Predicción de Carga Segura y Privada en Smart Grids mediante Incentivos Basados en Blockchain

Joan Ferré-Queralt *, Jordi Castellà-Roca*, Alexandre Viejo*

* Universitat Rovira i Virgili
Dept. d'Enginyeria Informàtica i Matemàtiques, CYBERCAT,
Av. Països Catalans 26, E-43007 Tarragona, Spain
{joan.ferreq, alexandre.viejo, jordi.castella}@urv.cat

***Resumen*—Éste es un resumen extendido de la publicación [1]. Las redes eléctricas inteligentes requieren predicciones de carga precisas, y aun utilizando *federated learning*, éste sigue presentando riesgos de privacidad y ataques de participantes maliciosos. Este trabajo presenta un marco seguro para *federated learning* en *smart grids*, basado en envíos anónimos, *zero-knowledge proofs* y firmas ciegas para proteger la identidad y los datos del usuario. Además, se introduce un mecanismo de reputación en *blockchain* que incentiva el comportamiento honesto y un *clustering* distribuido para manejar la heterogeneidad de los datos. La evaluación muestra que el sistema preserva la privacidad, mantiene la integridad del modelo y funciona eficientemente en dispositivos IoT.**

***Index Terms*—smart grid, privacidad, seguridad, load forecasting, federated learning, blockchain**

I. Introducción

La predicción precisa de la demanda eléctrica constituye un elemento clave en las *smart grids* modernas, especialmente ante el incremento global del consumo [2]. La gestión eficiente de los recursos energéticos se complica aún más con la creciente adopción de recursos energéticos distribuidos (DERs). Aunque estos recursos contribuyen a la reducción de las emisiones y de la dependencia de combustibles fósiles, su variabilidad inherente aumenta la complejidad del equilibrio entre generación y demanda, reforzando la necesidad de técnicas avanzadas de *load forecasting* [3].

La *Advanced Metering Infrastructure* (AMI) facilita este proceso al recopilar datos detallados y en tiempo real del consumo energético de los usuarios. Diversos enfoques han demostrado mejoras significativas en la precisión de las predicciones [4], por ejemplo modelos de *deep learning* como las redes LSTM [3]. Sin embargo, el acceso a datos tan detallados plantea serios riesgos de privacidad: los patrones de consumo pueden revelar información sensible sobre los hábitos, presencia en el hogar y actividades diarias de los usuarios [5].

Para ello, resulta fundamental diseñar mecanismos que garanticen la confidencialidad de los datos y fomenten la participación segura de los usuarios. En este contexto, la tecnología *blockchain*, gracias a su naturaleza descentralizada y resistente a la manipulación, ofrece capacidades prometedoras para construir entornos de intercambio de datos confiables [6].

I-A. Trabajos relacionados

Los primeros trabajos en privacidad para *load forecasting*, como Soykan et al. [7], aplicaron *differential privacy* a modelos LSTM, pero dependían de un entrenamiento centralizado que introduce riesgos al concentrar los datos en un único agregador. Como alternativa, el *federated learning* evita compartir datos brutos y permite entrenar modelos de forma distribuida. Taik et al. [8] demostraron su viabilidad en smart grids, y posteriores estudios compararon variantes como FedSGD y FedAVG [9]. Aún así, la privacidad puede verse comprometida al inferirse información sensible a partir de las actualizaciones del modelo [10].

Diversas propuestas han tratado de reforzar la privacidad: Zhao et al. [11] incorporan *differential privacy* en las actualizaciones, aunque con pérdida de precisión; He et al. [12] aplican *k-means* para abordar la heterogeneidad, pero requieren compartir datos finos de consumo con el agregador; Savi et al. [13] y Tun et al. [14] intentan mitigar esta exposición compartiendo solo promedios o propiedades del hogar, aunque aún exponen información y requieren confiar en el agregador.

