Secure and Privacy-Preserving Load Forecasting in Smart Grids Through Blockchain-Based Incentives

Joan Ferré-Queralt[1 [0009−0002−0014−0065]], Jordi Castellà-Roca[1 [0000−0002−0037−9888]], and Alexandre Viejo[1 [0000−0003−2342−5100]]

Universitat Rovira i Virgili
Departament d'Enginyeria Informàtica i Matemàtiques
CYBERCAT-Center for Cybersecurity Research of Catalonia
Av. Països Catalans 26, E-43007 Tarragona, Catalonia, Spain
{joan.ferreq,jordi.castella,alexandre.viejo}@urv.cat

Abstract. Modern smart grids increasingly rely on accurate load forecasting to manage energy resources efficiently amid the growing integration of renewable and distributed energy sources. While federated learning offers a promising approach by enabling decentralized model training without raw data sharing, it still exposes users to privacy risks through model updates and remains vulnerable to malicious participants. This paper presents a secure and privacy-preserving federated learning framework tailored for smart grid load forecasting. The proposed system enhances privacy by employing anonymous update submissions, zero-knowledge proofs, and blind signatures, thereby unlinking user identities from their model contributions. To ensure model integrity, a blockchain-based reputation mechanism is introduced, incentivizing honest participation while penalizing malicious behavior without unfairly targeting legitimate users. The system also incorporates a distributed clustering process to address data heterogeneity, enabling the training of specialized models across user clusters. The framework is evaluated in terms of privacy, security, and performance, showing that it preserves user confidentiality, ensures model integrity, and operates efficiently on resource-constrained IoT devices.

Keywords: Smart Grid · Load Forecasting · Privacy · Security · Federated Learning · Blockchain · Zero-Knowledge Proofs

1 Introduction

Accurate load forecasting is a critical component of modern smart grids, especially as global electricity demand continues to rise due to urbanization, the proliferation of electronic devices, and population growth [1]. Managing energy resources efficiently becomes increasingly challenging with the shift towards renewable and distributed energy resources (DERs), such as solar panels, wind turbines, and local storage systems. While DERs help reduce reliance on fossil

fuels and mitigate environmental impacts [24], their inherent variability makes predicting energy supply and demand more complex. This unpredictability highlights the importance of effective load forecasting for ensuring smooth grid operation and reliable integration of these renewable sources.

To address these challenges, load forecasting provides grid operators with the ability to anticipate fluctuations in demand and supply, enabling efficient resource management and improved integration of renewable sources [2]. Advanced Metering Infrastructure (AMI) supports these forecasting efforts by collecting detailed, real-time energy consumption data from consumers [21]. By applying advanced techniques such as statistical methods [3], support vector machines [7, 27], random forest regressors [8, 15], and deep learning models like Long Short-Term Memory (LSTM) networks [2], significant improvements in forecasting accuracy have been achieved.

However, the collection of detailed consumption data raises serious privacy and security concerns. Training forecasting models relies on accessing consumers' load data [14], which can inadvertently reveal sensitive information about their habits and lifestyles. Detailed energy consumption patterns collected by AMI can reveal when residents are home or away, their daily routines, work schedules, and even specific activities such as cooking or entertainment. If unauthorized parties access this granular data, it can be exploited for harmful purposes, including targeted marketing, surveillance, or criminal activities [4].

These privacy risks highlight the importance of developing developing privacy-preserving solutions that protect user data throughout the forecasting process. Achieving this requires innovative approaches that encourage user participation while ensuring data confidentiality and security. Blockchain technology, known for its decentralized, transparent, and secure nature, offers a promising solution by providing a trustworthy environment for data exchange [19]. By leveraging these strengths, blockchain-based incentive mechanisms can enhance participation by rewarding users who share data safely and honestly, while simultaneously discouraging harmful actions. This approach not only safeguards user privacy but also fosters greater trust and collaboration within smart grid systems.

1.1 Related work

Soykan et al. [22] took an early step towards reducing privacy leakage in load forecasting by applying differential privacy to LSTM models. They add noise to the input data to hide individual data points. While this approach provides privacy, it relies on centralized training, where data is collected and processed by a trusted third party. This dependency creates vulnerabilities, as the central aggregator could be targeted by attacks or misuse the data.

A promising alternative to centralized learning is federated learning, which allows models to be trained collaboratively across distributed devices without requiring raw data to be shared with a central entity. By keeping data local and only sharing model updates, this approach inherently provides stronger privacy guarantees, significantly reducing the risk of privacy leakage. One of the first implementations of federated learning in smart grid load forecasting was proposed

by [23], who demonstrated that sharing only model updates with the aggregator effectively preserves user's raw data privacy.

Several other studies have explored the use of federated learning for load forecasting. For instance, [9] compares two federated learning methods, FedSGD and FedAVG, with FedAVG [17] demonstrating superior performance in smart grid load prediction. While these methods prevent direct transmission of raw data to the aggregator, private information can still be inferred from model updates through certain techniques [18].

Another example is presented by [28], where federated learning is combined with differential privacy by applying noise to updated parameters to prevent privacy leakage. Although this approach enhances privacy, it comes at the cost of reduced accuracy, as the added noise degrades the quality of the parameter updates used to train the model. Additionally, the authors do not address the heterogeneity of data from different users, which further reduces the model's overall accuracy.

