Computers and Electrical Engineering 123 (2025) 110012

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Blockchain-enabled distributed learning for enhanced smart grid security and efficiency

Alaa Awad Abdellatif [a,*], Khaled Shaban [a], Ahmed Massoud [b]

[a] Computer Science and Engineering Department, Qatar University, Qatar
[b] Electrical Engineering Department, Qatar University, Qatar

ARTICLE INFO

Keywords:
Distributed learning
Blockchain
Microgrid
Sidechains
Federated learning

ABSTRACT

This study introduces a secure, adaptable, and decentralized learning framework empowered by blockchain technology to enhance smart grid security and efficiency. Security is achieved through blockchain's ledger, ensuring data integrity, privacy, and resilience. Adaptability refers to the framework's ability to adjust to changing conditions, supporting multiple learning paradigms. Decentralization enhances fault tolerance by distributing control across nodes. Our framework excels in scalability, data-exchange security, and rapid response times, aiming to establish an intelligent blockchain-based smart grid supporting centralized learning (CL), federated learning (FL), and active federated learning (AFL). We present an innovative blockchain-based architecture customized to optimize information sharing and security within the blockchain. Our solution addresses various learning paradigm requirements by: (i) Selecting reliable entities for participation based on high-quality training data models; (ii) Acquiring a reliable subset of data for CL and AFL, balancing learning performance, latency, and cost; (iii) Adjusting blockchain configuration to align with specific learning paradigm requirements. Results from real-world datasets demonstrate superior performance compared to existing solutions. Our framework achieves high learning performance while minimizing latency and blockchain costs.

1. Introduction

The landscape of autonomous systems is undergoing a profound transformation driven by technologies such as the Internet of Things (IoT), distributed learning, and blockchain. The key to this shift lies in the assimilation of data from diverse smart sensors and/or IoT devices, seamless integration, meticulous processing, and implementation of robust security measures across distributed devices. This convergence has the potential to revolutionize the services offered by future smart cities.

Smart grids, a recent emergence, necessitate extensive data collection from various IoT devices across various locations. This data is crucial for comprehensive monitoring and management of intelligent services but presents significant challenges that must be addressed for optimal service delivery. Notably, real-time data acquisition from the Advanced Metering Infrastructure (AMI) poses challenges in smart grids [1], demanding efficient storage, thorough analysis, and seamless exchange to enhance the management, control, and efficiency of electrical grids. Furthermore, it is imperative to address security and privacy concerns. Safeguarding data integrity and user privacy during data processing and transmission, especially for secure remote access to valuable information like historical power consumption data, is essential for maintaining data trustworthiness.

* Corresponding author.
E-mail addresses: alaa.abdellatif@ieee.org (A.A. Abdellatif), khaled.shaban@qu.edu.qa (K. Shaban), ahmed.massoud@qu.edu.qa (A. Massoud).

https://doi.org/10.1016/j.compeleceng.2024.110012
Received 21 August 2024; Received in revised form 14 November 2024; Accepted 9 December 2024

Available online 21 December 2024

Intelligent services in smart grids, including demand response, Distributed Energy Resource (DER) management, and predictive maintenance, emphasize the need to address challenges related to data-intensive collection and processing, communication networks, and computational intelligence. By effectively addressing these challenges, we can enable seamless integration, comprehensive data utilization, and delivery of reliable, high-quality services. To address these challenges, leveraging distributed learning and blockchain technology pivotal. In this paper, we harness local resources distributed across various entities to collectively undertake specific learning tasks, such as Federated Learning (FL), distributed inference [2], and multi-agent reinforcement learning. Specifically, *security* is ensured by leveraging the decentralized ledger of blockchain technology, which guarantees data integrity and privacy by preventing unauthorized access and ensuring that data and models exchanged between entities remain tamper-proof [3]. This provides a secure environment for the implementation of various learning paradigms on the blockchain, including Centralized Learning (CL), FL, and Active Federated Learning (AFL) [4]. Furthermore, *adaptability* is achieved through the framework's ability to dynamically adjust to changing network conditions. The system can seamlessly switch between different learning paradigms (CL, FL, AFL) depending on the grid's requirements, such as scalability, resource availability, and data privacy concerns. Also, the decentralized nature of the blockchain allows for *decentralization*, meaning control is distributed across multiple nodes rather than relying on a central authority. This decentralization enhances the system's fault tolerance, as failures in individual nodes do not compromise the overall functionality or security of the smart grid network.

In summary, our main contributions can be categorized as follows:

1. *SDL Architecture:* We introduce a secure, distributed learning (SDL) architecture designed for large-scale smart grids. This framework combines blockchain technology and distributed learning to facilitate seamless data and model sharing among various entities.
2. *Entities' Interaction:* We explore three distinct modes for participating entities to utilize blockchain-shared information for distributed learning, including data sharing, model sharing, and a hybrid approach. Furthermore, we analyze the effects of each operational mode on learning performance, network cost, and latency.
3. *Blockchain Configuration Optimization:* We propose a reconfigurable blockchain configuration scheme that adapts to various learning paradigms, effectively balancing security, latency and cost considerations. Specifically, we formulate a novel optimization problem applicable to various learning paradigms, considering aspects such as learning quality, latency, cost, security level, and data characteristics within different entities. We introduce an analytical solution for this problem, achieved through its decomposition into two suboptimizations. Remarkably, these two subproblems exhibit manageable complexity, and their analytical proofs confirm convergence to the optimal solution of the original problem.
4. *Performance Evaluation:* Our findings demonstrate the effectiveness of the proposed framework across a variety of real-world datasets. Moreover, the framework's adaptability and effectiveness are affirmed through its successful application in scenarios like power consumption prediction.

In the following sections, we provide an overview of related work in Section 2, introduce the proposed SDL architecture and distributed learning paradigms in Section 3, outline the performance metrics and the problem formulated in Section 4, present the analytical solution for the formulated node, data, and blockchain configuration problem in Section 5, discuss the performance evaluation of the proposed framework in Section 6, and conclude the paper while highlighting future research directions in Section 7.

2. Related work

Blockchain technology has found applications in various domains of IoT systems, including autonomous vehicles, smart grid energy management, healthcare data sharing, and Internet of Drones (IoD) [5]. In the context of smart grids, blockchain networks serve multiple purposes, such as enabling peer-to-peer (P2P) energy trading [6], energy prediction [7], overseeing Distributed Energy Resources (DER), and AMI, as well as managing distributed energy resources. Various types of blockchain platforms, including public, consortium, and private blockchains, have been proposed for integration into smart grid systems [8]. Public blockchains excel in secure and distributed data sharing, while private or permissioned blockchains are more effective in contexts requiring greater control and privacy. Consortium and private blockchains, characterized by trusted entities, offer advantages such as reduced computational complexity and faster transaction approvals.