Para eliminar la dependencia del agregador durante el *clustering*, se han explorado métodos distribuidos. He et al. [15] presentan un enfoque jerárquico que mejora este aspecto. Sin embargo, no solucionan el problema de que se pueda inferir información a partir de las actualizaciones del modelo de los usuarios.

Otros trabajos emplean cifrado homomórfico y MPC para ocultar completamente las actualizaciones [16]. Aunque evita filtraciones, introduce un problema crítico: las actualizaciones cifradas no pueden atribuirse a usuarios concretos, impidiendo mecanismos de seguridad esenciales como Krum [17], cuya aplicación requiere inspeccionar los gradientes para detectar ataques de *model poisoning*.

Finalmente, Jebreel et al. [18] proponen un esquema basado en fragmentación de parámetros y reputación que protege parcialmente la privacidad. No obstante, asume la homogeneidad de datos y su sistema reputacional puede ser manipulado: un atacante puede enviar fragmentos maliciosos a usuarios honestos, provocando que los marque erróneamente como adversarios. Además, la gestión centralizada del registro de reputación concentra la confianza en el agregador, permitiendo comportamientos injustos como bloquear recompensas legíti-

mas.

I-B. Contribución

La principal contribución de este trabajo es la incorporación de un mecanismo de reputación dentro del federated learning, adaptado específicamente a entornos de smart grid. La propuesta aborda retos clave del *load forecasting*, como la heterogeneidad de datos, los riesgos de privacidad y la necesidad de colaboración confiable.

II. Visión general del sistema

Esta sección presenta los actores que intervienen, los requisitos principales y un resumen del funcionamiento del protocolo.

II-A. Actores y componentes

Los *prosumidores* (U_i) son los usuarios finales que poseen datos privados de consumo X_i, formados por distintos patrones eléctricos. Cada patrón diario E_d^i contiene los valores de carga del día d:

$$E_d^i = \begin{bmatrix} E_{d,1}^i & E_{d,2}^i & \dots & E_{d,T}^i \end{bmatrix}.$$

Los prosumidores entrenan localmente un modelo utilizando estos datos, evitando su revelación a terceros y preservando la desvinculación entre su identidad y las actualizaciones enviadas.

El objetivo del sistema es entrenar un modelo general de *load forecasting*. El modelo $\mathcal{M}(E_{d-1}^i, \theta)$ predice la demanda del día siguiente $Y_{i,d}$ a partir del perfil del día anterior:

$$Y_{i,d} = \mathcal{M}(E_{d-1}^i, \theta) = \begin{bmatrix} y_1 & y_2 & \dots & y_T \end{bmatrix}.$$

Los parámetros θ se ajustan minimizando el error cuadrático medio $MSE(\theta)$. En federated learning, cada usuario mantiene un modelo local $\mathcal{M}^i$ con parámetros θ^i, actualizados mediante descenso de gradiente $\theta^i \leftarrow \theta^i - \alpha \nabla MSE(\theta^i)$.

Tras las actualizaciones locales, se realiza la agregación global mediante FedAVG [19]:

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

Para mejorar la precisión y tratar la heterogeneidad de datos, un *General Aggregator* (A_G) coordina un proceso distribuido de *clustering*, generando grupos de usuarios con características similares. Cada grupo es gestionado por un *Cluster Aggregator* (A_C), encargado de su entrenamiento federado.

El sistema integra además un mecanismo de reputación implementado mediante los siguientes *smart contracts*:

- *Reputation Smart Contract* ($\mathcal{RSC}$): Gestiona la puntuación de reputación de cada usuario.
- *Membership Smart Contract* ($\mathcal{MSC}_l$): Define los usuarios participantes en la ronda l, gestiona recompensas y actualiza reputaciones.
- *Bulletin Board Smart Contract* ($\mathcal{BBSC}$): Publica de forma transparente las recompensas emitidas por el proveedor del servicio.

