transparent, verifiable, and secure incentive mechanism is ensured under decentralized EV charging environments.

Algorithm 1 summarizes the entire process. Local QFedFormer training with SHAP pruning and DP, Merkle anchoring, and token issuance are included.

Algorithm 1: Federated training with blockchain audit and token incentives.

Initialize: global model θ_0, token ledger $L \leftarrow \varnothing$
for $t = 1$ to T **do**
  Sample client subset $\mathcal{S} \subset \{1, \ldots, N\}$ with participation rate p
  for all $n_i \in \mathcal{S}$ **in parallel do**
    Receive global model θ_{t-1}
    $\hat{\theta}_i, \hat{D}_i(\cdot), \boldsymbol{\psi}_i \leftarrow \textsc{LocalUpdate}(\theta_{t-1}, \mathcal{D}_i)$
    $U_i^{\text{inc}}(t) \leftarrow \textsc{UtilityScore}(\boldsymbol{\psi}_i, \mathcal{D}_i, F_i(t))$ ▷ Eq. (14)
    $h_i \leftarrow \text{SHA256}\big(\hat{D}_i(t{+}1) \| \boldsymbol{\psi}_i \| \mathcal{M}_i\big)$ ▷ Merkle anchoring input; post-hoc audit only
    Submit $(\hat{\theta}_i, h_i, U_i^{\text{inc}}(t))$ to server
  end for
  $\theta_t \leftarrow \textsc{Aggregate}\big(\{\hat{\theta}_i \mid n_i \in \mathcal{S}\}\big)$ ▷ masked averaging over DP-protected updates
  Commit Merkle root of $\{h_i\}$ to blockchain ▷ post-hoc audit; no feedback to learning or pricing
  for all $n_i \in \mathcal{S}$ **do**
    $R_i^{\text{tok}}(t) \leftarrow \textsc{IssueTokens}(U_i^{\text{inc}}(t))$ ▷ Eq. (15)
    $L[n_i] \leftarrow L[n_i] + R_i^{\text{tok}}(t)$
  end for
end for
return θ_T, token ledger L
function LOCALUPDATE($\theta_{t-1}, \mathcal{D}_i$)
  Local train: $\theta_i \leftarrow \text{SGD/AdamTrain}(\theta_{t-1}, \mathcal{D}_i, E)$

(Continued)

Algorithm 1 (continued)

Explainability-driven compression: $\tilde{\theta}_i \leftarrow \mathcal{Q} \circ \mathcal{P}_{\text{SHAP}}(\theta_i)$ ▷ SHAP-prune + b-bit quantize; Eqs. (7) and (8)
Local inference: compute $\hat{D}_i(\cdot)$ using $\tilde{\theta}_i$
SHAP explain: $\psi_i \leftarrow \text{SHAPExplain}(\tilde{\theta}_i, \mathcal{D}_i)$ ▷ post-hoc explanation of local forecasts
Client-side DP: $\hat{\theta}_i \leftarrow \text{clip}(\tilde{\theta}_i, C) + \mathcal{N}(0, \sigma^2 I)$ ▷ apply Eq. (10) after compression, before transmission
return $\hat{\theta}_i$, $\hat{D}_i(\cdot)$, ψ_i
end function
function UTILITYSCORE($\psi_i, \mathcal{D}_i, F_i(t)$)
Compute $C_i(t)$ from validated client participation
Compute $Q_i(t)$ from local data-quality indicators
return $U_i^{\text{inc}}(t)$ per Eq. (14) ▷ combines flexibility, participation, and data quality
end function

4 Experimental Setup and Results

This section provides a structured description of the dataset, simulation environment, implementation details, evaluation metrics, and baseline models used for performance evaluation.

4.1 Data Preparation and Simulation Environment

For the performance evaluation of the proposed QFedFormer framework for decentralized dynamic pricing in EV charging networks, we implemented a high-fidelity simulation environment based on the real-world dataset from Jiaxing, China [30,31]. The dataset is used as a basis for simulation of non-IID FL across dispersed clients. Comprehensive EV charging activity from public charging stations is included.

For each charging session, start time, end time, charged energy (kWh), geographic coordinates, ToU pricing signals, station identifiers, service fees, and user behavior indicators such as interruptions and cancellations are provided [30,31]. A total of 24 structured features are included. Local meteorological variables are also included to capture exogenous effects on charging behavior. Demand-side heterogeneity, charging flexibility, and contextual dependencies are represented by this feature set. These factors are essential for accurate utility scoring and federated price forecasting.

A fixed ToU pricing schedule is included in the dataset to represent conventional pricing behavior. These rates are not used as fixed inputs. They are used as reference signals from which dynamic pricing patterns are derived. The regional ToU schedule, shown in Table 3, divides the day into seven temporal intervals. Prices range from off-peak values of 73.10 KRW/kWh to peak values of 232.84 KRW/kWh. These values are obtained through linear rescaling of the original tariff data to a KRW/kWh reporting range. No exchange-rate-based economic conversion is applied.

Table 3: Reference ToU schedule used as a baseline signal in model training (KRW/kWh).

Time Period	Tariff (KRW/kWh)
00:00–08:00	73.10
08:00–11:00	173.96
11:00–13:00	73.10
13:00–19:00	173.96

(Continued)

Table 3 (continued)

Time Period	Tariff (KRW/kWh)
19:00–21:00	232.84
21:00–22:00	173.96
22:00–24:00	73.10

Real-world policy varies by season and by weekday or weekend. However, this representative structure is adopted based on the original dataset [30,31]. This choice supports the forecasting task and allows deviations and optimal pricing schemes to be identified under different demand conditions.

Daily weather conditions such as temperature, humidity, and precipitation are aligned with session timestamps and appended to each sample. These exogenous variables influence energy consumption and user preferences. More robust and context-aware model training is achieved.

For the federated simulation, the data is split among the 100 clients, each of which simulates a different charging point or geographic area. This division is done in a way that a non-IID data distribution is ensured for each client. This is done by retaining the local charging patterns, as well as weather and demographic influences, for each client. This closely represents the current EV infrastructure setup, which is quite different for each charging point, thereby presenting a challenge for FL's dynamic pricing.

4.2 Implementation Details

QFedFormer is implemented in PyTorch v2.1 with TensorFlow Federated for training orchestration and SHAP v0.42 for attribution-based pruning. Smart contracts and token mechanisms are implemented in Solidity v0.8 and deployed on a Hyperledger Besu PoA blockchain emulator.

Each client trains a transformer model with 2 encoder layers, 4 attention heads, and 128-dimensional hidden states. SHAP pruning and 8-bit affine quantization are applied prior to transmission. DP is enforced with parameters $(C = 1, \sigma = 1.6, \varepsilon = 2.0, \delta = 10^{-5})$.

Detailed hardware configuration, communication cost formulation, and blockchain cost modeling are provided in Appendix A.

Training is performed for $R = 50$ rounds with 2 local epochs per client, batch size of 32, and the Adam optimizer with learning rate 10^{-3} and weight decay 10^{-5}. Demand-side flexibility is modeled using a ToU tariff schedule from Jiaxing, China, normalized and mapped to KRW/kWh as described in Section 3.1. Dynamic-pricing smart contracts anchor batched predictions and SHAP vectors on-chain as SHA-256 Merkle roots.

The weighting parameters $\beta_1^{\text{inc}}, \beta_2^{\text{inc}}, \beta_3^{\text{inc}}$ and $\gamma_1, \gamma_2, \gamma_3$ are selected via normalized grid search under the constraints $\sum \beta_i^{\text{inc}} = 1$ and $\sum \gamma_i = 1$. The weights $\gamma_1, \gamma_2, \gamma_3$ are used only for evaluation and are distinct from the incentive weights β_i^{inc} defined in Section 3.6. The final configuration $\beta_1^{\text{inc}} = 0.4, \beta_2^{\text{inc}} = 0.3, \beta_3^{\text{inc}} = 0.3$ and $\gamma_1 = 0.5$, $\gamma_2 = 0.3$, $\gamma_3 = 0.2$ is used in all experiments. Detailed search space and sensitivity analysis are provided in Appendix A.

Blockchain performance is evaluated using a PoA-based emulator. Audit latency and transaction throughput are measured under controlled load conditions. Detailed cost formulation and gas modeling are provided in Appendix A.