Extensive research has been conducted within the literature, covering diverse aspects related to blockchain frameworks, smart contracts, and consensus algorithms [9]. For instance, in [10], a hierarchical blockchain architecture was demonstrated as a solution to enhance the security of distributed energy trading within microgrids, mitigating potential data manipulation attacks. Another proposal, outlined in [11], introduced an edge blockchain architecture to address data access control in the context of smart grid data. This framework was designed specifically for lightweight IoT devices, effectively offloading the computational burdens of end users to edge servers within a consortium blockchain structure. Authors in [12] introduced a three-layer smart grid data collection method employing edge computing and blockchain. Data from smart meters was initially processed locally on a dedicated edge server, followed by global collection, and data is sent to the blockchain to maintain aggregated data privacy, with exclusive access by the control center. A recent study in [13] proposed an optimization approach for blockchain-based IIoT systems, reducing inefficiencies and bottlenecks through improved resource allocation and decision-making. The algorithm outperforms traditional methods, offering a more scalable and efficient solution for industrial applications, relevant to blockchain optimization in IIoT contexts.

In [8], a decentralized blockchain-based P2P energy marketplace is proposed, addressing privacy, trust, and governance issues. The marketplace utilizes a private permissioned blockchain, Hyperledger Fabric (HF), and its smart contracts. It incorporates a regulator to oversee operations and ensures data privacy through HF's private data collections, while maintaining transaction

integrity and auditability. In [14], a framework for P2P energy trading within and across microgrids was presented. This framework utilized FL to predict energy demand while upholding trust and privacy among all participants through the utilization of blockchain technology. Authors in [15] introduced a decentralized frequency control system for an islanded microgrid using blockchain and FL fractional order recurrent neural network. It employed a self-adaptive proportional–integral–derivative controller to address uncertainties in power generation during prosumer participation in islanded microgrid trading. [16] presented a decentralized FL scheme, named blockchain-based clustered FL, for non-intrusive load monitoring. This approach combined blockchain mechanisms with clustered FL, encouraging eligible clients to join FL through rewards based on data size and model performance. [17] utilized a permissioned energy blockchain to establish secure charging services for electric vehicles (EVs). This approach included a reputation-based delegated Byzantine fault tolerance consensus method and optimal smart contract design to meet EV requirements and operator objectives. In [7], the authors proposed a privacy-preserving, communication-efficient FL-based energy predictor for net-metering systems. Using a real power consumption/generation dataset, they developed a hybrid deep learning model for accurate future predictions and an Inner-Product Functional Encryption (IPFE) scheme for secure data aggregation during FL training, ensuring customer privacy by encrypting model parameters.

Notably, none of the existing studies have optimized blockchain behavior to support secure and scalable distributed learning in large-scale smart grids. This paper is the first to address this research gap by providing a solution specifically tailored to this challenge while ensuring the security of various distributed learning paradigms. A preliminary version of this study was presented in [18], which introduced a distributed architecture for secure data and model exchange among different microgrids. Although [18] offers a heuristic approach to reducing costs within the proposed architecture, it does not consider the optimal trade-off between security, latency, and cost. In this paper, we extend the work in [18] by: (i) mathematically analyzing the trade-offs between security, latency, and cost across various learning paradigms; (ii) formulating a novel optimization problem that accounts for learning quality, latency, cost, security level, and data characteristics within different microgrids; (iii) proposing an analytical solution to this problem; and (iv) enhancing the performance evaluation of the proposed framework, including a comparison with state-of-the-art solutions.

3. SDL architecture and distributed learning paradigms

This section introduces the SDL system architecture and outlines various distributed learning paradigms it can support for a variety of smart grid services.

3.1. SDL architecture

Real-time smart grid data access is crucial for effective monitoring and fault detection. However, sharing sensitive data among entities raises security and privacy concerns. To address this, we propose a scalable architecture with a main blockchain and multiple side chains (microchains), as shown in Fig. 1. While adaptable to various IoT applications, this paper focuses on the smart grid as a case study. In particular, our SDL architecture enhances accessibility and data sharing between different smart grid entities, ensuring reliable smart grid services while safeguarding against cybersecurity threats. The architecture leverages blockchain to enable a limited number of authorized entities to securely access the shared data by incorporating access control rules into smart contracts. Blockchain's stability and decentralization align well with the SDL architecture, making it a suitable choice for secure data sharing. The design of the micro-chain allows each microgrid operator to verify the authenticity and integrity of acquired information before sharing it within the main chain. This architecture facilitates secure data exchange and storage for power consumers, prosumers, DERs, energy storage, and microgrid operators. The main chain is owned by the government and supports secure access, processing, and exchange of information between main-grid entities, including microgrid operators, distributed network operators, and the General Electricity Corporation.

The security of this architecture relies on the collective resources of entities validating each transaction through a consensus mechanism like Delegated Proof of Stake (DPoS) [19]. Cyber attack would need to overcome the collective resources of all these entities to compromise data integrity. Moreover, integrating multi-signature protocols requires multiple independent verifications from distinct parties, adding an extra layer of security by ensuring that no single entity can unilaterally approve a transaction. This further strengthens data integrity and resilience against unauthorized access.

The SDL system involves entities taking on roles as data/model senders, Blockchain Manager (BM), and verifiers. This setup follows the traditional DPoS consensus algorithm [20]. Furthermore, it allows collaboration among smart grid entities, including consumers, DERs, and network operators, using interconnected side chains to distribute the workload between the main-chain and micro-chains, securing data exchange while confining the impact of potential attacks to specific micro-chains.

3.2. Distributed learning paradigms

The introduced SDL architecture accommodates various distributed learning paradigms, enabling participating entities to continuously learn online by sharing data, models, or both through information exchanged via blockchain. Each learning paradigm offers a unique balance between learning performance and network load. For instance, achieving high learning accuracy often necessitates extensive datasets, which, in turn, involves sharing a substantial volume of data to train complex models. However, this approach can be constrained by network costs and data privacy concerns. On the contrary, sharing pre-trained models can help preserve data privacy but might have a detrimental effect on learning accuracy, especially with non-Independent and Identically Distributed (non-IID) data [21]. In Fig. 2, we outline three types of information that can be exchanged in SDL and the associated blockchain modes, each with different blockchain demand and security levels, including:

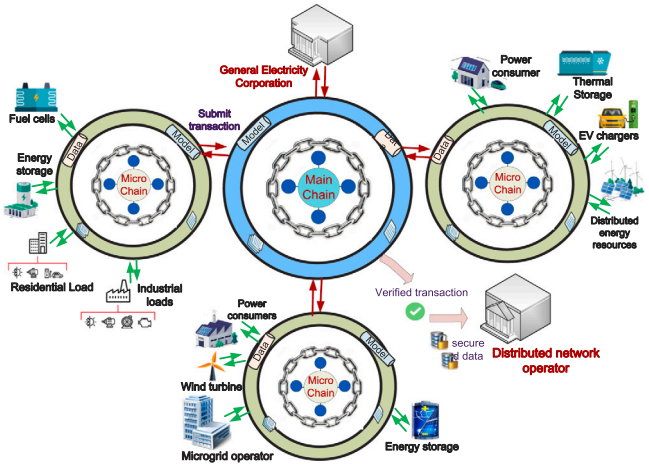
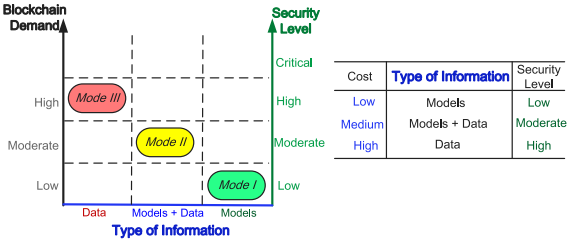


Fig. 1. Proposed SDL system architecture for large-scale smart grids.