II-B. Requisitos del sistema

El marco de federated learning propuesto para *load forecasting* debe cumplir los siguientes requisitos:

1. *Reputación dinámica*: Debe incorporarse un sistema de reputación que evalúe continuamente la fiabilidad de los participantes y oriente su selección.
2. *Canal anónimo*: Las comunicaciones no deben permitir vincular identidades con actualizaciones.
3. *Autenticación privada*: Los participantes deben autenticarse sin revelar su identidad real.
4. *Detección de actualizaciones maliciosas*: El *Cluster Aggregator* debe identificar intentos de *poisoning* sin que terceros accedan a los parámetros.
5. *Incentivos*: El sistema debe recompensar comportamientos honestos y penalizar contribuciones maliciosas.
6. *No falsificación de recompensas*: Las recompensas deben ser válidas únicamente si derivan de participación legítima.
7. *Prevención de doble gasto*: Cada recompensa debe canjearse una sola vez por ronda.
8. *Eficiencia en IoT*: El diseño debe ser viable en dispositivos con recursos limitados, como los smart meters.

II-C. Protocolo de Federated learning

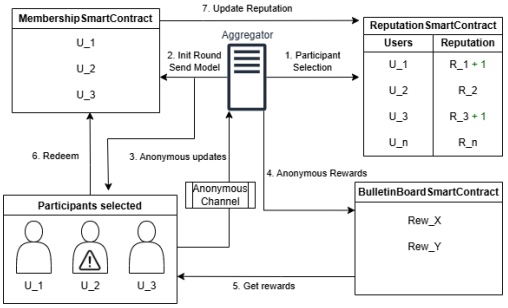


Figura 1. Visión general del protocolo propuesto.

Tras la fase de *clustering* distribuido coordinada por el agregador general (A_G), el protocolo continúa según los pasos mostrados en la Figura 1, involucrando a los *cluster aggregators* (A_C) y a los usuarios (U_i):

1. En cada ronda, A_C selecciona un subconjunto de usuarios basándose en sus puntuaciones de reputación almacenadas en $\mathcal{RSC}$.
2. A_C despliega un smart contract $\mathcal{MSC}$ para registrar a los usuarios seleccionados y envía el modelo inicial.
3. Cada U_i realiza el entrenamiento local y envía sus actualizaciones mediante un canal anónimo, junto con un token anónimo para reclamar la recompensa y una prueba de participación.
4. A_C verifica las actualizaciones y las pruebas recibidas. Si estas son válidas, firma el token correspondiente y lo publica en la $\mathcal{BBSC}$.
5. Cada U_i verificado como honesto puede recuperar su token firmado desde la $\mathcal{BBSC}$.
6. El usuario canjea el token en la $\mathcal{MSC}_l$, siguiendo un segundo protocolo de verificación para recibir la recompensa.
7. Finalmente, la reputación de cada U_i recompensado se actualiza en la $\mathcal{RSC}$.

III. Nuestra Propuesta

Esta sección describe de forma resumida el funcionamiento del sistema propuesto, estructurado en tres fases principales: clustering distribuido, inicialización del sistema y federated learning basado en reputación.

III-A. Paso-1: Clustering distribuido

El sistema emplea el esquema de clustering distribuido de [15] para agrupar a los usuarios sin compartir datos brutos. El agregador general (A_G) publica los centroides iniciales y, en cada iteración:

1. Cada usuario U_i asigna sus patrones diarios E_d^i al centroide más cercano.
2. Cada U_i calcula sus medias locales y envía las actualizaciones $\Delta_k^{i,l}$ y $n_k^{i,l}$ a A_G.
3. A_G actualiza los centroides.

El proceso finaliza por convergencia, y cada usuario se asigna al centroide más cercano según su patrón medio $\bar{E}^i$.