4.3 Evaluation Metrics

QFedFormer is assessed across five dimensions that match its design goals: predictive accuracy, privacy, auditability, fairness, and client-level benefit.

4.3.1 Predictive Accuracy

Predictive performance is measured by root mean square error (RMSE) and mean absolute error (MAE):

$$\text{RMSE} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(\hat{D}_i(t) - D_i(t)\right)^2}, \tag{17}$$

$$\text{MAE} = \frac{1}{T}\sum_{t=1}^{T}\left|\hat{D}_i(t) - D_i(t)\right|, \tag{18}$$

where $\hat{D}_i(t)$ and $D_i(t)$ denote predicted and actual demands, respectively.

4.3.2 Privacy

Client-level privacy is measured by the cumulative DP budget ε. This value is tracked with an RDP accountant under fixed parameters (C, p, R, δ).

4.3.3 Auditability

Audit latency is defined as the average verification time for blockchain-anchored model updates and SHAP attributions. Validation is performed through Merkle tree proofs and zero-knowledge checks.

4.3.4 Fairness

Incentive equity is assessed by the Gini coefficient over token balances:

$$\text{Gini} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}|T_i - T_j|}{2N\sum_{i=1}^{N}T_i}, \tag{19}$$

where T_i is the token balance of client i. Lower values indicate a more uniform token distribution across participants.

4.3.5 Client Utility

An evaluation-oriented utility score is defined as

$$U_i^{\text{eval}} = \gamma_1\left(1 - \widetilde{\text{MAE}}_i\right) + \gamma_2 F_i + \gamma_3 \frac{T_i}{T_{\text{total}}}, \tag{20}$$

where $\widetilde{\text{MAE}}_i \in [0,1]$ denotes the normalized MAE of client i, F_i denotes its declared flexibility, and T_i/T_{total} denotes its relative token share. The weights $\gamma_1, \gamma_2, \gamma_3$ are model-level parameters. They are distinct from the weights used for token issuance in Section 3.6.

4.3.6 Grid-Level Pricing Impact

In addition to predictive accuracy, four complementary metrics are used to assess the effect of the pricing mechanism on grid-level objectives. The relative decrease in maximum aggregate demand is used as

a measure of peak reduction:

$$\Delta P_{\text{peak}} = \frac{P_{\max}^{\text{static}} - P_{\max}^{\text{dynamic}}}{P_{\max}^{\text{static}}}. \tag{21}$$

Load smoothness is measured by the decrease in demand variance over time:

$$\Delta\sigma^2 = \frac{\text{Var}(D_{\text{static}}) - \text{Var}(D_{\text{dynamic}})}{\text{Var}(D_{\text{static}})}. \tag{22}$$

Price volatility is assessed by the total variation of the tariff trajectory:

$$\text{TV}(P) = \sum_{t=1}^{T-1} |P(t+1) - P(t)|, \tag{23}$$

which captures price changes across time intervals. User welfare is approximated by the average charging cost per session:

$$C_{\text{avg}} = \frac{1}{N} \sum_{i=1}^{N} P_i(t)\, D_i(t), \tag{24}$$

which reflects the economic effect of dynamic pricing on end users.

These metrics provide a multidimensional evaluation of QFedFormer. Accuracy, privacy, transparency, fairness, incentive alignment, and grid-level operational impact are assessed jointly.

4.4 Comparative Baselines

QFedFormer is evaluated against representative baselines from non-learning methods, standard FL, and blockchain-enhanced approaches. Static Pricing uses a fixed normalized tariff level within the KRW/kWh reporting range for nonadaptive comparison. *FedAvg* [32] and *FedAvg+DP* are used as standard FL baselines. In *FedAvg+DP*, Gaussian noise is added to achieve (ε, δ)-DP. Auditability and interpretability are not provided in this setting. *Block-FeDL* [13] uses blockchain to ensure model provenance in EV forecasting. Dynamic pricing and incentive design are not considered. *FedPT-V2G* [9] is a transformer-based FL model adapted for G2V forecasting from V2G. QFedFormer combines quantized FL with DP, SHAP-based pruning, blockchain-based incentives, and Merkle-root-based auditability. All models are trained and tested on the same dataset with identical privacy parameters and hardware settings. Architectural differences are preserved.

5 Results and Discussion

In this section, prediction accuracy, privacy preservation, incentive mechanism effectiveness, scalability, communication overhead, and comparative performance against baseline methods are analyzed.

All reported results are based on mean values over 15 independent runs. Variability is reported as standard deviation or is indicated by error bars in the figures.

5.1 Forecast Accuracy vs. Model Efficiency

The effect of model compression on inference efficiency and forecast accuracy is evaluated. The QFedFormer model is tested under three quantization levels: full-precision (32-bit), 8-bit, and 4-bit. A trade-off frontier is shown in Fig. 3. Each point is associated with a quantization level and its corresponding RMSE value.

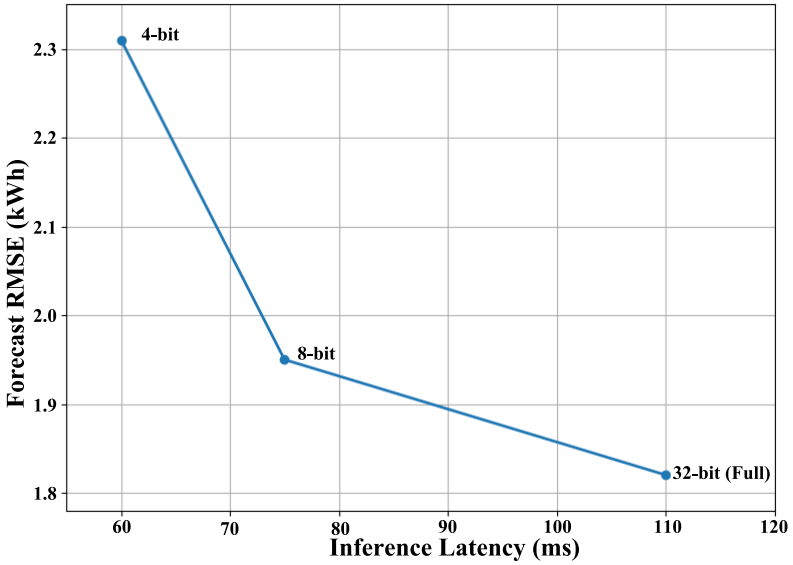


Figure 3: Trade-off frontier between forecast accuracy and latency across quantized models. Each point represents the mean over 15 independent runs. Variability across runs remains within ±1 standard deviation.

Fig. 3 shows that 8-bit quantization is the most favorable operating point. Latency is reduced from 110 to 75 ms, while RMSE is increased from 1.82 ± 0.01 to 1.95 ± 0.01 kWh. A latency reduction of 32% is achieved with an accuracy loss of 7.1%. Most efficiency gain is obtained before severe accuracy degradation is observed.

The 4-bit case shows different behavior. Latency is further reduced, but RMSE is increased to 2.31 kWh. This result indicates that useful temporal information is removed by excessive compression. Therefore, 8-bit quantization is selected as the practical setting because forecast quality is preserved and edge inference cost is reduced.

Table 4 provides a quantitative comparison of forecasting accuracy under different quantization levels. RMSE and MAPE are reported to evaluate the effect of model compression on prediction quality.

Table 4: Effect of quantization level on forecasting accuracy (mean ± std over 15 runs).

Quantization Level	RMSE (kWh)	MAPE (%)
32-bit (Full Precision)	1.82 ± 0.02	2.7 ± 0.2
8-bit	1.95 ± 0.01	3.1 ± 0.2
4-bit	2.31 ± 0.02	3.9 ± 0.3

RMSE is increased from 1.82 kWh at 32-bit precision to 1.95 kWh at 8-bit precision and 2.31 kWh at 4-bit precision. A similar trend is observed for MAPE. This result confirms that accuracy is preserved under moderate quantization, while predictive fidelity is reduced under aggressive compression.

To isolate the effect of SHAP-guided structural pruning from quantization, the quantization level is fixed at 8-bit, and the privacy budget is fixed at $\varepsilon = 2.0$. Only the pruning threshold δ_{prune} is varied. Fig. 4 shows the resulting RMSE and the fraction of pruned parameters.