Cost	Type of Information	Security Level
Low	Models	Low
Medium	Models + Data	Moderate
High	Data	High

Fig. 2. Blockchain modes based on the type of exchanged information and required security level.

1. **Mode I**: Exchanging acquired local data through Centralized Learning (CL);
2. **Mode II**: Exchanging learning models through Federated Learning (FL);
3. **Mode III**: Exchanging both models and small portions of data through Active Federated Learning (AFL).

Compared to CL, FL provides the added benefit of preserving data privacy, making it suitable for contexts with sensitive or legally protected data, such as users' power consumption in a smart grid. FL holds great promise for improving learning quality while maintaining data privacy, but it requires precise design and implementation to achieve high model accuracy, especially with non-IID data. In contrast to traditional FL, the proposed AFL paradigm allows limited data exchange among participating entities, aiming to alleviate data distribution imbalances and accelerate the learning process. This approach enhances overall FL performance.

The SDL system enables various entities within micro-chains and the main chain to exchange data and models while considering different security levels and costs. For example, during FL setup, sharing local models reduces blockchain costs, particularly in a less restrictive blockchain structure with fewer verifiers. In contrast, CL requires a fully-restricted blockchain with the maximum number of verifiers to ensure higher security, which increases costs. The number of verifiers directly impacts both security and cost, with more verifiers enhancing security but raising costs [20]. Furthermore, we highlight that a traditional proof-of-work (PoW) consensus algorithm can be used in CL to support higher security, while FL and AFL benefit from lightweight consensus algorithms, such as proof-of-authority, to enhance scalability and reduce computational overhead. Also, smart contracts can be customized for each paradigm: in CL, they enforce data access protocols and ensure integrity, whereas in FL and AFL, they enable dynamic node selection and flexible validation of data models.

4. Learning process characterization and problem formulation

This section presents the main performance metrics considered in this work and formulates optimization problem related to blockchain configuration with node and data selection.

4.1. Performance metrics

In our case study, we consider a permissioned blockchain platform that utilizes a DPoS consensus algorithm, which requires pre-selected verifiers for the consensus process with moderate cost. Additionally, we implement the Blockchain Manager (BM) concept, allowing any entity within the architecture to assume the role of BM for managing blockchain configurations. The BM's

responsibilities include collecting transactions, sending unverified blocks to verifiers, updating blockchain configurations based on requested modes, and finalizing the block verification process. The flexibility of blockchain modes is essential to strike the right balance between data quality, security, and cost in alignment with the CL paradigm's requirements.

We consider *data* to be both readings and extracted features from IoT devices or MEN. We represent the set of MEN used for data acquisition as N, with x_n denoting the amount of data or samples acquired from MEN $n \in N$. Each data sample j is denoted as (d_j, y_j), where d_j is the data observed by MEN, y_j is the associated class or label, with $y_j \in \mathcal{L}$, and $\mathcal{L}$ is the set of all possible classes.

For selecting the most representative nodes and data for training a machine learning (ML) model, we assess their impact on learning performance. Consequently, in the following sections, we analytically present key performance metrics that influence model learning, including node reliability, learning error, latency, and associated costs.

4.1.1. Nodes' reliability

In a collaborative learning setup, the learning model (LM) can still be generated by the *Requester* even if one or more participant nodes have responded with their data or local models before the deadline. Thus, the total reliability obtained at the *Requester* can be defined as [22]:

$$R_t = 1 - \prod_{i=1}^{\tilde{N}} \left(1 - R_i\right) = 1 - \prod_{i=1}^{\tilde{N}} \left(1 - r_i^{\tilde{m}} \cdot D_i^q\right), \tag{1}$$

where R_i is the reliability of node i, and there are a total of $\tilde{N}$ selected nodes. The node's reliability is assessed based on two quality metrics: the normalized model reliability $r_i^{\tilde{m}}$ and the data quality indicator D_i^q. The model reliability is a function of the number of epochs e executed at a specific node. Based on our experiments and survey of various measurements [4,23], we find that the model reliability of most LMs empirically follows a logarithmic law, i.e.,

$$r_i^m = c_1 \log(c_2 + c_3 \cdot e_i), \tag{2}$$

where the coefficients $c_1 - c_3$ depend on the ML model and dataset being used. It is important to note that for CL, the model's reliability is always 1 ($r_i^m = 1$) because the performance of CL depends solely on the acquired data from different nodes.

To define data quality, we utilize the Kullback–Leibler divergence (KLD) to measure data diversity. KLD quantifies the difference between the data distribution at a participant node H_i and the global data distribution Q of all nodes [21], defined as:

$$D_{KL}(H_i \,||\, Q) = \sum_{j=1}^{J} H_i(l_j) \log \frac{H_i(l_j)}{Q(l_j)}, \tag{3}$$

where $\sum_{j=1}^{J} H_i(l_j) = 1$ and $\sum_{j=1}^{J} Q(l_j) = 1$, as well as $H_i(l_j) > 0$ and $Q(l_j) > 0$, for any $l_j \in \mathcal{L}$, such that $\mathcal{L} = \{l_1, l_2, \ldots, L\}$ is the set of all possible classes in the global *virtual* dataset (i.e., includes all data classes available at all nodes). A KLD of zero indicates that all data classes are available and uniformly distributed at this node, i.e., the local data at this node are IID. Thus, the data quality indicator is defined as:

$$D_i^q = 1 - \tilde{D}_{KL}(H_i \,||\, Q). \tag{4}$$

where $\tilde{D}_{KL}(H_i \,||\, Q)$ is the normalized KLD indicator.

4.1.2. Learning quality

The relationship between the average size X of a local dataset and the learning quality is *empirically predictable* [24,25]. Learning quality q is a critical metric for assessing the performance of LMs. For most LMs, learning quality is represented by a logarithmic relationship, i.e., $q \propto \log(c_4 + c_5 \cdot X)$, where c_4 and c_5 are constants dependent on model architecture and dataset, and X is the number of acquired data samples for training. This relationship holds for CL paradigm, while for other paradigms like FL and AFL paradigms, additional factors affect learning quality including the number of communication rounds, number of nodes participating in the training, class distribution of local datasets (i.e., the similarity of the local datasets), and local dataset sizes [26].

4.1.3. Latency

Latency measures the time taken for block verification within the blockchain. It encompasses four main steps: (i) unverified block transmission from the BM to the verifiers, (ii) block verification time, (iii) broadcasting verification results and comparing them among different verifiers, and (iv) transmission of verification feedback from the verifiers to the BM. Hence, the block latency T can be defined, according to [20], as

$$T = \frac{B}{r_d} + \max_{i \in \{v, \ldots, M\}} \left(\frac{a}{A_i}\right) + \psi \cdot B \cdot m + \frac{O}{r_u}, \tag{5}$$

where B is the block size, a is the required computational resources for block verification task, A_i is the amount of available computational resources at verifier i, O is the verification feedback size, and r_d and r_u are, respectively, the downlink and the uplink transmission rate, from the BM to the verifiers and vice versa. In (5), ψ is a predefined parameter that can be defined using the statistics from previous block verification processes (as detailed in [20]). It is worth noting that the time required for block verification can exhibit variations, primarily influenced by factors such as the choice of blockchain implementation framework, the endorsement/validation policy used, and the consensus algorithm used [27].