III-B. Paso-2: Inicialización del sistema

Tras el *clustering*, se inicializan los componentes necesarios:

1. Cada A_C despliega los smart contracts $\mathcal{RSC}$ y $\mathcal{BBSC}$.
2. Cada usuario genera un par de claves $pk_i = \mathcal{H}(sk_i)$ y se identifica mediante credenciales verificables.
3. A_C registra los pk_i en $\mathcal{RSC}$ e inicializa reputaciones.

III-C. Paso-3: Federated learning con reputación

El entrenamiento se desarrolla en rondas. Cada ronda l consta de cuatro subetapas.

III-C1. Paso-3.1: Inicialización de la ronda:

1. A_C genera un par de llaves EdDSA $(sk_{A_C}^l, PK_{A_C}^l)$.
2. Selecciona participantes según su reputación.
3. Genera un smart contract $\mathcal{MSC}_l$.
4. Registra los pk_i seleccionados en $\mathcal{MSC}_l$ mediante un Merkle tree.
5. Publica puntos aleatorios de la curva P_i' y envía el modelo global θ a cada U_i.

III-C2. Paso-3.2: Actualizaciones anónimas: Cada usuario entrena localmente su modelo y envía sus actualizaciones de forma anónima:

1. U_i construye un hashclaim $HC_i = \mathcal{H}(sk, P_i')$ y una prueba ZK π_i^P demostrando pertenencia al Merkle tree y conocimiento de sk_i sin revelar pk_i ni sk_i.
2. Reclama un punto P_i' en $\mathcal{MSC}_l$ desde una cuenta anónima.
3. Ciega el punto y genera un token t y su versión cegada t' utilizando el punto.
4. Construye otra ZK-proof π_i^T demostrando validez del token y posesión de sk_i.
5. Encripta la actualización θ^i con ECIES y envía (c_i, t', HC_i, π_i^T) a A_C por un canal anónimo.

III-C3. Paso-3.3: Actualización del modelo global y distribución de recompensas:

1. A_C desencripta c_i para obtener θ^i.
2. Verifica π_i^T y descarta aportaciones inválidas.
3. Detecta comportamientos maliciosos usando, por ejemplo, Krum [17].
4. Firma cada token cegado aceptado σ_i' y publica (t_i', σ_i') en $\mathcal{BBSC}$.

III-C4. Paso-3.4: Descegado del token y canjeado:

1. U_i recupera σ_i' y lo descega: σ.
2. Construye una ZK-proof π_i^R demostrando la validez de la firma sin revelar información sensible.
3. Envía $(P_i, pk_i, \sigma, \pi_i^R)$ a $\mathcal{MSC}_l$.
4. Si todo es válido, $\mathcal{MSC}_l$ marca el token como canjeado y actualiza la reputación del usuario en $\mathcal{RSC}$.

IV. Evaluación

Esta sección evalúa los requisitos de la Sección II-B.

Claim 1. *El sistema incorpora un mecanismo robusto de reputación almacenado de forma inmutable y resistente a manipulaciones. (R1)*

Proof: Las reputaciones se almacenan en $\mathcal{RSC}$, protegido por primitivas de blockchain: hashes resistentes a colisiones y firmas no falsificables. Solo se actualizan tras una redención válida en $\mathcal{MSC}_l$, lo que impide aumentos fraudulentos. ■

Claim 2. *El sistema es compatible con cualquier implementación de canal anónimo que asegure desvinculación entre identidad y mensaje. (R2)*

Proof: El protocolo requiere únicamente anonimato y unlinkability; no depende de propiedades específicas del canal. Por tanto, puede emplear cualquier tecnología de canal anónimo estado del arte que satisfaga estos requisitos. ■

Claim 3. *Cada participación en una ronda se verifica como proveniente de un usuario preseleccionado. (R3)*

Proof: Como se describe en el protocolo, cada usuario envía un token cegado y una prueba ZK que demuestra pertenencia al Merkle tree de $\mathcal{MSC}_l$ sin revelar su identidad. Solo los usuarios seleccionados pueden generar tokens válidos. ■