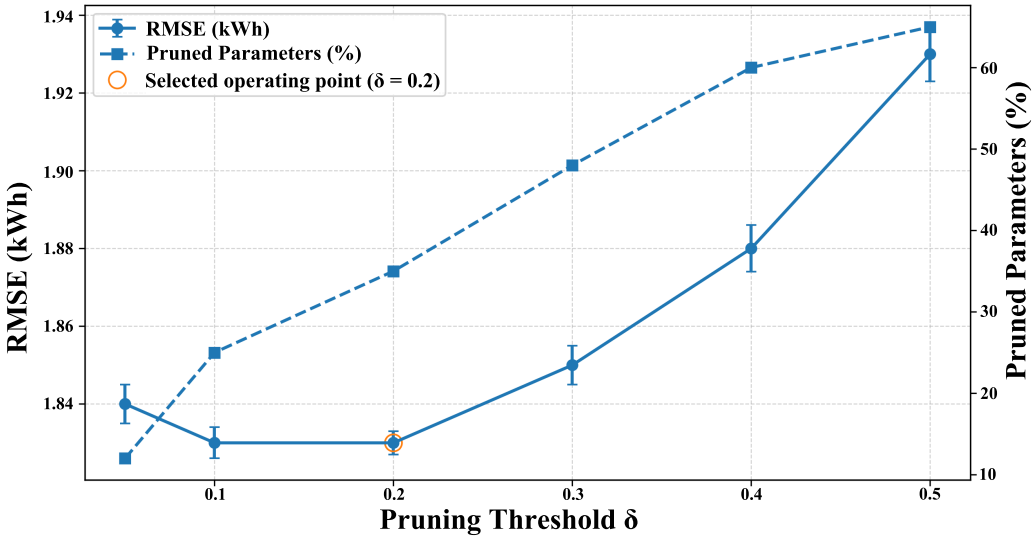


Figure 4: Privacy–utility trade-off curve showing RMSE and audit latency across privacy levels. Points represent mean values over 15 independent runs.

As δ_{prune} increases, attention heads with low relevance scores are removed. Model sparsity is increased monotonically. The value of δ_{prune} is selected to maximize sparsity under a bounded accuracy loss. Specifically, $\delta_{\text{prune}} = 0.20$ is selected as the smallest value such that $\text{RMSE}(\delta_{\text{prune}}) \leq (1 + \epsilon_{\text{tol}})$ $\min_{\delta'_{\text{prune}}} \text{RMSE}(\delta'_{\text{prune}})$, with $\epsilon_{\text{tol}} = 1.5\%$.

Under this setting, approximately 33% pruning is achieved. RMSE is maintained within approximately 1.1% of the minimum value. A reduction in model size is achieved with limited impact on forecasting accuracy.

5.2 SHAP Faithfulness and Audit Stability

A feature-deletion test is used to assess the faithfulness of SHAP explanations. Input features are removed progressively in descending order of absolute SHAP attribution. The resulting degradation in forecasting accuracy is recorded. A steeper error increase is interpreted as higher explanation faithfulness.

Across clients, RMSE is increased by 18.6% when the top 20% of SHAP-ranked features are removed. In contrast, an increase of 6.1% is observed under random feature removal. These values correspond to mean behavior over 15 runs.

Explanation stability is assessed by the Spearman rank correlation of SHAP feature importance across consecutive training rounds. An average rank correlation above 0.82 is observed. Stability is therefore maintained under DP noise and federated aggregation.

This stability is required for on-chain anchoring. It ensures that SHAP hashes reflect persistent model behavior rather than transient training noise.

5.3 Component Ablation Study

Table 5 shows the role of each component in the proposed method. When SHAP-guided pruning is replaced with random pruning at the same sparsity ratio, RMSE increases from 1.82 to 1.91 kWh. This result indicates that attribution-based head selection preserves more useful temporal representations than random removal.

Table 5: Component ablation analysis (mean ± std over 15 runs).

Configuration	RMSE (kWh)	Audit Latency (ms)	Token Gini
QFedFormer (Full)	1.82 ± 0.02	58 ± 2	0.42
w/o SHAP (Random prune, 33%)	1.91 ± 0.03	58 ± 2	0.42
w/o DP ($\varepsilon \to \infty$)	1.75 ± 0.02	54 ± 2	0.41
w/o Blockchain (Fed-only)	1.80 ± 0.02	N/A	N/A

The no-DP case improves RMSE to 1.75 kWh, but the privacy guarantee is removed. This result confirms the expected privacy–utility trade-off. The Fed-only case maintains similar forecast accuracy, but audit latency and token fairness are not available. This result shows that blockchain mainly contributes to auditability and incentive traceability rather than direct forecast improvement.

5.4 Privacy–Utility Trade-Off under DP

DP in QFedFormer provides the guarantee that client updates are computationally indistinguishable. Although DP helps to alleviate concerns regarding data leakage, it increases the Gaussian noise involved in learning and may have an impact on prediction accuracy as well as blockchain auditing latency.

To measure the strength of this privacy-accuracy trade-off, the privacy budget ε is adjusted from a high-privacy scenario with $\varepsilon = 0.1$ to a low-privacy comparison method with $\varepsilon = 6.0$, based on the adjustment of the noise scale σ while maintaining a fixed setting of $C = 1$, client sampling probability $p = 0.1$, $R = 50$, and $\delta_{\mathrm{DP}} = 10^{-5} = 10^{-5}$. These values were calculated using the RDP accountant as seen in Section 3.3. The deployment configuration in this study is calibrated at $\sigma = 1.6$ to achieve $(\varepsilon = 2.0, \delta_{\mathrm{DP}} = 10^{-5} = 10^{-5})$ client-level DP.

As shown in Fig. 5, decreasing ε produces a monotonic increase in RMSE and audit latency. RMSE increases from 1.82 kWh at $\varepsilon = 6.0$ to 2.16 kWh at $\varepsilon = 0.1$, based on mean values over 15 runs. Audit latency increases from 52 to 72 ms due to additional cryptographic operations and randomized masking.

At the calibrated deployment point $\varepsilon = 2.0$, RMSE is 1.95 kWh, which corresponds to an increase of approximately 7.1% from the baseline, while audit latency is 58 ms. This result indicates a balanced trade-off between prediction accuracy and privacy.

Fig. 5 indicates that the selected value $\varepsilon = 2.0$ provides a balanced privacy setting. At this point, the accuracy loss remains limited, and audit latency remains suitable for short-interval EV pricing. Lower values of ε provide stronger privacy, but the error and latency increase. Higher values improve accuracy, but the privacy guarantee is weaker. Thus, $\varepsilon = 2.0$ is used as the deployment setting in the remaining experiments.

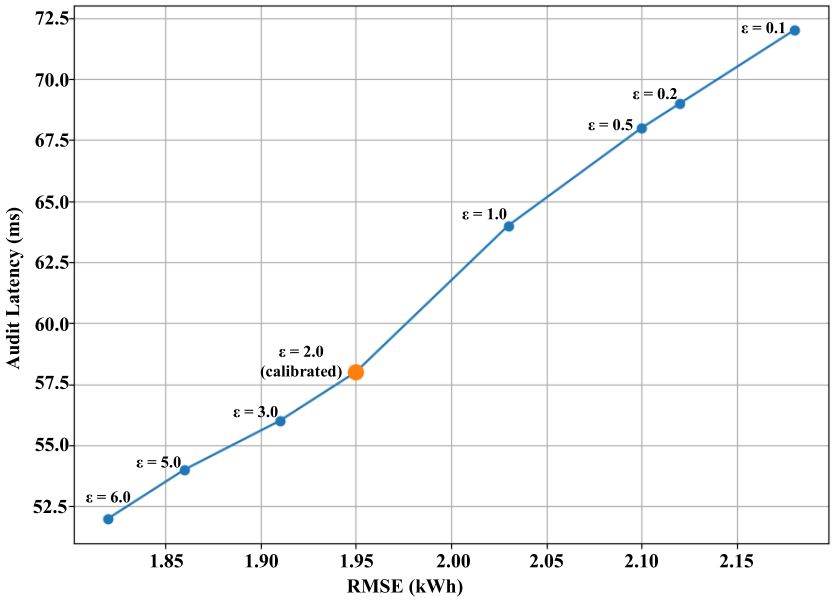


Figure 5: Privacy–utility trade-off curve: RMSE vs. audit latency, annotated by privacy budget ε.