He et al. [11] address data heterogeneity by applying k-means clustering to group users with similar load profiles before training separate models using federated learning. This approach aims to improve accuracy by tailoring models to specific user clusters. However, to perform clustering, users must share their fine-grained consumption data with the aggregator, posing significant privacy risks when dealing with an honest-but-curious aggregator.

To mitigate the aforementioned privacy risks, [20] proposes sharing only average load data with the aggregator during the k-means clustering process to enhance privacy. However, even average load profiles can still reveal meaningful information about daily electricity consumption patterns. Building on this approach, [25] further improves privacy by sharing only household properties instead of detailed load data during clustering. While this method offers better protection, it still requires trusting the aggregator and involves sharing some private information.

To address the privacy concerns associated with sharing data with a central aggregator for clustering, recent studies have explored distributed k-means clustering, where each user performs training locally and only sends model updates to the aggregator. For example, [26] proposes an aggregator-assisted distributed k-means clustering scheme that reduces communication overhead but does not account for outliers and considers only a single load profile per user. To overcome these limitations, [12] introduces a hierarchical federated framework that combines distributed k-means clustering with an aggregator, effectively handling outliers and supporting multiple load profiles per user.

Despite advancements in federated learning, users must still share individual parameter updates during training, which can expose patterns in their load data to an honest-but-curious aggregator. To address this issue, [16] employs homomorphic encryption and multiparty computation (MPC) to perform distributed k-means clustering and federated learning without revealing individual parameter updates. Although this approach effectively prevents raw data exposure, it introduces a significant drawback: encrypted updates cannot be traced back to

individual users. As noted in [6], this limitation not only increases computational overhead but also prevents the application of essential security measures, such as the Krum aggregation method [5], which detects and mitigates model poisoning attacks by identifying malicious users. Without the ability to inspect encrypted updates, detecting or attributing malicious activity remains a critical challenge.

Finally, to address the limitations of homomorphic encryption, [13] propose a fragmented federated learning scheme where each participant exchanges encrypted parameter fragments, rather than full model updates, with another user to protect privacy. This method assumes data homogeneity and uses a two-level reputation system (local and global) to filter out malicious participants by favoring those with higher scores. However, an attacker can exploit this system by exchanging bad fragments with honest users, leading the global layer to wrongly flag them as malicious. Although the local layer can eventually identify and isolate the attacker, the approach does not apply for a smart grid system with a large quantity of users: a single adversary can harm many honest users before being blocked. In addition, the global reputation ledger is managed centrally by the aggregator, which concentrates trust and creates the risk of unfair behavior, such as withholding rewards from honest participants.

1.2 Contributions and plan of this paper

The main contribution of this work is the integration of a reputation-driven mechanism into federated learning, specifically tailored for smart grid environments. By addressing the unique challenges of load forecasting, such as data heterogeneity, privacy risks, and the need for trustworthy collaboration, this approach significantly enhances participant privacy through smart contracts, zero-knowledge proofs, and blind signatures; strengthens user trust by avoiding penalties for honest participants; and improves overall system security.

The remainder of this paper is organized as follows. In Section 2, an overview of the proposed system is presented, describing the key requirements, defining the actors involved, and introducing the basic definitions and concepts necessary for understanding the approach. In Section 3, the proposed framework is detailed, followed by its evaluation in Section 4. Finally, Section 5 concludes the article by summarizing the contributions and discussing potential directions for future research.

2 System overview

This section defines the actors and components involved, outlines the key system requirements, and presents a summary of the protocol steps.

2.1 Actors and components

The main actors of the proposed system are the *prosumers* (U_i), who are the end-users possessing private load data (X_i), which contain several electricity

patterns. An electricity pattern E_d^i represents the daily load profile of user U_i for day d, consisting of load values at each time slot:

$$E_d^i = \left[E_{d,1}^i \; E_{d,2}^i \; \dots \; E_{d,T}^i\right].$$

Prosumers can train a local model using this data, thereby preventing its disclosure to third parties. This data is sensitive, as it can reveal behavioral patterns. Additionally, users aim to prevent any linkage between their identity and the parameter updates they submit.

The objective of the system is to train a general load forecasting model using data from prosumers. The model $\mathcal{M}(E_{d-1}^i, \theta)$ is defined as a function parameterized by a set θ, used to predict the next day's load $Y_{i,d}$ given the previous day's load E_{d-1}^i as input:

$$Y_{i,d} = \mathcal{M}(E_{d-1}^i, \theta) = \left[y_1 \; y_2 \; \dots \; y_T\right].$$

The model parameters θ are optimized by minimizing the mean squared error (MSE):

$$\min_{\theta} \; MSE(\theta) = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{2}(y_t - E_{d-1,t}^i)^2.$$

In the federated learning setup, each user U_i maintains a local model $\mathcal{M}^i$ with parameters θ^i. These local models are updated using gradient descent, where α is the learning rate:

$$\theta^i \leftarrow \theta^i - \alpha \nabla MSE(\theta^i)$$

This process is iterative, consisting of local updates followed by global aggregation using the FedAVG algorithm [17], which computes the global model $\mathcal{M}$ by averaging the parameters of the local models, where N denotes the number of users:

$$\theta = \frac{1}{N}\sum_{i=1}^{N}\theta^i,$$

To improve model accuracy and enhance data homogeneity, a *General Aggregator* (A_G) is employed to supervise the execution of a distributed clustering algorithm across the system, resulting in multiple groups of users with similar data characteristics. Each group is then assigned a *Cluster Aggregator* (A_C) responsible for managing the federated learning process.