Claim 4. *El agregador no aprende ninguna información identificativa del participante durante la verificación. (R3)*

Proof: El uso de blinding y pruebas ZK asegura que el agregador verifica validez y pertenencia sin aprender pk_i, sk_i ni el vínculo entre token y usuario. ■

Claim 5. *El agregador puede detectar y filtrar actualizaciones maliciosas mediante análisis estadístico. (R4)*

Proof: Una vez descifradas, las actualizaciones permiten emplear técnicas robustas como Krum [17] para identificar gradientes anómalos y excluirlos del modelo. ■

Claim 6. *El sistema incentiva la participación honesta y penaliza el comportamiento malicioso. (R5)*

Proof: Las simulaciones (Figura 2) muestran que las reputaciones de usuarios honestos crecen y se estabilizan, mientras que los usuarios maliciosos acumulan menor reputación; además, su presencia en rondas sucesivas disminuye. ■

Claim 7. *Los tokens son infalsificables; solo un participante legítimo puede producir y canjear un token válido. (R6)*

Proof: Los tokens cegados t' se firman con EdDSA una llave privada. Sin conocerla, generar una firma válida es computacionalmente inviable. Hashes y firmas garantizan unicidad y autenticidad. ■

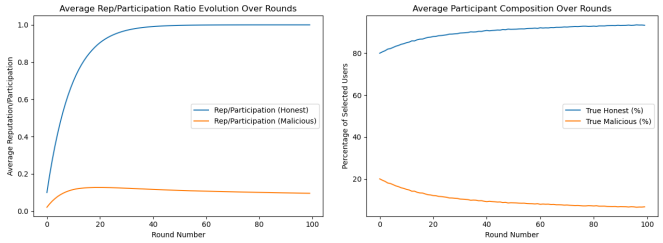


Figura 2. El primer gráfico muestra la evolución de la reputación, el segundo, la composición de los usuarios en cada ronda.

Claim 8. *Un usuario no puede gastar dos veces un token para recibir más de una recompensa por ronda. (R7)*

Proof: Cada $\mathcal{MSC}_l$ marca cada pk_i como "redeemed" tras el primer canje. Además, cada ronda emplea un $PK^l_{A_C}$ distinto, por lo que tokens entre rondas no son reutilizables. ■

Claim 9. *Las operaciones criptográficas son eficientes en dispositivos IoT. (R10)*

Proof: Las pruebas realizadas en una Raspberry Pi 3B (Tabla I) muestran que las operaciones principales (blind signatures, ZK proofs y verificación) finalizan en ∼23 s aun con 32 usuarios, con un crecimiento logarítmico, lo que confirma su viabilidad práctica. ■

# Users	Total
8 Users	22.39
16 Users	22.59
32 Users	22.99

Tabla I
TIEMPO DE EJECUCIÓN

V. CONCLUSIONES Y TRABAJO FUTURO

El trabajo propone un marco de federated learning seguro y preservador de la privacidad para *smart grids*, combinando anonimato, zero-knowledge proofs, blind signatures y un sistema de reputación basado en blockchain. El enfoque garantiza anonimato, integridad del modelo, detección de comportamientos maliciosos y rendimiento adecuado en dispositivos IoT. Los resultados muestran que el sistema gestiona eficazmente la heterogeneidad, incentiva la honestidad y reduce el impacto de participantes adversarios.

Como trabajo futuro, se planea implementar un entorno realista de federated learning para evaluar ataques avanzados y extender el marco a escenarios como *dynamic pricing*.

AGRADECIMIENTOS

Esta investigación está financiada por el Proyecto HERMES, financiado por la Unión Europea NextGenerationEU/PRTR a través de INCIBE; por los Proyectos PID2021-125962OB-C32 ≪SECURING/DATA≫ y PID2024-156914OB-C42 ≪SAFE/DATA≫, financiados por MCIN/AEI/10.13039/501100011033; y por la Ayuda SGR2021-00115, financiada por AGAUR, Generalitat de Catalunya. Joan Ferré-Queralt cuenta con el apoyo del Gobierno de España mediante una beca FPU (ref. FPU23/01275). Alexandre Viejo es Serra-Húnter Fellow.