4.1.4. Learning cost

The learning cost is the total of operational and communication expenses incurred by both computing nodes and selected data nodes, defined as:

$$C = \sum_{n=1}^{N} \lambda_n \cdot x_n \cdot c_n + \left(\sum_{n=1}^{N} \lambda_n \cdot \frac{x_n}{B} \right) \sum_{i=1}^{m} c_i, \tag{6}$$

where c_n is incentive cost paid to a node n to participate in the learning process, x_n is the amount of acquired data from node n, and c_i is required cost per block for verifier i to participate in the verification process [22]. We emphasize that in the context of the FL paradigm, x_n specifically refers to the size of the exchanged model.

4.1.5. Security level

The security level is a key metric for assessing blockchain security. It is defined as a function of network scale, such that $S = \kappa \cdot m^{\theta}$, where κ is a constant given by the system and $\theta \geq 2$ is a coefficient representing network scale [20]. However, evaluating the security level of a blockchain involves considering various factors, including the consensus algorithm's inherent security properties, the total hash power dedicated to network security, and the number of nodes in the network.

4.2. Node, data, and blockchain configuration optimization

To address the challenge of finding an optimal balance between learning quality, latency, cost, and security, we propose a reconfigurable framework that for optimizing nodes, data and blockchain configuration. This framework aims to support different learning paradigms, ensuring the selection of the most representative nodes for collaborative learning, determining the amount of data or models to be shared, and optimizing the number of verifiers. This customization allows the blockchain to be tailored to the specific requirements of each learning paradigm while considering available resources. Hence, our framework is built on two fundamental premises:

1. The performance of LMs is highly dependent on the quantity and distribution of acquired data or models from different nodes.

2. It is important to carefully select the blockchain configuration, specifically regarding the number of selected nodes and verifiers. This ensures that an optimal trade-off among security, latency, and cost is maintained.

In light of these consideration, our framework aims to support various learning paradigms by: (i) Selecting a set of notes that best represents the collaborative learning process, (ii) Determining the minimum amount of information to be shared based on the specific learning paradigm, and (iii) Optimizing the number of verifiers. Given these requirements, we formulate a general optimization problem that considers the data characteristics of different nodes while accounting for learning quality, latency, cost, and security level. The primary objective of this optimization problem is to select the optimal set of notes $\tilde{N}$, where $\tilde{N} \in \mathcal{N}$, to participate in the learning process, the number of verifiers, and the amount of data x_n to be acquired from these nodes to minimize learning error. Thus, the node, data, and blockchain optimization problem is formulated as follows:

$$\mathbf{P}: \min_{\lambda_n, x_n, m} \quad \epsilon(p) \tag{7}$$

$$\text{such that: } \left(\sum_{n=1}^{N} \lambda_n \cdot \frac{x_n}{B} \right) T \leq T_{max}, \tag{8}$$

$$\sum_{n=1}^{N} \lambda_n x_n c_n + \left(\sum_{n=1}^{N} \lambda_n \frac{x_n}{B} \right) \sum_{i=1}^{m} c_i \leq C_{max}, \tag{9}$$

$$S_{min}(p) \leq \kappa \cdot m^{\theta}, \tag{10}$$

$$\upsilon \leq m \leq M, \tag{11}$$

$$\sum_{n=1}^{N} \lambda_n \leq N, \tag{12}$$

$$\lambda_n \in \{0, 1\}, \quad \forall n \in N \tag{13}$$

where λ_n is the node selection indicator, such that $\lambda_n = 1$ when node n is selected to participate in the learning process. Hence, $\tilde{N} = \sum_{n=1}^{N} \lambda_n$ is the number of selected nodes. The constraint in (8) ensures that the obtained latency is within the specified target delay deadline T_{max}, and the constraint in (9) limits the total learning cost to not exceed the maximum cost C_{max}. The constraints in (10) and (11) guarantee that a minimum security level $S_{min}(p)$, which varies depending on the learning mode p. The number of chosen verifiers is denoted as m, with its upper and lower bounds being M and υ, respectively. The constraints in (12) and (13) ensure that the number of selected nodes does not exceed the total number of available nodes.

The unknowns in (7) are the λ_n's, x_n's, and m indicating node selection, the amount of data or models to be acquired from each selected node, and the number of selected verifiers, respectively. It is worth noting that the variable x_n takes on different meanings based on the specific learning paradigm:

- For CL, x_n represents the size of acquired raw data.

- For FL, x_n denotes the size of exchanged models.
- For AFL, x_n encompasses both the size of acquired raw data and the size of exchanged models.

The formulated problem **P** is a combinatorial optimization problem, since it includes a large number of binary variables λ_n, equal to the number of available nodes, in addition to integer variables x_n's and m, which makes it hard to be solved using conventional optimization methods. In the following Lemma, we first prove that this problem is NP hard. In the next section, we propose an Optimal Node, Data, and Blockchain Configuration (ONDB) approach to solve this problem.

Lemma 1. *The node and data selection problem formulated in (7)–(13), is NP-hard*

The expressed node and data selection problem is a reduction from the Knapsack problem; a combinatorial optimization problem that aims at selecting a subset of items $S \subseteq \{1, 2, \dots, n\}$ from a group of n items with size $s_1, s_2, \dots, s_n$, values $v_1, v_2, \dots, v_n$, capacity B, and maximum total value V, such that $\sum_{i \in S} s_i \leq B$ and $\sum_{i \in S} v_i \geq V$. The knapsack is an NP-complete problem, and the formulated optimization problem in (7)–(13) maps to any given instance of the knapsack problem. Indeed, the objective of the formulated problem **P** is to select a subset $\mathcal{S}$ of $\tilde{N}$ nodes from the total N available nodes, and the amounts of data x_n to be acquired from these nodes. Therefore, the decisions are related to $\mathcal{S}$, which is a vector of $\tilde{N}$ elements that maps the s_i variables in the knapsack problem. A node n is selected if and only if $\lambda_n = 1$. By mapping the associated learning cost of the selected nodes into the size term in the knapsack problem, and mapping the term value in the knapsack problem to the opposite of the latency, the formulated optimization problem can be transformed into a special case of an NP-hard problem, which proves our Lemma.

It is important to note that the formulated problem in **P** is solved when training is required, typically during the initial deployment of the LM or in rare situations when significant network changes necessitate retraining.

5. Optimal node, data, and blockchain configuration solution

This section presents our approach to tackle the NP-hard problem outlined in **P**. We decompose the problem into two sub-problems, each dependent on specific decision variables. By solving these sub-problems independently and iteratively, the optimal solution for the original problem is maintained.

5.1. Optimization decomposition

To effectively break down the problem into two manageable sub-problems, we categorize the optimization variables in **P** into two groups: (i) node selection variables λ_n, and (ii) data and blockchain variables (x_n and m). This approach enables the decomposition of the original problem into a node selection sub-problem, and data selection with blockchain configuration sub-problem.

Lemma 2. *The optimization problem presented in **P** can be decomposed into the following two sub-optimization problems. Solving these sub-problems individually maintains the optimal solution.*

$$\textbf{SP1}: \min_{\lambda_n} \quad \epsilon(p) \tag{14}$$
subject to (8), (9), (12), (13),

and

$$\textbf{SP2}: \min_{x_n, m} \quad \epsilon(p) \tag{15}$$
subject to (8), (9), (10), (11).