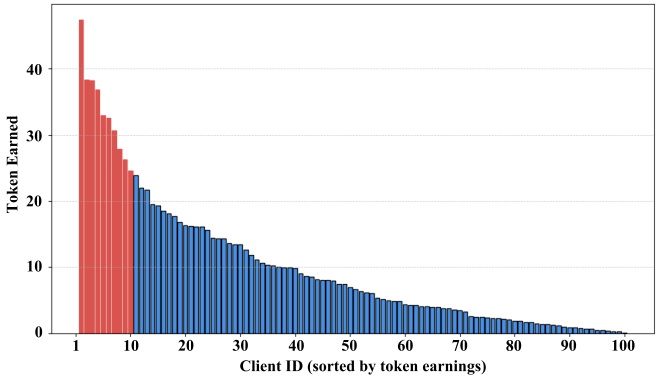
This analysis provides a sensitivity characterization of the privacy budget. Small changes in ε lead to predictable variations in both accuracy and audit latency. Therefore, the privacy parameter can be tuned according to application requirements, such as stricter privacy constraints or higher prediction accuracy.

5.5 Token Incentives and Auditability Performance

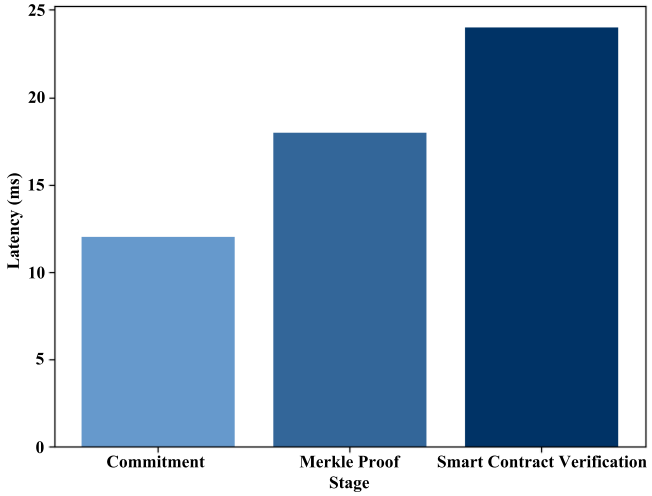
Fig. 6 evaluates whether the incentive and audit mechanisms remain practical under the selected privacy setting. Three outcomes are analyzed: reward concentration, verification latency, and cumulative token share. These outcomes indicate whether token allocation is useful, auditable, and sufficiently balanced across clients.

Fig. 6a shows token allocation to 100 users based on the utility score defined in Eq. (14). More than 55% of the total tokens is assigned to the top 10 users, based on mean values across runs. A Gini index of 0.42 indicates a moderate level of inequality. This result shows that reward allocation follows the contribution pattern. However, concentration among top users is observed. A control mechanism is therefore required. A cap on accumulation or an adaptive rebalancing rule can be applied to improve fairness.

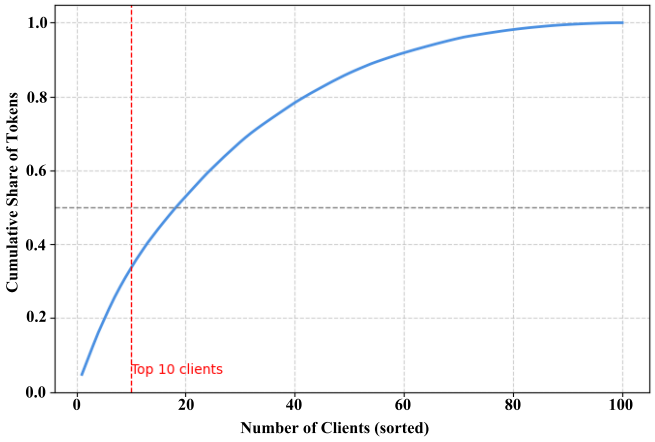
Accuracy measures in demand forecasting are reported under *no-DP* conditions ($\varepsilon \to \infty$) to avoid mixing accuracy and privacy-related overhead. Blockchain audit latency is evaluated under the DP setting ($\varepsilon = 2.0, \delta_{\mathrm{DP}} = 10^{-5}$).



(**a**) Token distribution by utility score. Top 10 clients are highlighted.



(**b**) Audit latency by blockchain verification stage.



(**c**) Cumulative token share by client rank.

Figure 6: Evaluation of token incentives and auditability. Quantitative values are computed over 15 independent runs where applicable.

A comparative summary of token fairness, measured by the Gini coefficient, and on-chain audit latency for all baseline methods is presented in Table 6. A reference for incentive concentration and audit latency is provided.

Table 6: Comparative forecasting performance and on-chain auditability (mean ± std over 15 runs).

Model	RMSE (kWh)	MAE (kWh)	Audit Latency (ms)	Token Gini
Static Pricing	2.45 ± 0.00	2.03 ± 0.00	N/A	N/A
FedAvg	2.12 ± 0.02	1.79 ± 0.02	N/A	N/A
FedAvg + DP	2.26 ± 0.02	1.91 ± 0.03	N/A	N/A
FedProx	2.05 ± 0.02	1.74 ± 0.02	N/A	N/A
SCAFFOLD	2.03 ± 0.02	1.72 ± 0.02	N/A	N/A
Block-FeDL	2.01 ± 0.01	1.71 ± 0.02	44 ± 1	N/A
FedPT-V2G	1.97 ± 0.02	1.68 ± 0.01	N/A	N/A
QFedFormer (Ours, no DP)	1.82 ± 0.01*	1.55 ± 0.01*	N/A	0.42
QFedFormer (Ours, DP $\varepsilon = 2.0$)	1.95 ± 0.01	1.62 ± 0.01	58 ± 1	0.42

Note: *Statistical testing is performed on run-level RMSE and MAE using a paired t-test across identical random seeds (same data split and initialization per run), with $p < 0.05$. Forecasting metrics are reported under the non-private training regime ($\varepsilon \to \infty$), while on-chain audit latency is measured under the deployed DP setting ($\varepsilon = 2.0,\ \delta = 10^{-5}$).

Fig. 6b shows the decomposition of average audit latency across three blockchain verification stages: SHA-256 commitment, Merkle proof generation, and smart contract verification. A total latency of approximately 58 ms per round is observed at the deployed privacy level ($\varepsilon = 2.0$). A baseline latency of approximately 52 ms is observed at $\varepsilon = 6.0$.

The cumulative token distribution for each client rank is shown in Fig. 6c. A linear trend is observed. This result indicates that a minority fraction of clients receives a large share of the tokens. This observation is consistent with Fig. 6a.

These results support the viability of tokenized incentives as a participation motivator and a transparency mechanism. Merkle-root-based audit logs and on-chain verification enable tamper-evident tracking of model contributions and support regulatory alignment. To mitigate risks such as reward manipulation or Sybil attacks, additional mechanisms can be applied in future work. Examples include client rate limiting, stake-based participation, and privacy-preserving identity attestation. These mechanisms complement the zk-SNARK-based protocol verification mechanism.