As part of the framework's security and privacy guarantees, a reputation system is incorporated through the deployment of the following smart contracts:

- *Reputation Smart Contract* ($\mathcal{RSC}$): Maintains a mapping between users and their reputation scores.
- *Membership Smart Contract* ($\mathcal{MSC}_l$): Stores the set of users participating in training round l, manages reward distribution, and updates user reputations.

– *Bulletin Board Smart Contract* ($\mathcal{BBSC}$): Serves as a public interface where the service provider publishes rewards.

2.2 System requirements

The proposed federated learning framework for load forecasting shall satisfy the following enhanced requirements to ensure robust, secure, and privacy-preserving model training in smart grid environments based on IoT devices:

R1: *Data Heterogeneity Management*: The system must support diverse user load profiles by enabling effective clustering and personalized model training.

R2: *Dynamic Reputation Management*: The system must include a reputation mechanism that continuously evaluates participant credibility and uses immutable scores to guide participant selection, helping to mitigate the risks of malicious or unreliable contributions.

R3: *Secure Cluster Aggregator Communications*: The cluster aggregator must deliver model updates through secure, authenticated channels, ensuring correctness and preventing tampering or unauthorized access.

R4: *Independent Anonymous Communication Channel*: The system must support anonymous channels that decouple updates from user identities, ensuring unlinkability and preserving anonymity during transmission.

R5: *Private Participant Authentication*: All participants must be authenticated within the federated learning protocol without revealing their identities.

R6: *Malicious Update Detection*: The cluster aggregator must detect anomalies or poisoning attempts in user updates while ensuring that only the aggregator can access the update parameters.

R7: *Incentives*: The system should reward honest participants with incentives and increased reputation, while discouraging malicious behavior.

R8: *Reward Unforgeability*: It must be ensured that rewards cannot be forged or obtained without legitimate participation.

R9: *Double-spending Rewards*: Each reward must be redeemable only once per participation round, preventing double-spending.

R10: *IoT Deployment Feasibility*: The framework should be optimized for deployment on IoT devices, such as smart meters with constrained computational resources.

2.3 Federated learning protocol

After the distributed clustering phase conducted by the general aggregator (A_G), the proposed protocol proceeds through the steps illustrated in Figure 1, involving the cluster aggregators (A_C) and the users (U_i):

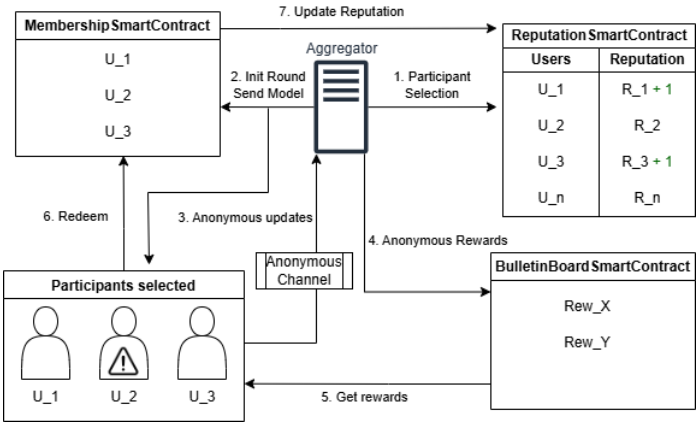


Fig. 1. An overview of the proposed protocol.

1. In each round, A_C selects a subset of users based on their reputation scores recorded in $\mathcal{RSC}$.
2. A_C initializes a smart contract $\mathcal{MSC}$ to register the identities of the selected users and sends the model to them.
3. Each selected U_i performs local model training and transmits the parameter updates via an anonymous channel, along with an anonymous token to be used for claiming a reward. A proof of participation in the round is also included.
4. A_C processes and verifies the received updates and proofs to train the global model. If verification is successful, the token is signed and uploaded to the $\mathcal{BBSC}$.
5. Each U_i who has been verified as honest and rewarded by A_C can retrieve his or her signed token from the $\mathcal{BBSC}$.
6. Each honest U_i retrieves the signed token and redeems the reward through the corresponding $\mathcal{MSC}_l$, following a secondary verification protocol.
7. Finally, the reputation of each rewarded U_i is updated accordingly in the $\mathcal{RSC}$.

3 Our Proposal

This section presents the proposed system in detail, describing each stage of the protocol and its corresponding procedures.

3.1 Step-1: Distributed clustering

The proposed system adopts the distributed clustering approach introduced in [12] to group users based on their electricity consumption patterns without sharing raw data. This method is summarized below for completeness.

1. Initially, A_G selects the centroids $P^0 = \{P_1^0, \ldots, P_K^0\}$ and broadcast them to all U_i.
2. Next, for each iteration $l \in \{1, \ldots, L\}$:
   (a) Each U_i assigns each daily pattern E_d^i to its closest centroid based on the following minimization:
   $$c_d = \arg\min_k D(E_d^i, P_k^l).$$
   (b) Each U_i computes the local mean $\mu_k^{i,l}$ for each cluster k, and calculates the update:
   $$\Delta_k^{i,l} = n_k^{i,l}(\mu_k^{i,l} - P_k^l),$$
   where $n_k^{i,l}$ denotes the number of patterns assigned to cluster k.
   (c) Each U_i sends $\Delta_k^{i,l}$ and $n_k^{i,l}$ to A_G.
   (d) A_G then updates each centroid:
   $$P_k^{l+1} = P_k^l + \frac{\sum_i \Delta_k^{i,l}}{\sum_i n_k^{i,l}}.$$
   and broadcasts the updated centroids P^{l+1} to all U_i.
   (e) If convergence is detected (i.e., centroids no longer change significantly), the iteration stops. The final centroids are then denoted by P^L.
3. Upon convergence, each U_i computes the average of his or her daily patterns,
   $$\bar{E}^i = \frac{1}{D}\sum_{d=1}^{D} E_d^i,$$
   and assigns to the closest final centroid:
   $$c^i = \arg\min_k D(\bar{E}^i, P_k^L).$$