REFERENCIAS

[1] J. Ferré-Queralt, J. Castellà-Roca, and A. Viejo, "Secure and privacy-preserving load forecasting in smart grids through blockchain-based incentives," in *International Conference on Network and System Security*. Springer, 2025, p. TOAPPEAR.

[2] I. E. Agency, "Global electricity demand rising in 2021," *IEA*, 2021. [Online]. Available: https://www.iea.org/reports/global-energy-review-2021

[3] N. Ahmad, Y. Ghadi, M. Adnan, and M. Ali, "Load forecasting techniques for power system: Research challenges and survey," *IEEE Access*, vol. 10, pp. 71 054–71 090, 2022.

[4] N. Amral, C. Ozveren, and D. King, "Short term load forecasting using multiple linear regression," in *2007 42nd International universities power engineering conference*. IEEE, 2007, pp. 1192–1198.

[5] Z. Bilgin, E. Tomur, M. A. Ersoy, and E. U. Soykan, "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)*. IEEE, 2019, pp. 1–7.

[6] M. B. Mollah, J. Zhao, D. Niyato, K.-Y. Lam, X. Zhang, A. M. Ghias, L. H. Koh, and L. Yang, "Blockchain for future smart grid: A comprehensive survey," *IEEE Internet of Things journal*, vol. 8, no. 1, pp. 18–43, 2020.

[7] E. U. Soykan, Z. Bilgin, M. A. Ersoy, and E. Tomur, "Differentially private deep learning for load forecasting on smart grid," in *2019 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2019, pp. 1–6.

[8] A. Taïk and S. Cherkaoui, "Electrical load forecasting using edge computing and federated learning," in *ICC 2020-2020 IEEE international conference on communications (ICC)*. IEEE, 2020, pp. 1–6.

[9] M. N. Fekri, K. Grolinger, and S. Mir, "Distributed load forecasting using smart meter data: Federated learning with recurrent neural networks," *International Journal of Electrical Power & Energy Systems*, vol. 137, p. 107669, 2022.

[10] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 691–706.

[11] Y. Zhao, W. Xiao, L. Shuai, J. Luo, S. Yao, and M. Zhang, "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)*. IEEE, 2021, pp. 1399–1404.

[12] Y. He, F. Luo, G. Ranzi, and W. Kong, "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)*. IEEE, 2021, pp. 77–82.

[13] M. Savi and F. Olivadese, "Short-term energy consumption forecasting at the edge: A federated learning approach," *IEEE Access*, vol. 9, pp. 95 949–95 969, 2021.

[14] Y. L. Tun, K. Thar, C. M. Thwal, and C. S. Hong, "Federated learning based energy demand prediction with clustered aggregation," in *2021 IEEE International Conference on Big Data and Smart Computing (BigComp)*. IEEE, 2021, pp. 164–167.

[15] Y. He, F. Luo, M. Sun, and G. Ranzi, "Privacy-preserving and hierarchically federated framework for short-term residential load forecasting," *IEEE Transactions on Smart Grid*, vol. 14, no. 6, pp. 4409–4423, 2023.

[16] J. Lei, L. Wang, Q. Pei, W. Sun, X. Lin, and X. Liu, "Privgrid: Privacy-preserving individual load forecasting service for smart grid," *IEEE Transactions on Information Forensics and Security*, 2024.

[17] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Advances in neural information processing systems*, vol. 30, 2017.

[18] N. M. Jebreel, J. Domingo-Ferrer, A. Blanco-Justicia, and D. Sánchez, "Enhanced security and privacy via fragmented federated learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[19] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *20th International Conference on Artificial In telligence and Statistics (AISTATS)*, 2017.