5.6 Comparative Evaluation with Baseline Models

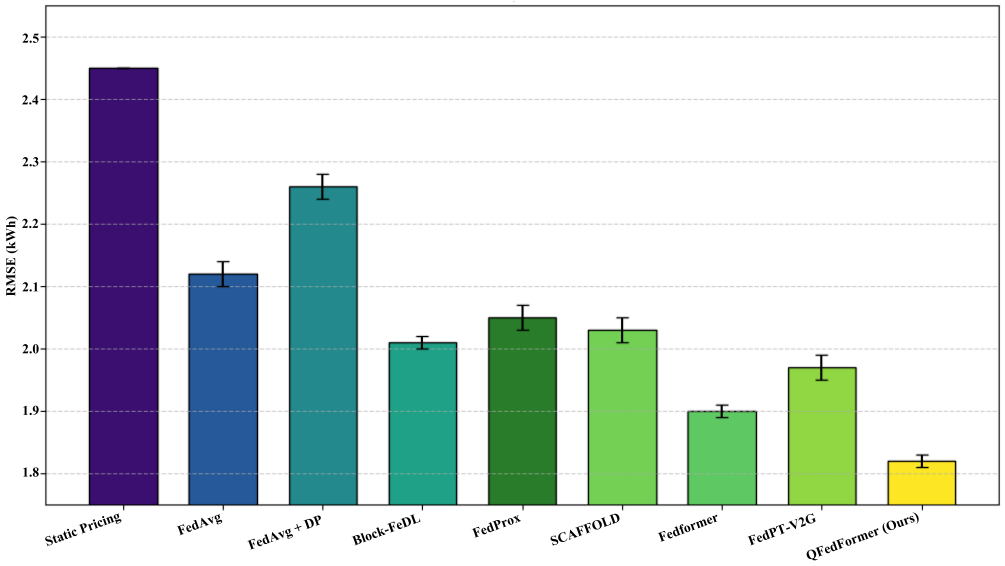
QFedFormer is compared with Static Pricing, FedAvg, FedAvg+DP, Block-FeDL, FedPT-V2G, FedProx, SCAFFOLD, and FedFormer. These baselines are selected to isolate the effects of standard FL, DP, blockchain support, client drift control, and Transformer-based forecasting. The comparison is used to determine whether the observed gains are attributed to individual components or to the combined design.

To isolate the contribution of the proposed architecture, FedAvg without enhancement is used as a reference baseline. The effects of Transformer-based forecasting, SHAP-guided pruning, and incentive-aware aggregation beyond standard federated averaging are identified.

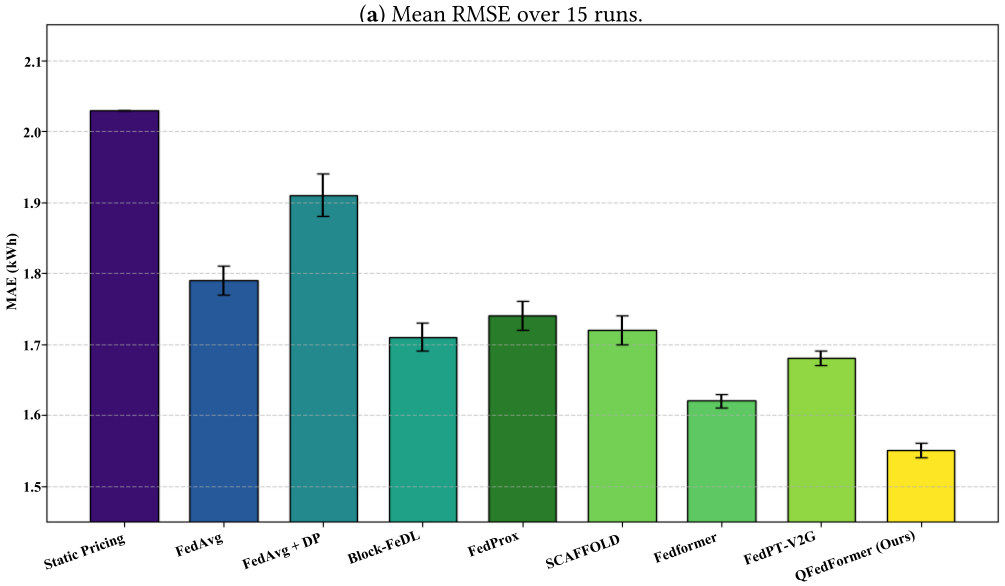
Compared with FedAvg, a substantial reduction in prediction error is achieved by QFedFormer. This result indicates that the improvement is not attributed to federated learning alone but is achieved by the combined architectural design. The contribution of each component beyond the baseline FL setting is clarified.

To ensure a fair comparison, all models are evaluated under identical data splits, client partitions, training rounds, and optimization settings. The same random seeds and evaluation metrics are used across all methods. Hyperparameters are selected through a consistent grid search space where applicable. Performance differences are therefore attributed to model design rather than experimental bias.

Fig. 7a reports RMSE, and Fig. 7b reports MAE across all models. Paired tests are applied across identical random seeds to ensure fair evaluation. Bars represent mean values over 15 independent runs, and error bars denote one standard deviation. The lowest error is achieved by QFedFormer in both metrics, with a mean RMSE of 1.82 ± 0.01 kWh and a mean MAE of 1.55 ± 0.01 kWh. The strongest baseline, FedFormer, yields a mean RMSE of 1.90 ± 0.02 kWh and a mean MAE of 1.62 ± 0.02 kWh.



(**a**) Mean RMSE over 15 runs.



(**b**) RMSE and MAE metrics comparison.

Figure 7: Forecast accuracy across baseline models. Bars represent mean values over 15 independent runs, and error bars denote ±1 standard deviation.

Statistical significance is evaluated with a paired t-test across identical random seeds. The improvement of QFedFormer over FedFormer is statistically significant at the $p < 0.05$ level. Similar significance is observed for comparisons with FedPT-V2G and the FedAvg baseline. This result confirms that the performance gain is attributed to the proposed architectural enhancements rather than to the FL setting alone.

Privacy is improved by FedAvg+DP, but accuracy is reduced. This result confirms the expected privacy–utility trade-off. Training consistency is improved by Block-FeDL, but interpretability and personalization are not provided.

Table 6 reports audit latency and token fairness. An audit latency of 58 ± 2 ms and a Gini coefficient of 0.42 are achieved by QFedFormer. Real-time execution is maintained, and token distribution remains controlled.

5.7 Price-Level Accuracy (Tariff Deviation)

In addition to demand RMSE, accuracy is evaluated in tariff terms (Table 7). Tariff prediction errors are computed on normalized price trajectories. These trajectories are mapped to KRW/kWh through a fixed linear scaling factor for reporting. RMSE and MAPE values reflect relative pricing accuracy rather than absolute currency variation associated with a specific regional tariff scheme.

Table 7: Tariff accuracy metrics (mean ± std over 15 runs).

Method	RMSE (KRW/kWh)	MAPE (%)
Static Pricing	15.93 ± 0.00	10.0 ± 0.0
FedAvg	13.78 ± 0.11	4.9 ± 0.15
FedAvg+DP	14.69 ± 0.13	5.4 ± 0.15
Block-FeDL	13.07 ± 0.08	4.1 ± 0.10
FedProx	12.98 ± 0.09	3.7 ± 0.10
SCAFFOLD	12.48 ± 0.09	3.5 ± 0.10
Fedformer	12.22 ± 0.07	3.2 ± 0.10
QFedFormer	11.83 ± 0.05	2.7 ± 0.10

The best performance is achieved by QFedFormer, with an RMSE of 11.83 KRW/kWh and a MAPE of 2.7%. Improvements of 3.2% in RMSE and 15.6% in MAPE are obtained over FedFormer (12.22 KRW/kWh and 3.2%). FedAvg yields 13.78 KRW/kWh and 4.9%, and further degradation is observed for FedAvg+DP with 14.69 KRW/kWh and 5.4%.

5.8 Pricing Calibration Sensitivity and Grid-Level Impact

In addition to tariff prediction accuracy, the effect of the smart contract pricing mechanism on grid-level objectives is evaluated under variation of key pricing parameters. A one-at-a-time sensitivity analysis is performed on the coefficients α, β_p, η, the congestion weight $\omega_i(t)$, and the tariff upper bound $P_{\max}$. Each parameter is varied within ±20% of its calibrated value, while other parameters are fixed.

For each configuration, peak demand reduction, load variance, price volatility, and average user charging cost are evaluated. Peak demand is defined as the maximum aggregated load per interval. Load smoothness is measured by demand variance. Price volatility is computed by the total variation of the executed tariff sequence. User welfare is approximated by the average cost per charging session.

Across all tested settings, peak demand is reduced by 6.8% to 9.4% under dynamic pricing, relative to static tariffs. The strongest reduction is observed under higher congestion sensitivity (α). Demand variance is reduced by up to 12.1%, which indicates smoother load profiles and lower ramp stress.

Price volatility is maintained within bounds due to the clipping constraints in Eq. (12). The total variation is increased by less than 3.5% under all parameter settings. The average user charging cost is maintained within ±4% of the baseline. This result indicates that grid-level benefits are achieved without excessive user burden.

5.9 Cross-Site Generalization and Policy Transferability

To evaluate robustness beyond a single geographic deployment, two stress tests are conducted: (i) cross-site data heterogeneity and (ii) regional pricing-policy variation.