3.2 Step-2: System setup

Following the clustering phase described in the previous step, the system initializes the necessary components for federated learning. This includes deploying smart contracts, generating key pairs, authenticating users, and registering their identifiers and reputation scores with the assigned cluster aggregator (A_C). This process unfolds as follows:

1. Each A_C deploys its corresponding smart contracts $\mathcal{RSC}$ and $\mathcal{BBSC}$.
2. Each U_i generates an asymmetric key pair (sk_i, pk_i), where sk_i is a securely generated random value and $pk_i = \mathcal{H}(sk_i)$. The public key pk_i serves as U_i's identifier.
3. Each U_i authenticates to its corresponding A_C using a verifiable credential (this process is out of the scope of this work) and provides the necessary information required to participate in the federated learning process.
4. The A_C registers each public key pk_i (only one per registered U_i) in its $\mathcal{RSC}$ and initializes U_i's reputation score to zero.

3.3 Step-3: Reputation-based federated learning

This step focuses on collaboratively training load forecasting models in a privacy-preserving and secure manner, leveraging the reputation-based mechanisms established during system initialization.

The protocol runs over multiple training rounds, each consisting of a sequence of subprotocols executed in a fixed order. The number of rounds can be set in advance or determined dynamically based on convergence criteria. The subprotocols executed in each training round l are discussed next:

Step-3.1: Round initialization. At the beginning of each training round l, the cluster aggregator A_C prepares the round by selecting participants, setting up smart contracts, and securely distributing the current global model. This process is carried out as follows:

1. A_C generates an EdDSA key pair to be used in the blinding signing protocol for anonymous user rewarding:

$$(sk^l_{A_C} = x, \quad PK^l_{A_C} = x \cdot G \mod N).$$

2. A_C selects a subset of users for round l based on their reputation scores and prior participation frequency. Let r_i denote the reputation score of user U_i stored in $\mathcal{RSC}$, and let f_i be the number of times U_i has been selected in previous rounds. Each user is assigned a trust value:

$$w_i = \frac{r_i + 1}{f_i + 1},$$

   and the selection probability for user U_i is computed as:

$$p_i = \frac{w_i}{\sum_{j=1}^{|U|} w_j}.$$

3. A_C randomly selects n users (without replacement) according to these probabilities, forming the participant subset ψ_l for the current round.
4. A_C inserts the public keys pk_i of the selected users in ψ_l into a Merkle tree and records its root in a newly instantiated smart contract $\mathcal{MSC}_l$, alongside the public key $PK^l_{A_C}$ for the current round.
5. A_C generates n random nonces k_i and computes the corresponding blinding points $P'_i = k_i \cdot G$ for use in the blind signature protocol. Each pair (k_i, P'_i) is stored locally with k_i kept private, and A_C uploads all P'_i values to $\mathcal{MSC}_l$.
6. A_C sends the global model parameters θ to each selected user U_i via a secure and private communication channel. A_C must authenticate itself and ensure the integrity and authenticity of θ.

Step-3.2: Anonymous updates. Each selected user U_i receives the global model parameters θ and trains locally on dataset X_i, obtaining updated parameters θ^i. To submit θ^i anonymously and verifiably, U_i engages in a cryptographic protocol with the aggregator. The process is as follows:

1. U_i generates a hashclaim $HC_i = \mathcal{H}(PK^l_{A_C}, sk_i)$ and constructs a non-interactive zero-knowledge proof π_i^P verifying that:
   - Knowledge of sk_i such that $pk_i = \mathcal{H}(sk_i)$ is included in the Merkle tree stored in $\mathcal{MSC}_l$.
   - Correct construction of HC_i using $PK^l_{A_C}$ and sk_i as pre-images.
2. U_i submits a transaction to the smart contract $\mathcal{MSC}_l$ from an ephemeral anonymous account, including HC_i. In response, $\mathcal{MSC}_l$ verifies π_i^P and, if valid, returns the first unclaimed point, marks it as claimed, and associates it with HC_i. This transaction requires a deposit, which is held by the contract to discourage dishonest participation. From this step onward, the claimed point for user U_i is denoted as P'_i.
3. U_i generates a random blinding factor pair (a, b) and computes:

$$P_i = P'_i + (a \cdot G) + (b \cdot PK^l_{A_C}).$$

4. U_i computes a token t and blinds it to obtain t' as follows:

$$t = \mathcal{H}(P_i,\, PK^l_{A_C},\, pk_i), \quad t' = t + b.$$

5. U_i computes a non-interactive zero-knowledge proof π_i^T to demonstrate knowledge of the secret key sk_i satisfying the following conditions:
   - $pk_i = \mathcal{H}(sk_i)$.
   - $t' = \mathcal{H}(P_i,\, PK^l_{A_C},\, pk_i) + b$.
   - pk_i is included in the Merkle tree stored in $\mathcal{MSC}_l$.
   - $HC_i = \mathcal{H}(PK^l_{A_C}, sk_i)$.