This lemma asserts that the optimization problem in **P** can be decomposed into two sub-problems, and solving these sub-problems individually yields the optimal solution. The learning error $\epsilon(p)$ is primarily influenced by the distribution and the size of the training dataset (in CL and AFL), and the number of global iterations (in FL and AFL). To minimize this error, achieving a balanced dataset is crucial, as demonstrated and formally proved in [28] and experimentally presented in Section 4.1.2. Therefore, selecting the minimum number of nodes that contain samples from all data classes ensures the lowest error. By choosing nodes with the highest reliability while respecting cost and time constraints, we can always guarantee the minimum $\epsilon(p)$.

Proposition 1. *Optimal values of λ_n can be obtained by independently solving the sub-optimization problem in **SP1** for any values of x_n and m.*

For the amount of exchanged data in blockchain, the learning error is consistently a decreasing function of the acquired data size (in CL and AFL), and the number of global iterations (in FL and AFL). Hence, for any optimal λ_n, selecting the minimum number of verifiers that meet security constraints in (10), (11), and maximum number of data samples or global iterations to satisfy time and cost constraints in (8) and (9), respectively, will maintain the minimum error.

Proposition 2. *Optimal values of x_n and m can be obtained by independently solving the sub-optimization problem in **SP2** for any values of λ_n.*

Proposition 1 and Proposition 2 provides the basis for the validity of Lemma 2. This decomposition simplifies the original problem significantly and facilitates its analysis and implementation across various ML models and datasets.

5.2. Node selection optimization

In **SP1**, where the objective is to minimize the learning error ϵ, an efficient approach is needed to solve the node selection problem. To do this, we reformulate objective function as follows:

$$\begin{aligned} Z &= \max_{\lambda_n} R_t \\ &= \min_{\lambda_n} \prod_{i=1}^{\tilde{N}} \left(1 - r_i^m \cdot D_i^q\right), \\ &= \max_{\lambda_n} \left(\prod_{i=1}^{\tilde{N}} r_i^m \cdot D_i^q\right), \end{aligned} \tag{16}$$

This reformulation means that to minimize $\epsilon(p)$ for any learning paradigm p, we should selected nodes with the maximum reliability while adhering to the constraints in **SP1**. In other words, the nodes with the highest model reliability r_i^m and data quality D_i^q should be chosen.

To maximize model reliability r_i^m, we should select nodes with the maximum number of epochs e_n according to (2). Similarly, to maximize data quality at the *Requester* D_r^q, nodes with the maximum number of new classes and the minimum KLD should be prioritized according to (4).

In order to maximize D_r^q, (and therefore, minimize KLD), we should select nodes that collectively possess all data classes, as specified in (3). Thus, the optimization problem in **SP1** can be solved by selecting the minimum number of nodes with maximum number of epochs e_n, the highest number of new data classes $\tilde{k}_n$, and the lowest learning cost c_n. To accomplish this, a weighting coefficient δ_n for each node n is introduced. This coefficient allows nodes with the highest weights to be selected, taking into account the new information that will be gained by selecting a particular node, i.e., the new data classes that will be included. The weighting coefficient δ_n is defined as:

$$\delta_n = r_n^m \cdot D_r^q \cdot \frac{C^{max}}{C_n}. \tag{17}$$

By using this weighting coefficient, the node selection procedure, at each iteration, selects nodes with the highest weights. This helps maximize the number of acquired data classes and ensures that nodes with the lowest learning cost are chosen, this satisfying the constraints in (9). When a node with the maximum weight is selected, the weights of the remaining nodes are updating based on their data classes and those already selected.

5.3. Data selection with blockchain configuration optimization

Following the selection of the best nodes in **SP1**, we now proceed to determine the optimal amount of data to be acquired from these nodes and the number of verifiers needed for validation. This involves solving the optimization problem in **SP2**, for which we offer a closed-form solution.

First, to minimize latency and cost in **SP2**, we select the minimum number of verifiers m, that satisfies the constraint in (10). This is done using the following equation:

$$m = \left(\frac{S}{\kappa}\right)^{\frac{1}{\theta}}. \tag{18}$$

Second, to minimize the objective function in **SP2**, we aim to maximize the size of the total acquired data, denoted as X. This data can be raw data in the CL/AFL setup or shared models in the FL/AFL setup. The formula for X is as follows:

$$X = \sum_{n=1}^{N} \lambda_n x_n = \sum_{n=1}^{N} \lambda_n \left(x_r^n + I \cdot |W|\right), \tag{19}$$

where x_r^n is the amount of collected raw data from node n, I is the number of global iterations in FL/AFL, and $|W|$ is the size of the shared learning model in FL/AFL. Constraints in (8) and (9) must also be satisfied while maximizing X. To determine the maximum values of acquired data, denoted as X^t, according to the time constraint in (8), the following relationship is used:

$$\begin{aligned} T_{max} &= \frac{X^t}{B} \cdot T = \left(\sum_{n=1}^{N} \lambda_n \cdot \frac{x_n}{B}\right) T, \\ X^t &= \frac{T_{max} \cdot B}{T}. \end{aligned} \tag{20}$$

To satisfy the constraint in (9), the maximum value of acquired data, denoted as X^c, can be derived as:

$$\begin{aligned} C_{max} &= X^c \cdot c_n + \frac{X^c}{B} \sum_{i=1}^{m} c_i, \\ X^c &= \frac{C_{max}}{c_n + \sum_{i=1}^{m} c_i / B}. \end{aligned} \tag{21}$$

From (20) and (21), the optimal amount of data to be acquired from all nodes is:

$$X = \min\left\{X^c, X^t\right\}. \tag{22}$$

In the case of FL, $x_r^n = 0$, hence

$$x_n^{FL} = I \cdot |W| = \frac{X}{\tilde{N}}. \tag{23}$$

In the CL and AFL cases, the aim is to collect an equal amount of data for each data class from all selected nodes. Hence, the amount of data to be collected from a selected node n in the CL case is:

$$x_n^{CL} = \sum_{l_n} \frac{X}{L \cdot \tilde{n}_l}, \tag{24}$$

where L represents the total number of data classes across all nodes, and $\tilde{n}_l$ is the number of selected nodes that have class l. In the AFL case, only a portion of the missing data classes is exchanged between participating nodes to ensure a balanced dataset. It is assumed that if $\tilde{l}$ classes are missing, the amount of raw data shared through the blockchain will be $\tilde{l} \cdot \bar{x}$. The amount of data to be collected from a selected node n in the AFL scenario is determined based on whether n is in the set of selected nodes S that will initially share the missing data:

$$x_n^{AFL} = \begin{cases} \frac{X - \tilde{l} \cdot \bar{x}}{\tilde{N}} + \tilde{l} \cdot \bar{x}, & \text{if } n \in S, \\ \frac{X - \tilde{l} \cdot \bar{x}}{\tilde{N}}, & \text{if } n \notin S, \end{cases} \tag{25}$$

where S is the set of selected nodes that will initially share the missing data.