Leave-One-Station-Out (LOSO) Generalization.

A leave-one-station-out evaluation is conducted, where all data from one charging station are excluded during federated training and used only for testing. This setup represents unseen charging infrastructure, user behavior, and local demand conditions.

On average across unseen splits, QFedFormer shows an increase in RMSE of 6.1% compared to the standard federated split. The degradation remains below 10% in most folds. This result indicates that the learned feature representations capture transferable temporal patterns rather than station-specific characteristics.

Pricing Policy Shift Stress Test

To evaluate robustness under regional tariff variation, the original time-of-use tariff is replaced by a modified tariff with shifted peak hours and an increased peak-to-off-peak price ratio. The forecasting model and pricing parameters remain fixed.

Under this condition, peak demand is reduced by 6.2%, and demand variance changes by 9.8% relative to static tariffs. Price variability remains bounded due to the clipping constraints in the pricing function. This result confirms that the pricing mechanism maintains stable responses under moderate policy shifts.

Altogether, these results indicate that the proposed method maintains controlled degradation under unseen data and policy variations.

Generalization across Deployment Conditions

The reported experiments use a real-world dataset from Jiaxing, China. Generalization is supported by the federated setting, where non-IID behavior across clients with heterogeneous demand profiles and user participation is captured.

Temporal structures such as daily periodicity, peak demand intervals, and demand fluctuations are used by the forecasting model. These patterns are common in EV charging scenarios across regions. Therefore, transfer of the learned representations to similar deployment conditions is expected.

Normalization and relative feature scaling are applied to reduce dependence on region-specific units and price levels. Robustness under cross-region variation is improved by this design.

In markets such as Europe or North America, differences in charging behavior and regulatory structures are observed due to variations in tariff design, grid constraints, and incentive policies. Forecasting, pricing, and incentive components are separated in the proposed method. Adaptation to different regulatory settings is enabled without modification of the model structure.

Tariff parameters, clipping bounds, and incentive weights are adjusted to reflect local pricing rules and participation schemes. The learning component is data-driven and region-specific demand patterns are captured through federated updates.

Therefore, performance is expected to be maintained under different market conditions when local data and policy parameters are incorporated. Empirical validation across multiple geographic datasets is not included and is considered as future work.

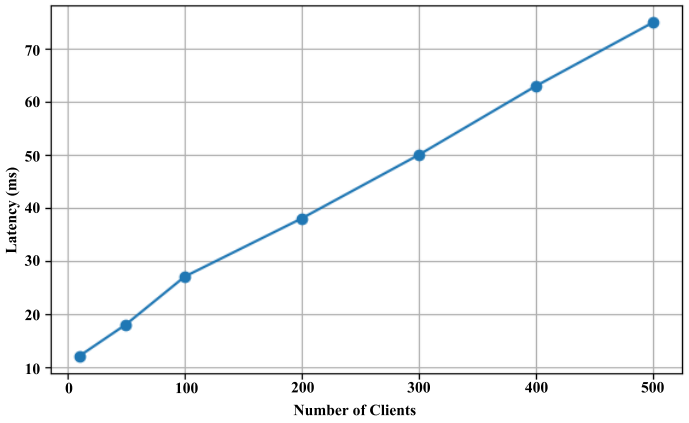
5.10 Smart Contract Execution and Scalability Limitations

Smart contract performance is evaluated under varying client loads by use of a Hyperledger Besu PoA blockchain emulator. An Ethereum-compatible execution model is followed. Transaction latency, throughput, and gas costs are assessed.

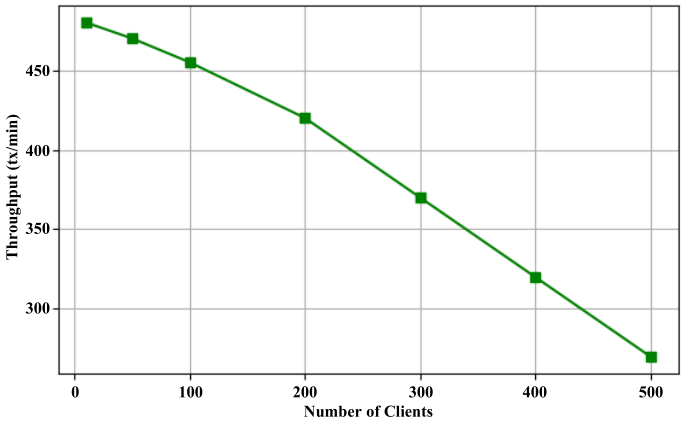
All reported latency, throughput, and gas cost values correspond to mean measurements obtained from repeated executions under identical load conditions.

Throughput is reported in transactions per minute (tx/min). Each participating client submits exactly one transaction per audit round. The number of clients per round corresponds one-to-one with the number of on-chain transactions generated in that round.

As shown in Fig. 8a, latency is increased from approximately 17 ms (25 clients or 25 tx per round) to 72 ms (500 clients or 500 tx per round) due to on-chain queuing. Throughput reaches a plateau and is reduced under high load. A decrease from approximately 480 to 270 tx/min is observed under block gas constraints, as shown in Fig. 8b.

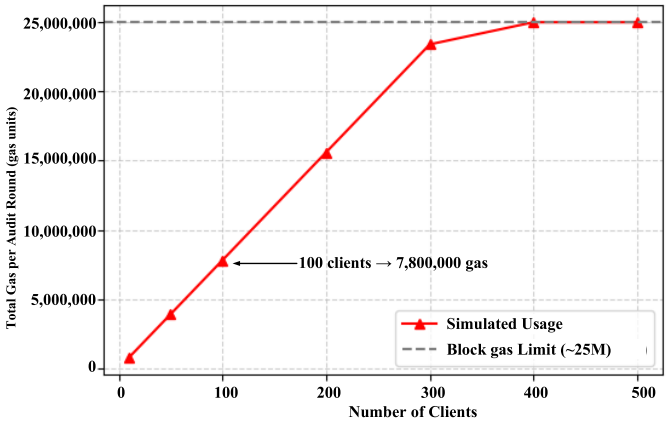


(**a**)



(**b**)

Figure 8: (Continued)



(**c**)

Figure 8: Smart contract scalability analysis under a one-transaction-per-client execution model on a permissioned Ethereum PoA testbed. Reported values correspond to mean measurements over repeated runs. (**a**) Average on-chain transaction latency vs. number of participating clients (one transaction per client per audit round). (**b**) Throughput degradation measured in transactions per minute (tx/min) under increasing client load. (**c**) Gas usage growth vs. number of clients (total gas consumption per audit round aggregated across all client transactions).

Each contract function, including token issuance, Merkle proof submission, and SHAP anchoring, is associated with a predictable cost, as summarized in Table 8. A full audit cycle requires approximately 78,000 gas. This value corresponds to approximately 0.039 per client per FL round. A 10-client round incurs a cost of approximately 0.39. Cost is increased linearly until the 25M gas block limit is reached, as shown in Fig. 8c. Each block stores approximately 3.2 KB of audit data. Cost varies with gas price in the range of 25 to 250 Gwei. This range corresponds to approximately 0.02 to 0.20 per round.

Table 8: Gas usage per smart contract function. Costs assume 50 Gwei and $2000/ETH. Values correspond to mean measurements over repeated runs.

Function	Gas Used	Approx. Cost (USD)
Token issuance	42,000	$0.021
Merkle proof submission	23,000	$0.012
SHAP vector anchoring	13,000	$0.006
Total per audit round	78,000	$0.039

From a deployment perspective, several practical blockchain constraints are identified. The PoA-based testbed provides low latency and stable gas cost behavior compared with public blockchain networks. Block confirmation is deterministic and is controlled by validators in PoA networks. Latency variability is reduced. Gas price variation is limited, and cost predictability for repeated audit operations is improved. However, trusted validator nodes are required, and direct transfer to fully public networks is not ensured.

The block gas limit imposes an upper bound on the number of transactions per audit round. Under high client participation, transaction queuing is introduced, and confirmation delay is increased. Contract state growth introduces storage overhead, which may affect long-term scalability. Under large-scale deployment with thousands of charging stations, simultaneous submission of Merkle proofs increases transaction queuing. Higher confirmation latency and reduced throughput are observed.