   The proof π_i^T confirms that the user is a legitimate participant in the round (by virtue of membership in the Merkle tree) and that the token t corresponds to pk_i, for which the user possesses sk_i. The proof reveals none of the values sk_i, pk_i, t, P_i, or b. Its public output consists solely of t' and the Merkle tree root, which are used by A_C for verification. The proof also demonstrates possession of the hash claim HC_i.
6. U_i encrypts the local model update θ^i using ECIES [10] with $PK^l_{A_C}$ to obtain $c_i = \mathrm{Enc}(\theta^i)$.
7. Finally, U_i sends (c_i, t', HC_i, π_i^T) to A_C, transmitted via an anonymous channel. This channel must preserve the anonymity of the sender U_i; however, specific implementation details are beyond the scope of this work.

Step-3.3: Global model update and reward distribution. After collecting all messages (c_i, t', HC_i, π_i^T) from the selected users U_i, the aggregator A_C verifies their validity, filters out malicious contributions, and aggregates the valid updates into a new global model. The protocol proceeds as follows:

1. A_C decrypts each received c_i using its private key to recover the model update θ^i.
2. A_C verifies the zero-knowledge proof π_i^T, ensuring that its public outputs are consistent with t_i' and the Merkle tree root stored in $\mathcal{MSC}_l$. It also checks that the claimed point P_i' is correctly linked to the hash claim HC_i. Messages failing these checks are discarded.
3. For the remaining valid submissions, A_C analyzes the model updates to detect potentially malicious behavior, using methods such as the previously mentioned Krum aggregation [5]. Updates identified as malicious are excluded from aggregation and reward eligibility. A_C then applies the Federated Averaging (FedAvg) algorithm [17] to the valid updates to compute the new global model parameters θ.
4. For each user U_i whose update was accepted and deemed honest, A_C signs the corresponding blinded token using EdDSA by computing:

$$\sigma_i' = t_i' \cdot x + k_i,$$

where x is A_C's EdDSA private key corresponding to $PK_{A_C}^l$, and k_i is the private nonce used to generate P_i'. A_C then publishes (t_i', σ_i') to $\mathcal{BBSC}$.

Step-3.4: Token unblinding and redemption. Once the blinded signature σ' is published on $\mathcal{BBSC}$, each U_i unblinds the signature and redeems the corresponding token. The procedure unfolds as follows:

1. U_i retrieves σ' from $\mathcal{BBSC}$ and uses the blinding factor b, selected during token generation, to unblind the signature:

$$\sigma = \sigma' + b.$$

as a result, the tuple (P_i, σ, t) forms a valid signature corresponding to the public key pk_i.
2. U_i generates a zero-knowledge proof π_i^R to enable signature verification without revealing sensitive information. This proof demonstrates that, given:

$$s = \mathcal{H}(P_i,\, PK_{A_C}^l,\, pk_i),$$

the following verification equation holds:

$$\sigma \cdot G = P_i + (PK_{A_C}^l \cdot s).$$

The public inputs to the proof are P_i, $PK_{A_C}^l$, pk_i, and σ. U_i submits these public values, along with the proof π_i^R, in a transaction to $\mathcal{MSC}_l$.
3. Upon receiving the transaction, $\mathcal{MSC}_l$ verifies the following:
   (a) pk_i is registered within $\mathcal{MSC}_l$.
   (b) pk_i has not previously been used to redeem a token.
   (c) σ is valid, as confirmed by the zero-knowledge proof π_i^R.

4. Upon successful verification, $\mathcal{MSC}_l$ notifies $\mathcal{RDC}$, which marks pk_i as redeemed and updates its reputation in $\mathcal{RSC}$. Additionally, $\mathcal{MSC}_l$ refunds the amount originally paid by the user for the point. If malicious users participated in the round, they do not receive any reward, and their funds are withheld and redistributed among honest participants, further incentivizing trustworthy behavior.

4 Evaluation

This section evaluates the privacy, security, and practical feasibility of the proposed framework with respect to the requirements outlined in Section 2.2. The evaluation is structured around 12 claims, each supported by a corresponding proof sketch that provides the underlying security argument.

Claim 1 *The system is capable of handling heterogeneous load data by partitioning users into homogeneous clusters, thereby enabling the training of specialized forecasting models with improved accuracy. (R1)*

Proof sketch. The system addresses data heterogeneity through a distributed clustering algorithm (see Section 3.1). Each user U_i locally computes its average daily load profile, and is then grouped with other users exhibiting similar profiles. This process reduces intra-cluster variance, and leads to more homogeneous data distributions within each cluster. As a result, training a dedicated forecasting model per cluster improves convergence and reduces prediction error. □

Claim 2 *The system incorporates a robust reputation management mechanism that assigns and updates reputation scores exclusively upon verified participation, ensuring these scores are stored immutably and resistant to tampering. (R2)*

Proof sketch. Reputation management is handled by a dedicated Reputation Smart Contract (i.e., $\mathcal{RSC}$) deployed on a secure blockchain. Each user's reputation score r_i is stored in this immutable ledger, which relies on cryptographic primitives such as collision-resistant hash functions and unforgeable digital signatures to ensure integrity. Scores are updated only after a participant successfully redeems a valid, signed token via the Membership Smart Contract (i.e., $\mathcal{MSC}_l$). Any attempt to alter reputation scores without valid participation would require breaking the blockchain's cryptographic guarantees, which is computationally infeasible. Therefore, the system ensures that reputation updates are tamper-resistant and based solely on verifiable activity. □