5.4. Optimal node, data, and blockchain configuration (ONDB)

In this subsection, we present the complete procedure for solving the formulated node, data, and blockchain configuration problem, referred to as the Optimal Node, Data, and Blockchain Configuration (ONDB) algorithm. Initially, we assume that the selected data amount x_n and the number of verifiers m are arbitrary and consistent across all nodes, i.e., $x_n^* = x, \quad \forall n \in N$. First, we solve the problem in **SP1** to determine the optimal nodes participating in the learning process. Following that, we solve the problem in **SP2** to obtain the optimal values for x_n and m. Upon solving the problem in **SP2**, we repeat the node selection procedure to determine the actual values of λ_n based on the obtained x_n and m. We highlight that the node and data selection process in ONDB is automated, based on the reliability of nodes and the quality of data, while also considering cost and time constraints. The main steps of the proposed ONDS algorithm are outlined in Algorithm 1.

Finally, we prove that the proposed ONDB algorithm has a linear *worst-case* time complexity.

Property 1. *ONDB algorithm has a worst-case computational complexity of* $\mathcal{O}(N)$.

Using the optimization decomposition strategy, the proposed ONDB algorithm has only one loop, which runs at most N times. As for the amounts of data to be acquired from the selected nodes, x_n in Line 2, they can be pre-computed according to the considered learning, as shown in Section 5.3; thus, they do not influence the overall complexity.

It is also worth highlighting that Property 1 depicts the *worst-case* complexity, and the algorithm often performs even more efficiently in practice, thanks to the constraints defined in (8) and (9). Also, the ONDB algorithm is executed at most two times; once for initial values of x_n and m^*, and once for their optimized values.

6. Performance evaluation

In this section, we first present the simulation environment used to obtain our results. We will then proceed to evaluate the performance of the ONDB framework in comparison to state-of-the-art techniques. Our primary focus will be on assessing the performance of various learning paradigms and examining the impact of blockchain on this performance.

6.1. Simulation environment

For our performance evaluation, we used two datasets: *energy consumption* and *MNIST*. The *energy consumption* dataset, published by the energy company UK Power Networks in London [29], consists of fine-grained energy consumption readings from 5567 smart meters. These readings were reported every half-hour between November 2012 and February 2014. In the case of the FL and AFL setups, this dataset was divided among 5 separate entities to maintain a non-IID data distribution.

The *MNIST* dataset[1] is a well-known collection of handwritten digits, commonly used for training and testing various supervised ML algorithms. Each image in this dataset represents a handwritten digit, with pixel values corresponding to image fields, and labels indicating the digit types (ranging from 0 to 9). In the FL and AFL setups, this dataset was divided among ten separate entities, with each entity assigned 500 images from two different classes (as shown in Table 1). We employed the deep learning model described in [30] for training and testing, with batch size, local epochs, and learning rate set to 10, 5, and 0.01, respectively.

[1] https://www.kaggle.com/oddrationale/mnist-in-csv

Algorithm 1 Optimal Node, Data, and Blockchain Configuration (ONDB) Algorithm.

Input: p, T^{max}, τ_n, C^{max}, C_n, k_n, $x_n^* = x$, $m^* = m$, $t = 0$, $\lambda_n(t) = 0$.
Given x_n^*, calculate weighting coefficient δ_n for the available nodes.
Rank all nodes in descending order based on their weights δ_n.
while $n \leq N$ **do**
    Select and add a node with maximum weight to the selected set $\tilde{\mathcal{N}}$.
    if constraints (8) and (9) are satisfied. **then**
        Set $\lambda_n(t+1) = 1$, for the selected node n.
        Identify new classes that will be considered from this node $\tilde{k}_n$.
        Update weighting coefficients δ_n for the remaining nodes, based on the selected ones.
        Go to step 4.
    else
        Break while loop. ▷ $\lambda_n(t+1)$ are obtained.
    end if
end while
if $\lambda_n(t+1) = \lambda_n(t)$ **then**
    Break ▷ λ_n, x_n^*, m^* are obtained.
else
    if p = CL **then**
        $x_n^* = x_n^{CL}$; calculate x_n^{CL} using (24).
        Calculate m^* using (18).
    else if p = FL **then**
        $x_n^* = x_n^{FL}$; calculate x_n^{FL} using (23).
        Calculate m^* using (18).
    else
        $x_n^* = x_n^{AFL}$; calculate x_n^{AFL} using (25).
        Calculate m^* using (18).
    end if
end if
Go to step 3.
return λ_n, x_n^*, and m^*.

Table 1
Classes distribution of MNIST dataset at different entities.

Entity	1	2	3	4	5	6	7	8	9	10
Class	2,1	6,8	5,6	1,3	7,2	3,4	9,8	2,7	8,4	0,5

6.2. Results

We begin by examining the impact of the exchanged data size on the performance of different learning paradigms. Figs. 3, 4, and 5 illustrate how the Root Mean Square Error (RMSE) changes when predicting energy consumption one day ahead using CL. The results provide insights into how the RMSE varies with dataset size, considering *energy consumption* dataset. We also explore variations in data distribution, considering both IID and non-IID scenarios. As depicted, CL achieves better performance as the dataset size increases, especially in the case of IID data (see Fig. 3). However, this improvement comes at the cost of sharing a substantial amount of data while adhering to strict security constraints via the blockchain. This, in turn, leads to increased network load, latency, and costs.

To address these limitations, FL was introduced to reduce the network load by sharing only the local models. However, this reduction in network load is accompanied by a decrease in learning performance. As shown in Fig. 4, FL with non-IID data distribution only converges to the performance level of CL with non-IID data after 15 communication rounds, still falling short of the performance of CL with IID data.

To achieve convergence with the best performance, matching that of CL with IID data, AFL was introduced. As demonstrated in Fig. 4, AFL strikes a balance between learning performance and network load. It achieves this balance by exchanging a small portion of local datasets to maintain near-balanced datasets across various entities, significantly reducing the network load compared to CL. Similar behavior was observed when using the *MNIST* dataset, as depicted in Fig. 5. This figure illustrates the variations in classification accuracy with respect to the number of communication rounds while examining the impact of sharing different proportions of local data, specifically 10%, 20%, and 30%. As depicted, sharing data among different entities leads to a substantial improvement in classification accuracy compared to cases where data sharing was not employed, such as the conventional FL scheme. For example, when employing 10% data sharing, the required communication rounds to achieve an 80% accuracy level decreased

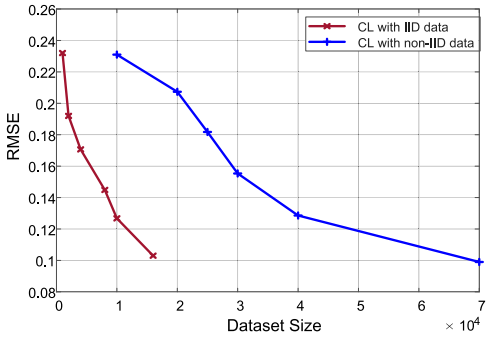


Fig. 3. The obtained RMSE with increasing dataset size, while considering the CL setup and energy consumption dataset.

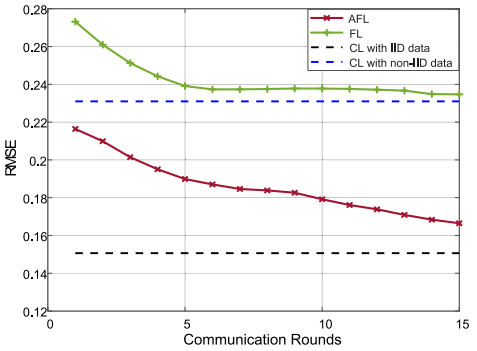


Fig. 4. RMSE variations with increasing communication rounds, while considering the energy consumption dataset along with FL and AFL setup. The dataset size is set to 10,000 readings.