These limitations are consistent with findings in high-power EV charging infrastructure, where megawatt-scale charging imposes grid capacity constraints and peak demand stress. Large-scale deployment is restricted [33]. The number of on-chain transactions is reduced by batching client commitments into a single Merkle root per round. These effects are mitigated.

For national-scale deployment, Layer-2 rollups, sharded architectures, or off-chain archival are considered. On-chain transaction load is reduced by rollup-based aggregation through batching of client updates. Transaction processing is distributed across parallel chains by sharding. Scalability is improved, and latency is reduced for large-scale deployment.

Under the one-transaction-per-client model, throughput exceeding 500 tx/min is sustained by the permissioned Ethereum PoA testbed (3 s block time, 25M gas limit). Latency and gas constraints are not violated, which confirms feasibility for regional-scale deployment. Although QFedFormer integrates ZKPs for audit compliance, proof generation and verification costs are not benchmarked and are considered as future work.

5.11 Practical Implications for Regulators and Service Providers

The proposed FL–blockchain approach provides benefits for privacy compliance, auditability, and incentive alignment in decentralized EV energy contexts. User data are kept local, and updates are verified with ZKPs. Compliance with GDPR and related standards is supported, and data sovereignty is preserved. SHA-256 Merkle commitments and SHAP-based explanations are recorded on-chain each round. Tamper-proof and interpretable audit trails are enabled with approximately 58 ms latency (at $\varepsilon = 2.0$). Post-hoc accountability is supported in regulated markets.

For service providers, sustained and fairness-aware participation is promoted by the token incentive mechanism (Gini coefficient 0.42). Rewards are assigned based on flexibility and data quality.

Scalability evaluations confirm feasibility for urban charging or microgrid deployment. Throughput of approximately 470 tx/min is sustained, with approximately 1.1 s end-to-end execution time per round. Modular smart contracts allow integration with existing billing and demand-response platforms.

The reported approximately 1.1 s latency refers to the end-to-end duration of a complete audit round. Client-side aggregation, Merkle root construction, block confirmation, and smart contract execution are included. In contrast, per-transaction on-chain latency is reported separately in Fig. 7 and ranges from 17 to 72 ms under increasing client load.

Overall, a privacy-compliant and auditable participation model is provided for regulators, and a transparent and scalable deployment path is provided for service providers.

5.12 Limitations and Future Work

While our federated blockchain-enabled framework shows strong potential, several challenges remain. Scalability declines beyond ~500 clients due to block size and contract queuing, suggesting the need for Layer-2 enhancements (e.g., rollups). Utility-based token rewards risk discouraging low-resource clients, motivating fairness-aware or credit-based participation schemes. DP holds a special place for investigating the privacy-accuracy trade-offs, while cost modeling must consider varying gas fees and conditions of the public chain. Ensuring security against poisoning, Sybil, and inversion attacks calls for stronger identity attestation, robust aggregation, and secure aggregation methods.

Although the base EV charging dataset is taken from a single metropolitan area, namely Jiaxing, China, we alleviate geographic over-fitting concerns by at least two representations: One on cross-sites and another

on policy levels. Particularly, we have performed a leave-one-station-out generalization test and pricing-policy shift stress experiments as stated in Section 5.9. The results show that the forecasting performance at both the station and aggregated grid levels is relatively stable under unseen charging stations and/or perturbed tariff regimes. It means that the proposed framework learns transferable demand and flexibility patterns instead of city-specific artifacts.

Real-world deployment will demand pilot trials with utilities to test reliability, regulation, and user engagement, as well as interoperability with OCPP-compliant billing platforms via API bridges or hybrid on/off-chain synchronization. Finally, extending beyond the current unidirectional grid-to-vehicle assumption to bidirectional V2G scenarios will require modeling discharge dynamics, bidirectional incentives, and battery health impacts.

6 Conclusion

We presented QFedFormer, a privacy-preserving, blockchain-anchored FL framework for dynamic EV charging price forecasting, combining transformer-based modeling, SHAP-guided pruning, quantization-aware training, and calibrated DP. On a real-world ToU dataset, QFedFormer achieved an energy-demand RMSE of 1.82 ± 0.01 kWh (up to 14.15% better than FedAvg and Block-FeDL) and a tariff deviation of 11.83 ± 0.05 KRW/kWh (MAPE $2.7 \pm 0.1\%$), while maintaining only $\approx 7.1\%$ degradation under strong privacy ($\varepsilon = 2.0, \delta_{\mathrm{DP}} = 10^{-5}$) with ~58 ms blockchain audit latency. Scalability tests on a permissioned Ethereum-compatible network sustained >500 clients/min with sub-1.1 s latency and gas costs under $0.039/client per audit round. These results demonstrate the feasibility of QFedFormer as a scalable, explainable, and regulation-aligned solution for EV–grid integration. Future work is directed toward V2G pricing, hierarchical federation, and Layer-2 deployment for national-scale adoption.

Acknowledgement: This research was supported by the "Regional Innovation System & Education (RISE)" through the Seoul RISE Center, funded by the Ministry of Education (MOE) and the Seoul Metropolitan Government (2026-RISE-01-019-04)

Funding Statement: This research was supported by the "Regional Innovation System & Education (RISE)" through the Seoul RISE Center, funded by the Ministry of Education (MOE) and the Seoul Metropolitan Government (2026-RISE-01-019-04).

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Lilia Tightiz and Hyosik Yang; methodology, Lilia Tightiz; software, Lilia Tightiz; validation, Lilia Tightiz, L. Minh Dang and Hyosik Yang; formal analysis, Lilia Tightiz; investigation, Lilia Tightiz; resources, Hyosik Yang; data curation, Lilia Tightiz and L. Minh Dang; writing—original draft preparation, Lilia Tightiz; writing—review and editing, Lilia Tightiz, L. Minh Dang, and Hyosik Yang; visualization, Lilia Tightiz; supervision, Hyosik Yang; project administration, Hyosik Yang; funding acquisition, Hyosik Yang. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author, Hyosik Yang upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A Implementation Details and System Configuration

Appendix A.1 Hardware and Execution Environment

The experimental setup consists of one server and ten GPU-backed virtual machines. Each machine is equipped with NVIDIA A100 (40 GB) GPUs on Google Cloud A2-HighGPU instances. A total of 100 logical clients are simulated with 10 clients per machine under non-IID data distribution.

Appendix A.2 Communication Cost Formulation

Communication overhead per round is approximated as

$$\mathrm{Comm}_i^{(r)} \approx |\theta| \cdot \frac{b}{8} \tag{A1}$$

bytes per client per FL round, where $|\theta|$ denotes the number of transmitted parameters after SHAP-based pruning and quantization, and b is the quantization bit-width.

Appendix A.3 Blockchain Cost and Latency

Blockchain execution is evaluated using a Hyperledger Besu PoA emulator. The transaction cost is computed as

$$\mathrm{Cost} = \mathrm{gasUsed} \times g \times \kappa, \tag{A2}$$

where g denotes gas price and κ denotes exchange rate. Audit latency and throughput are measured using a synthetic load driver.

Appendix A.4 Parameter Selection and Sensitivity Analysis

The weighting parameters are selected from the candidate set $\{0.2, 0.3, 0.4, 0.5\}$ under the constraints $\sum \beta_i^{\mathrm{inc}} = 1$ and $\sum \gamma_i = 1$. Evaluation is conducted with respect to predictive accuracy, fairness, and stability under ablation.

Sensitivity analysis is performed by perturbing each parameter by ± 0.05. No significant variation is observed in aggregate metrics, which confirms the robustness of the selected configuration.