Claim 3 *The distribution of the global model in each federated learning round is conducted securely and privately, ensuring that the aggregator is authenticated and the integrity and confidentiality of the transmitted model parameters are preserved. (R3)*

Proof sketch. The system distributes the global model using a trusted communication channel, such as TLS, which provides mutual authentication between the aggregator and the participants. This ensures that the aggregator's identity is verified and that model parameters are transmitted with confidentiality, integrity, and authenticity. As a result, participants receive unaltered model updates from a trusted source, maintaining the security guarantees essential to the federated learning process. □

Claim 4 *The system is able to work under any anonymous channel implementation, thus improving adaptability to more efficient and secure anonymous channel protocols. (R4)*

Proof sketch. The proposed system requires only that the anonymous channel ensures unlinkability between the sender and the transmitted message. It does not rely on any other specific properties or implementation details. As a result, any anonymous channel satisfying this unlinkability requirement can be used without impacting the security or functionality of the system. □

Claim 5 *Every participation in a training round is cryptographically verified to originate from a pre-selected participant. In particular, the aggregator can confirm that each submission is associated with a valid, well-constructed token that embeds the participant's public key. (R5)*

Proof sketch. As described in Step 3.2: "Anonymous updates" of the reputation-based federated learning protocol (see Section 3.3), each participant submits a blinded token along with a zero-knowledge proof that attests to the token's correctness and the participant's inclusion in the authorized set. The aggregator verifies the proof without learning the participant's identity or secret key. This guarantees that each submission originates from a legitimate, pre-selected user and is cryptographically bound to a valid public key. □

Claim 6 *During the verification of participation and token construction, the aggregator does not learn any identifying information about the user, thereby preserving participant anonymity. (R5)*

Proof sketch. The system preserves anonymity through the use of blinding and zero-knowledge proofs. As detailed in Section 3.3, each participant submits a zero-knowledge proof π_i^T that confirms knowledge of a secret key corresponding to a public key in the Merkle tree, without revealing the key or the participant's identity. Additionally, the token t is blinded to produce t', preventing the aggregator from linking it to any specific user. As a result, the aggregator can verify that the submission is valid and originates from an authorized participant, but cannot infer any identifying information. □

Claim 7 *Only the aggregator is capable of decrypting and reading the parameters sent by the participant users, ensuring that any external observer is unable to access the model updates. (R6)*

Proof sketch. As described in Step 3.2: "Anonymous Updates" and Step 3.3: "Global Model Update and Reward Distribution" (see Section 3.3), each participant encrypts the local model update using a secure encryption protocol (ECIES [10]. The update $theta_i$ is encrypted under a secure key. Accordingly, only the aggregator, who holds the corresponding key, can decrypt the masked update. Under standard cryptographic assumptions, no external party can extract information from the encrypted seed or the ciphertext. Thus, the confidentiality of model updates is preserved against external observers. □

Claim 8 *The aggregator is able to perform robust statistical analyses and employ advanced anomaly detection methods to identify and mitigate malicious updates within the received parameters. (R6)*

Proof sketch. After decryption, the individual model updates θ^i are available in plaintext exclusively to the aggregator. This enables the application of statistical analyses and anomaly detection techniques, such as the Krum algorithm [5], to identify adversarial inputs and ensure that the aggregation process remains robust even in the presence of malicious participants. □

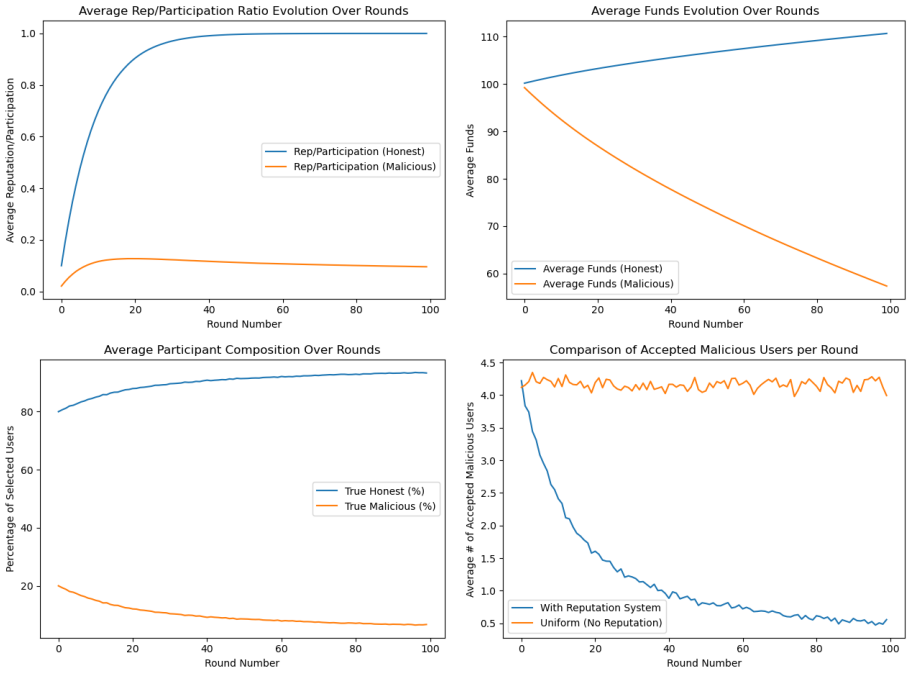


Fig. 2. Simulation results showing reputation evolution, funds distribution, participant composition, and accepted malicious update comparison.