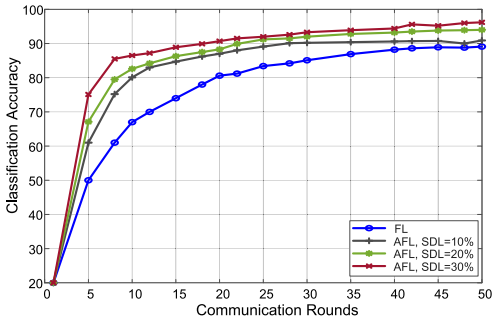


Fig. 5. The variations of classification accuracy as a function of the communication rounds and different shared data length (SDL), using MNIST dataset.

by 60%. This result is obtained because the introduction of new data at different entities helps mitigate the effects of non-IID data distribution and brings each entity closer to a more uniform data distribution.

The second aspect we investigate is how the proposed ONDB solution influences the performance of FL and AFL. In Fig. 6, we compare ONDB with state-of-the-art solutions, referred to as Fixed Node Selection (FNS) [26] and Select All (SA). The latter represents the conventional FL scheme, which involves all participating nodes in the learning process. The FNS scheme corresponds to the conventional FL scheme with a fixed number of entities selected for participation based on communication (or delay) performance, where nodes with minimum communication delay are chosen. As depicted in Fig. 6, within the FL setup, ONDB outperforms both the FNS and SA schemes. This superior performance can be attributed to ONDB's capability to judiciously select the optimal number of entities with the highest model and data quality, effectively reducing the impact of non-IID data distribution on FL performance while achieving rapid convergence. In contrast, FNS and SA either disregard the importance of model and data quality or involve a larger number of entities, resulting in performance degradation and necessitating an extensive number of communication rounds to reach peak performance.

Interestingly, the performance of the previously mentioned selection schemes exhibits variations in the context of the AFL setup (see Fig. 6). Thanks to the proposed AFL scheme, which facilitates the exchange of a small portion of data among participating entities to address the non-IID data challenge, both FNS and SA experience significant performance improvements.

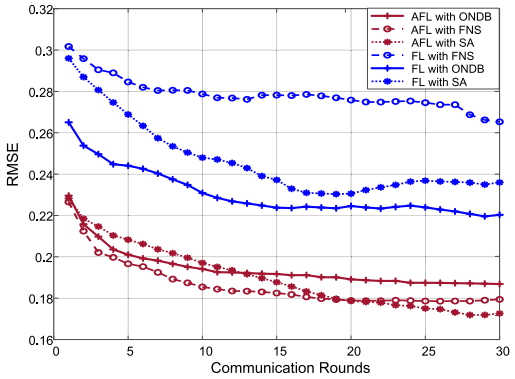


Fig. 6. RMSE variations versus communication rounds for energy consumption dataset, while considering FL and AFL with different node selection schemes.

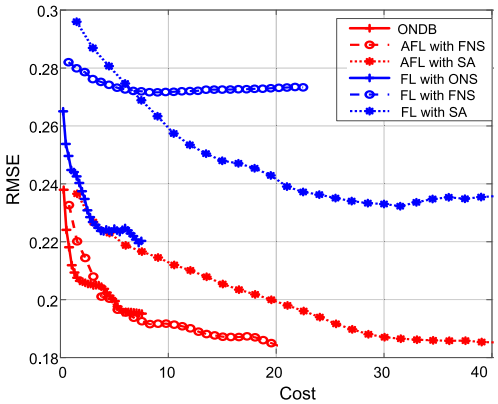


Fig. 7. RMSE variations versus communication rounds for energy consumption dataset, while considering FL and AFL with different node selection schemes.

In the AFL setting, involving a greater number of participating entities, as in FNS, can lead to performance enhancement, as the impact of non-IID data becomes less pronounced. However, this improvement is counterbalanced by increased costs and a higher blockchain network load and latency (see Fig. 7). On the other hand, SA schemes necessitate a substantial number of communication rounds to reach peak performance. On the contrary, ONDB obtains an optimal balance between learning performance, cost, and blockchain load. As shown in Fig. 7, ONDB solution could rapidly converge to the desired learning performance with the minimum cost. It achieves this by selecting the optimal number of entities that leads to the best performance and lowest cost. This number of selected entities is also significantly fewer than the number of entities chosen in FNS and SA. In Fig. 7, we assume a blockchain sharing cost of 0.05 per block.

Finally, we highlight that the proposed ONDB solution maintains stable performance and cost efficiency as the number of communication rounds and data requirements increase, validating its scalability. For instance, as shown in Fig. 6, ONDB achieves the best FL performance in only 15 communication rounds, whereas FNS and SA require significantly more rounds to approach a similar level of performance. This is further confirmed in Fig. 7, where ONDB converges to the best learning performance with minimal cost. This results in a cost reduction of approximately 71% compared to SA, highlighting ONDB's efficiency in selecting the optimal number of nodes. Accordingly, our results highlight ONDB's robustness and scalability in meeting the demands of large-scale smart grid deployments.

7. Conclusions

This paper introduces a scalable and Reconfigurable framework for secure, distributed collaborative learning using blockchains. The synergistic combination of distributed learning and blockchain technology presents a promising path to address the performance and security challenges in smart grids. By promoting collective efforts and ensuring data integrity, we can facilitate a more efficient and secure exchange of knowledge and insights among diverse smart grids' entities. The proposed SDL system integrates efficient distributed learning paradigms and pegged sidechains architecture to enable data and model exchange among various smart grid's entities. Different blockchain modes have been defined to meet the requirements of different services within the smart grid. The results indicate that the SDL system fulfills diverse QoS requirements while ensuring security and privacy.

Future work can include the exploration of various blockchain configuration parameters, such as block size, number of transactions per block, and transaction size, on learning performance. Optimizing these parameters may play a crucial role in achieving a balance between security, performance, and cost. For instance, adjusting block/transaction size can enhance the blockchain's behavior with different learning paradigms.

CRediT authorship contribution statement

Alaa Awad Abdellatif: Conceptualization, Design, Data collection, Analysis, Interpretation. **Khaled Shaban:** Conceptualization, Design, Interpretation. **Ahmed Massoud:** Conceptualization, Analysis, Interpretation.

Ethical considerations

The authors confirm that the study does not involve any human or animal subjects and complies with the ethical standards of research.

Confidentiality and intellectual property

The authors confirm that all the data and materials used in the manuscript are the authors' original work and have not been previously published or are under consideration for publication elsewhere. We trust that this manuscript and all associated materials have been prepared with the highest ethical standards, and we are committed to maintaining the integrity of the academic publishing process.

Funding disclosure

The research described in this manuscript did not receive any specific grant from funding agencies in the public, commercial, or not-for- profit sectors that would influence the outcome or interpretation of the findings.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was made possible by PDRA grant #PDRA7-0410-21004 from the Qatar National Research Fund (a member of Qatar Foundation). The findings achieved herein are solely the responsibility of the authors.

Data availability

The used data has been cited in the paper.