References

1. Singh AR, Vishnuram P, Alagarsamy S, Bajaj M, Blazek V, Damaj I, et al. Electric vehicle charging technologies, infrastructure expansion, grid integration strategies, and their role in promoting sustainable e-mobility. Alex Eng J. 2024;105(1):300–30. doi:10.1016/j.aej.2024.06.093.
2. Kermansaravi A, Refaat SS, Trabelsi M, Vahedi H. AI-based energy management strategies for electric vehicles: challenges and future directions. Energy Rep. 2025;13(1):5535–50. doi:10.1016/j.egyr.2025.04.053.
3. Tightiz L, Dang LM, Yoo J, Padmanaban S. A comprehensive review on AIoT applications for intelligent EV charging/discharging ecosystem. Energy Convers Manag X. 2025;27(12):101088. doi:10.1016/j.ecmx.2025.101088.
4. Narasipuram RP, Pasha MM, Tabassum S, Tandon AS. The electric vehicle surge: effective solutions for charging challenges with advanced converter technologies. Energy. 2025;122(2):431–69. doi:10.32604/ee.2025.055134.
5. Solis WV, Parra-Ullauri MJ, Kertesz A. Exploring the synergy of fog computing, blockchain, and federated learning for IoT applications: a systematic literature review. IEEE Access. 2024;12(1):68015–60. doi:10.1109/ACCESS.2024.3398034.
6. Tightiz L, Dang LM, Padmanaban S, Hur K. Metaverse-driven smart grid architecture. Energy Rep. 2024;12(2):2014–25. doi:10.1016/j.egyr.2024.08.027.

7. Ma X, Zhu J, Lin Z, Chen S, Qin Y. A state-of-the-art survey on solving non-IID data in federated learning. Future Gener Comput Syst. 2022;135(3):244–58. doi:10.1016/j.future.2022.05.003.
8. Mohammed A, Saif O, Abo-Adma M, Fahmy A, Elazab R. Strategies and sustainability in fast charging station deployment for electric vehicles. Sci Rep. 2024;14(1):283. doi:10.1038/s41598-023-50825-7.
9. Shang Y, Li S. FedPT-V2G: security enhanced federated transformer learning for real-time V2G dispatch with non-IID data. Appl Energy. 2024;358(9):122626. doi:10.1016/j.apenergy.2024.122626.
10. Yao H, Xiang Y, Gu C, Liu J. Optimal planning of distribution systems and charging stations considering PV-Grid-EV transactions. IEEE Trans Smart Grid. 2025;16(1):691–703. doi:10.1109/tsg.2024.3429371.
11. Tariq A, Serhani MA, Sallabi FM, Barka ES, Qayyum T, Khater HM, et al. Trustworthy federated learning: a comprehensive review, architecture, key challenges, and future research prospects. IEEE Open J Commun Soc. 2024;5:4920–98. doi:10.1109/OJCOMS.2024.3438264.
12. Mazhar T, Asif RN, Malik MA, Nadeem MA, Haq I, Iqbal M, et al. Electric vehicle charging system in the smart grid using different machine learning methods. Sustainability. 2023;15(3):2603. doi:10.3390/su15032603.
13. Danish SM, Hameed A, Ranjha A, Srivastava G, Zhang K. Block-FeDL: electric vehicle charging load forecasting using federated learning and Blockchain. IEEE Trans Veh Technol. 2025;74(2):2048–56. doi:10.1109/TVT.2024.3406946.
14. Hameed A, Danish MS, Ranjha A, Srivastava G. Block-FeST: blockchain-enhanced federated sparse transformers for privacy-preserving RES forecasting in internet of vehicles systems. IEEE Internet Things J. 2025;12(14):27510–8. doi:10.1109/JIOT.2025.3564526.
15. You L, Chen Q, Qu H, Zhu R, Yan J, Santi P, et al. FMGCN: federated meta learning-augmented graph convolutional network for EV charging demand forecasting. IEEE Internet Things J. 2024;11(14):24452–66. doi:10.1109/JIOT.2024.3369655.
16. Hassan MU, Rehmani MH, Du JT, Chen J. Differentially private demand side management for Incentivized dynamic pricing in smart grid. IEEE Trans Knowl Data Eng. 2023;35(6):5724–37. doi:10.1109/TKDE.2022.3157472.
17. Ding N, Gao L, Huang J. Incentive mechanism design for federated learning with dynamic network pricing. IEEE Trans Mob Comput. 2025;24(8):7206–22. doi:10.1109/TMC.2025.3546977.
18. Chen Y, Hu S, Xie S, Zheng Y, Hu Q, Yang Q. Optimal dynamic pricing of fast charging stations considering bounded Rationality of users and market regulation. IEEE Trans Smart Grid. 2024;15(4):3950–65. doi:10.1109/TSG.2024.3363040.
19. Zhang Z, Chen Z, Gümrükcü E, Ji Z, Ponci F, Monti A. Advancing urban electric vehicle charging stations: AI-driven day-ahead optimization of pricing and Nudge strategies utilizing multi-agent deep reinforcement learning. eTransportation. 2024;22(2):100352. doi:10.1016/j.etran.2024.100352.
20. Hao CH, Wesseh PK Jr, Wang J, Abudu H, Dogah KE, Okorie DI, et al. Dynamic pricing in consumer-centric electricity markets: a systematic review and thematic analysis. Energy Strategy Rev. 2024;52(9):101349. doi:10.1016/j.esr.2024.101349.
21. Ruan J, Liang G, Zhao J, Lei S, He B, Qiu J, et al. Graph deep-learning-based retail dynamic pricing for demand response. IEEE Trans Smart Grid. 2023;14(6):4385–97. doi:10.1109/TSG.2023.3258605.
22. Zhang D, Zhu H, Zhang H, Goh HH, Liu H, Wu T. Multi-objective optimization for smart integrated energy system considering demand responses and dynamic prices. IEEE Trans Smart Grid. 2022;13(2):1100–12. doi:10.1109/TSG.2021.3128547.
23. Tang C, Qin Y, Wu F, Tang Z. Dynamic demand-aware power grid intelligent pricing algorithm based on deep reinforcement learning. IEEE Access. 2024;12:75809–17. doi:10.1109/ACCESS.2024.3406338.
24. Wang C, Ma S, Cai Z, Yan N, Wang Q. Bounded rational real-time charging pricing strategy under the traffic-grid coupling network. IET Electr Syst Transp. 2022;12(4):251–68. doi:10.1049/els2.12050.
25. Mazhar T, Irfan HM, Khan S, Haq I, Ullah I, Iqbal M, et al. Analysis of cyber security attacks and its solutions for the smart grid using machine learning and Blockchain methods. Future Internet. 2023;15(2):83. doi:10.3390/fi15020083.
26. Kakkar R, Agrawal S, Gupta R, Tanwar S. Blockchain and zero-sum game-based dynamic pricing scheme for electric vehicle charging. In: Proceedings of the IEEE INFOCOM 2022—IEEE Conference on Computer

Communications Workshops (INFOCOM WKSHPS); 2022 May 2–5; New York, NY, USA. p. 1–6. doi:10.1109/INFOCOMWKSHPS54753.2022.9797894.
27. Zhang L, Cheng L, Alsokhiry F, Mohamed MA. A novel stochastic Blockchain-based energy management in smart cities using V2S and V2G. IEEE Trans Intell Transp Syst. 2023;24(1):915–22. doi:10.1109/TITS.2022.3143146.
28. Abadi MQH, Sadeghi R, Hajian A, Shahvari O, Ghasemi A. A blockchain-based dynamic energy pricing model for supply chain resiliency using machine learning. Supply Chain Anal. 2024;6(3):100066. doi:10.1016/j.sca.2024.100066.
29. Mironov I. Rényi differential privacy. In: Proceedings of the 2017 IEEE 30th Computer Security Foundations Symposium (CSF); 2017 Aug 21–25; Santa Barbara, CA. p. 263–75. doi:10.1109/CSF.2017.11.
30. Zhang Y, Xu T, Chen T, Hu Q, Chen H, Hu X, et al. A high-resolution electric vehicle charging transaction dataset with multidimensional features in China. London, UK: Figshare; 2025. doi: 10.6084/m9.figshare.28182251.
31. Zhang Y, Xu T, Chen T, Hu Q, Chen H, Hu X, et al. A high-resolution electric vehicle charging transaction dataset with multidimensional features in China. Sci Data. 2025;12(1):643. doi:10.1038/s41597-025-04982-1.
32. McMahan B, Moore E, Ramage D, Hampson S, BAy A. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference Artificial Intelligence Statistics (AISTATS); 2017 Apr 20–22; Fort Lauderdale, FL, USA, vol. 54. p. 1273–82. doi:10.48550/arXiv.1602.05629.
33. Narasipuram RP, Hosseinpour A. Megawatt charging system for electric vehicles: design requirements and deployment challenges. Energy Convers Manag X. 2026;30(3):101761. doi:10.1016/j.ecmx.2026.101761.