Claim 9 *The reputation system incentivizes honest participation by rewarding reliable behavior and penalizing dishonest updates. (R7)*

Proof sketch. Simulation results (see Figure 2) demonstrate the effectiveness of the reputation system in promoting honest participation. In a scenario with 1000 users, 100 selected per round over 100 rounds, including 20% malicious users and 80% detection accuracy, the system exhibits the following behaviors:

- Honest users' reputations increase steadily and stabilize, remaining unaffected by malicious users.
- Honest users accumulate funds, while malicious users experience declining earnings.
- The proportion of malicious participants selected per round decreases progressively.
- The number of accepted malicious updates is significantly lower than in a system without reputation control.

□

Claim 10 *Tokens in the proposed system are cryptographically unforgeable; only a legitimate user who participates honestly in a round can generate and redeem a valid token to gain reputation. (R8)*

Proof sketch. As described in Step 3.3: "Global model update and reward distribution" (see Section 3.3), the aggregator signs the blinded token t' using a secure EdDSA scheme with private key x and a nonce k_i tied to P_i'. Thanks to the unforgeability of the signature scheme and the collision resistance of the hash function used in token construction, it is computationally infeasible for an adversary without knowledge of x to produce a valid signature on an unauthorized token. Therefore, only a legitimate participant following the protocol can generate and redeem a valid token. □

Claim 11 *A user cannot double spend a token to receive more than one reward per round, ensuring that reputation is increased only in accordance with a single, valid participation. (R9)*

Proof sketch. In each training round, a dedicated Membership Smart Contract $\mathcal{MSC}_l$ is deployed, which records the public keys pk_i of the selected participants. The contract enforces that each pk_i may redeem at most one token per round. Upon successful redemption, the contract updates its state to prevent further claims from the same key. Furthermore, each round uses a unique aggregator public key $PK^l_{A_C}$, which binds tokens and signatures to that specific round. Together, the per-round membership restriction and the use of round-specific keys ensure that any attempt to reuse or duplicate a token is detected and rejected. As a result, double spending is cryptographically prevented. □

Claim 12 *The cryptographic operations required by the proposed system execute efficiently on IoT devices, completing within a practical time frame. (R10)*

Proof sketch. The core cryptographic operations, i.e., blind signature generation, zero-knowledge proofs for token construction and membership verification, and signature verification, were benchmarked on a Raspberry Pi 3B using a local Ganache blockchain. Table 1 reports execution times for rounds with 8, 16, and 32 users. The total computation time remains nearly constant, with only a logarithmic increase in the membership proof due to the increased Merkle tree depth. All operations complete in under 7 seconds (22 seconds in total), indicating that the proposed system is suitable for deployment on resource-constrained IoT devices. □

# Users	claimTx	blindSignGen	zkToken	zkRedeem	redeemTx	Total
8 Users	4.38	5.45	5.25	6.08	1.21	22.39
16 Users	4.38	5.51	5.37	6.02	1.30	22.59
32 Users	4.39	5.52	5.55	6.20	1.29	22.99

Table 1. Performance Metrics by User Group (in seconds)

5 Conclusions and Future Work

This paper presents a novel, secure, and privacy-preserving federated learning framework for smart grid load forecasting, effectively combining blockchain technologies and cryptographic mechanisms to address critical privacy and security concerns. The proposed system enhances user privacy by employing anonymous update submissions, zero-knowledge proofs, and blind signatures to unlink model updates from participants' identities. Additionally, the blockchain-based reputation system incentivizes honest behavior and penalizes malicious actions, significantly reducing the risk of compromised model training. The evaluation shows that this integrated approach effectively handles data heterogeneity, ensures participant anonymity and privacy, penalizes dishonest users without blaming honest ones, and operates efficiently on constrained IoT devices.

Future research will include the deployment of a real-world federated learning setup to simulate attacks and study their impact on model performance. We also plan to apply this framework in future work to explore dynamic pricing schemes in smart grids.

Acknowledgments and disclaimer

This research is supported by Project HERMES, funded by the European Union NextGenerationEU/PRTR via INCIBE; by Projects PID2021-125962OB-C32 "SECURING/DATA" and PID2024-156914OB-C42 "SAFE/DATA", funded by MCIN/AEI/ 10.13039/501100011033; and by Grant SGR2021-00115, funded by AGAUR, Generalitat de Catalunya. Joan Ferré-Queralt is supported by the Spanish Government under an FPU grant (ref. FPU23/01275). Alexandre Viejo is a Serra-Húnter Fellow.