References

[1] 2020 smart grid system report. 2020, https://www.energy.gov/sites/default/files/2022-05/2020%20Smart%20Grid%20System%20Report_0.pdf [Accessed: 2024-11-12].
[2] Baccour E, Mhaisen N, Abdellatif AA, Erbad A, Mohamed A, Hamdi M, Guizani M. Pervasive AI for IoT applications: A survey on resource-efficient distributed artificial intelligence. IEEE Commun Surv Tutor 2022.
[3] Abdellatif AA, Al-Marridi AZ, Mohamed A, Erbad A, Chiasserini CF, Refaey A. ssHealth: Toward secure, blockchain-enabled healthcare systems. IEEE Netw 2020;34(4):312–9. http://dx.doi.org/10.1109/MNET.011.1900553.
[4] Abdellatif AA, Chiasserini CF, Malandrino F, Mohamed A, Erbad A. Active learning with noisy labelers for improving classification accuracy of connected vehicles. IEEE Trans Veh Technol 2021;70(4):3059–70.
[5] Heidari A, Jafari Navimipour N, Unal M. A secure intrusion detection platform using blockchain and radial basis function neural networks for internet of drones. IEEE Internet Things J 2023;10(10):8445–54. http://dx.doi.org/10.1109/JIOT.2023.3237661.
[6] Mu C, Ding T, Shahidehpour M, Liu S, Chen B, Jia W, Ying H, Huang Y. A light blockchain for behind-the-meter peer-to-peer energy transactions in cyber-physical power systems. IEEE Trans Smart Grid 2023;1. http://dx.doi.org/10.1109/TSG.2023.3265536.
[7] Badr MM, Mahmoud MMEA, Fang Y, Abdulaal M, Aljohani AJ, Alasmary W, Ibrahem MI. Privacy-preserving and communication-efficient energy prediction scheme based on federated learning for smart grids. IEEE Internet Things J 2023;10(9):7719–36. http://dx.doi.org/10.1109/JIOT.2022.3230586.
[8] Tkachuk R-V, Ilie D, Robert R, Kebande V, Tutschku K. Towards efficient privacy and trust in decentralized blockchain-based peer-to-peer renewable energy marketplace. Sustain Energy Grids Netw 2023;35:101146.
[9] Wang Y, Su Z, Ni J, Zhang N, Shen X. Blockchain-empowered space-air-ground integrated networks: Opportunities, challenges, and solutions. IEEE Commun Surv Tutor 2022;24(1):160–209. http://dx.doi.org/10.1109/COMST.2021.3131711.
[10] Yang J, Dai J, Gooi HB, Nguyen HD, Wang P. Hierarchical blockchain design for distributed control and energy trading within microgrids. IEEE Trans Smart Grid 2022.

[11] Yang W, Guan Z, Wu L, Du X, Guizani M. Secure data access control with fair accountability in smart grid data sharing: An edge blockchain approach. IEEE Internet Things J 2020;8(10):8632–43.
[12] Lu W, Ren Z, Xu J, Chen S. Edge blockchain assisted lightweight privacy-preserving data aggregation for smart grid. IEEE Trans Netw Serv Manag 2021;18(2):1246–59.
[13] Zanbouri K, Darbandi M, Nassr M, Heidari A, Navimipour NJ, Yalcın S. A GSO-based multi-objective technique for performance optimization of blockchain-based industrial internet of things. Int J Commun Syst 2024;37(15):e5886.
[14] Bouachir O, Aloqaily M, Özkasap Ö, Ali F. FederatedGrids: Federated learning and blockchain-assisted P2P energy sharing. IEEE Trans Green Commun Netw 2022;6(1):424–36.
[15] Veerasamy V, Sampath LPMI, Singh S, Nguyen HD, Gooi HB. Blockchain-based decentralized frequency control of microgrids using federated learning fractional-order recurrent neural network. IEEE Trans Smart Grid 2023;1. http://dx.doi.org/10.1109/TSG.2023.3267503.
[16] Wang T, Dong Z. Blockchain based clustered federated learning for non-intrusive load monitoring. IEEE Trans Smart Grid 2023;1. http://dx.doi.org/10.1109/TSG.2023.3326194.
[17] Su Z, Wang Y, Xu Q, Fei M, Tian Y-C, Zhang N. A secure charging scheme for electric vehicles with smart communities in energy blockchain. IEEE Internet Things J 2018;6(3):4601–13.
[18] Abdellatif AA, Shaban K, Massoud A. SDCL: A framework for secure, distributed, and collaborative learning in smart grids. IEEE Internet of Things Mag 2024;7(3):84–90. http://dx.doi.org/10.1109/IOTM.001.2300059.
[19] Wang S, Wang J, Wang X, Qiu T, Yuan Y, Ouyang L, Guo Y, Wang F-Y. Blockchain-powered parallel healthcare systems based on the ACP approach. IEEE Trans Comput Soc Syst 2018;99:1–9.
[20] Kang J, Xiong Z, Niyato D, Ye D, Kim DI, Zhao J. Toward secure blockchain-enabled internet of vehicles: Optimizing consensus management using reputation and contract theory. IEEE Trans Veh Technol 2019;68(3):2906–20.
[21] Abdellatif AA, Mhaisen N, Mohamed A, Erbad A, Guizani M, Dawy Z, Nasreddine W. Communication-efficient hierarchical federated learning for IoT heterogeneous systems with imbalanced data. Future Gener Comput Syst 2022;128:406–19.
[22] Abdellatif AA, Allahham MS, Khial N, Mohamed A, Erbad A, Shaban K. Reliable federated learning for age sensitive mobile edge computing systems. In: ICC - IEEE international conference on communications. 2023, p. 1622–7. http://dx.doi.org/10.1109/ICC45041.2023.10278789.
[23] Justus D, Brennan J, Bonner S, McGough AS. Predicting the computational cost of deep learning models. In: IEEE international conference on big data (big data). IEEE; 2018, p. 3873–82.
[24] Malandrino F, Chiasserini CF, Molner N, De La Oliva A. Network support for high-performance distributed machine learning. IEEE/ACM Trans Netw 2022;(01):1–15.
[25] Abdellatif AA, Chiasserini CF, Malandrino F. Active learning-based classification in automated connected vehicles. In: IEEE INFOCOM 2020-IEEE conference on computer communications workshops (INFOCOM WKSHPS). IEEE; 2020, p. 598–603.
[26] Mhaisen N, Abdellatif AA, Mohamed A, Erbad A, Guizani M. Optimal user-edge assignment in hierarchical federated learning based on statistical properties and network topology constraints. IEEE Trans Netw Sci Eng 2021;9(1):55–66.
[27] Shalaby S, Abdellatif AA, Al-Ali A, Mohamed A, Erbad A, Guizani M. Performance evaluation of hyperledger fabric. In: IEEE international conference on informatics, IoT, and enabling technologies (ICIoT). 2020, p. 608–13. http://dx.doi.org/10.1109/ICIoT48696.2020.9089614.
[28] Duan M, Liu D, Chen X, Liu R, Tan Y, Liang L. Self-balancing federated learning with global imbalanced data in mobile systems. IEEE Trans Parallel Distrib Syst 2020;32(1):59–71.
[29] Smart meter energy consumption data in London households. 2022, https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households [Accessed: 2022-08-25].
[30] Ji S. A pytorch implementation of federated learning. 2018, Zenodo. http://dx.doi.org/10.5281/zenodo.4321561.