References

1. Agency, I.E.: Global electricity demand rising in 2021. IEA (2021), https://www.iea.org/reports/global-energy-review-2021
2. Ahmad, N., Ghadi, Y., Adnan, M., Ali, M.: Load forecasting techniques for power system: Research challenges and survey. IEEE Access **10**, 71054–71090 (2022)
3. Amral, N., Ozveren, C., King, D.: Short term load forecasting using multiple linear regression. In: 2007 42nd International universities power engineering conference. pp. 1192–1198. IEEE (2007)
4. Bilgin, Z., Tomur, E., Ersoy, M.A., Soykan, E.U.: Statistical appliance inference in the smart grid by machine learning. In: 2019 IEEE 30th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC Workshops). pp. 1–7. IEEE (2019)
5. Blanchard, P., El Mhamdi, E.M., Guerraoui, R., Stainer, J.: Machine learning with adversaries: Byzantine tolerant gradient descent. Advances in neural information processing systems **30** (2017)
6. Blanco-Justicia, A., Domingo-Ferrer, J., Martínez, S., Sánchez, D., Flanagan, A., Tan, K.E.: Achieving security and privacy in federated learning systems: Survey, research challenges and future directions. Engineering Applications of Artificial Intelligence **106**, 104468 (2021)
7. Che, J., Wang, J.: Short-term load forecasting using a kernel-based support vector regression combination model. Applied energy **132**, 602–609 (2014)
8. Dudek, G.: Short-term load forecasting using random forests. In: Intelligent Systems' 2014: Proceedings of the 7th IEEE International Conference Intelligent Systems IS'2014, September 24-26, 2014, Warsaw, Poland, Volume 2: Tools, Architectures, Systems, Applications. pp. 821–828. Springer (2015)
9. Fekri, M.N., Grolinger, K., Mir, S.: Distributed load forecasting using smart meter data: Federated learning with recurrent neural networks. International Journal of Electrical Power & Energy Systems **137**, 107669 (2022)
10. Gayoso Martínez, V., Hernández Encinas, L., Sánchez Ávila, C.: A survey of the elliptic curve integrated encryption scheme. JOURNAL OF COMPUTER SCIENCE AND ENGINEERING (2010)
11. He, Y., Luo, F., Ranzi, G., Kong, W.: Short-term residential load forecasting based on federated learning and load clustering. In: 2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm). pp. 77–82. IEEE (2021)
12. He, Y., Luo, F., Sun, M., Ranzi, G.: Privacy-preserving and hierarchically federated framework for short-term residential load forecasting. IEEE Transactions on Smart Grid **14**(6), 4409–4423 (2023)
13. Jebreel, N.M., Domingo-Ferrer, J., Blanco-Justicia, A., Sánchez, D.: Enhanced security and privacy via fragmented federated learning. IEEE Transactions on Neural Networks and Learning Systems (2022)
14. Kong, W., Dong, Z.Y., Jia, Y., Hill, D.J., Xu, Y., Zhang, Y.: Short-term residential load forecasting based on lstm recurrent neural network. IEEE transactions on smart grid **10**(1), 841–851 (2017)
15. Lahouar, A., Slama, J.B.H.: Day-ahead load forecast using random forest and expert input selection. Energy Conversion and Management **103**, 1040–1051 (2015)
16. Lei, J., Wang, L., Pei, Q., Sun, W., Lin, X., Liu, X.: Privgrid: Privacy-preserving individual load forecasting service for smart grid. IEEE Transactions on Information Forensics and Security (2024)

17. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., Agüera y Arcas, B.: Communication-efficient learning of deep networks from decentralized data. In: 20th International Conference on Artificial In telligence and Statistics (AISTATS) (2017)
18. Melis, L., Song, C., De Cristofaro, E., Shmatikov, V.: Exploiting unintended feature leakage in collaborative learning. In: 2019 IEEE symposium on security and privacy (SP). pp. 691–706. IEEE (2019)
19. Mollah, M.B., Zhao, J., Niyato, D., Lam, K.Y., Zhang, X., Ghias, A.M., Koh, L.H., Yang, L.: Blockchain for future smart grid: A comprehensive survey. IEEE Internet of Things journal **8**(1), 18–43 (2020)
20. Savi, M., Olivadese, F.: Short-term energy consumption forecasting at the edge: A federated learning approach. IEEE Access **9**, 95949–95969 (2021)
21. Selvam, C., Srinivas, K., Ayyappan, G., Sarma, M.V.: Advanced metering infrastructure for smart grid applications. In: 2012 International Conference on Recent Trends in Information Technology. pp. 145–150. IEEE (2012)
22. Soykan, E.U., Bilgin, Z., Ersoy, M.A., Tomur, E.: Differentially private deep learning for load forecasting on smart grid. In: 2019 IEEE Globecom Workshops (GC Wkshps). pp. 1–6. IEEE (2019)
23. Taïk, A., Cherkaoui, S.: Electrical load forecasting using edge computing and federated learning. In: ICC 2020-2020 IEEE international conference on communications (ICC). pp. 1–6. IEEE (2020)
24. Tsikalakis, A., Hatziargyriou, N.: Environmental benefits of distributed generation with and without emissions trading. Energy Policy **35**(6), 3395–3409 (2007)
25. Tun, Y.L., Thar, K., Thwal, C.M., Hong, C.S.: Federated learning based energy demand prediction with clustered aggregation. In: 2021 IEEE International Conference on Big Data and Smart Computing (BigComp). pp. 164–167. IEEE (2021)
26. Wang, Y., Jia, M., Gao, N., Von Krannichfeldt, L., Sun, M., Hug, G.: Federated clustering for electricity consumption pattern extraction. IEEE Transactions on Smart Grid **13**(3), 2425–2439 (2022)
27. Zhang, M.G.: Short-term load forecasting based on support vector machines regression. In: 2005 International Conference on Machine Learning and Cybernetics. vol. 7, pp. 4310–4314. IEEE (2005)
28. Zhao, Y., Xiao, W., Shuai, L., Luo, J., Yao, S., Zhang, M.: A differential privacy-enhanced federated learning method for short-term household load forecasting in smart grid. In: 2021 7th International Conference on Computer and Communications (ICCC). pp. 1399–1404. IEEE (2021)