

Enhancing Data Integrity in Federated Learning for Cloud Load Prediction Using Blockchain

MSc Research Project
MSc Cloud Computing

Abdullah Khan
Student ID: X23364971

School of Computing
National College of Ireland

Supervisor: Shreyas Setlur Arun

National College of Ireland
Project Submission Sheet
School of Computing



Student Name:	Abdullah Khan
Student ID:	X23364971
Programme:	MSc Cloud Computing
Year:	2024-2025
Module:	MSc Research Project
Supervisor:	Shreyas Setlur Arun
Submission Due Date:	11/12/2025
Project Title:	Enhancing Data Integrity in Federated Learning for Cloud Load Prediction Using Blockchain
Word Count:	7671
Page Count:	20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Abdullah Khan
Date:	6th December 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission , to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project , both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing Data Integrity in Federated Learning for Cloud Load Prediction Using Blockchain

Abdullah Khan
X23364971

Abstract

Federated learning (FL) allows decentralized machine learning and maintains privacy of the data, but it can be attacked by adversarial techniques, like model poisoning. This study suggests incorporating the use of blockchain technology and FL to help increase the integrity of data and model strength in the forecasting of cloud loads. A verification layer ensures that accountability and transparency are maintained via the creation of a blockchain-based involvement of model updates in a tamper-evident ledger. Tests on the system were carried out with clean state, attacked state, and attacked-verified state. The clean federated model performed better on the basis of Gradient Boosting Regressor ($MAE = 0.0898$, $RMSE = 0.1112$, $MAPE = 46.81\%$, $R^2 = 0.9507$). In the attacked model, the performance fell in a huge manner ($MAE = 0.2353$, $RMSE = 0.2934$, $MAPE = 85.70\%$, $R^2 = 0.6565$), whereas the attacked-verified model showed a recovery ($MAE = 0.0914$, $RMSE = 0.1173$, $MAPE = 53.30\%$, $R^2 = 0.9451$). Large datasets produced in the course of the training were stored and managed on AWS S3 with the help of which data synchronization became efficient, and their storage was also scaled. The results emphasize how the blockchain-verified FL framework can effectively preserve the accuracy and transparency of the model and reduce the adversarial risks, which were related to the creation of privacy-preserving and secure machine learning architectures.

1 Introduction

The surge in cloud computing and artificial intelligence that is distributed has led to a need to have valid, privacy guaranteeing, and tamper resistant load forecasting models. Predictive intelligence enables the distribution of energy, allocation of work, and the use of resources which is very crucial in dynamic environments to balance them (Xu et al.; 2021). Conventional centralized systems, though, come with the issue of data privacy and integrity, as well as regulatory flexibility (Aledhari et al.; 2020). Federated Learning (FL) solves those problems as it also allows the decentralization of model training without accessing raw data, enhancing privacy and scalability (Albshaier et al.; 2025).

However, FL has integrity and trust threats, such as malicious local updates manipulation, and this may compromise the accuracy of the global model, causing such failures as model poisoning and backdoor attacks and Byzantine attacks (Liang et al.; 2023). Such risks are particularly dangerous in the mission critical systems such as power and cloud management where load forecasting adversarial attacks can lead to the collapse of their operations (Shabbir et al.; 2024). The current aggregation models, such as Krum

and Median, are ineffective against collusive adversaries, and traditional FL lacks the necessary auditability, necessitating an update provenance mechanism, tampering detection mechanism, and verifying the authenticity of contributors (Fang et al.; 2025; Lo et al.; 2021).

A solution to this problem is the blockchain technology, which could guarantee transparency and tamper-proof collaboration through the creation of a decentralized immutable ledger to store and verify updates on the FL model (Mao et al.; 2024). It is possible to use cryptographic properties of blockchain as viewed through digital signatures and hashing to develop a reputation of a client and audit logs to eliminate manipulation of data (Jovanovic et al.; 2024). They are Smart contracts implemented to incentivize and validate more robust federated model training surroundings based on automated incentive/validation strategies encouraging a more solid federated model training setting (Kang et al.; 2019; Limbepe et al.; 2025; Shitharth et al.; 2023).

In this study, a blockchain-authenticated FL model is proposed to improve both the integrity of data and the resiliency of the models during cloud load forecasting. The system brings on board a validation layer that filters model updates according to normalization, validation loss and trust scores (Mao et al.; 2024; Shabbir et al.; 2024).

Accordingly, the research is guided by the following question:

Research Question: How effectively can a blockchain-verified federated learning framework mitigate accuracy degradation from poisoned clients while maintaining an auditable record of model update decisions across training rounds?

To answer this question, the research formulates the following objectives:

- **RO1:** To evaluate the verification layer by running clean, attacked, and attacked-verified modes and comparing RMSE/MAE/ R^2 across modes.
- **RO2:** To implement and demonstrate a tamper-evident ledger that logs round, client_id, verdict, reason, SHA-256 hash, and timestamp for every update.

This research aims to help the advancement of secure and explainable federated intelligence by incorporating blockchain-based trust validation in distributed learning (Jiang et al.; 2023; Mao et al.; 2024; Ning; 2024).

The rest of this research study is organized as follows:

Section	Description
Related Work	Introduces related works and theoretical background from which the work in this study is founded.
Methodology	Presents the proposed methodology and research design.
System Architecture	Describes the system architecture of the proposed framework.
Implementation	Provides a detailed explanation of how the code was implemented.
Evaluation	Discusses the results and effectiveness of the study, including accuracy.
Conclusion	Summarizes the study, results, contributions, and outlines future work.

Table 1: Organization of the Research Study

2 Related Work

2.1 Federated Learning and Privacy Challenges

Federated Learning (FL) allows training a distributed model without access to raw information, which is appropriate in the privacy-sensitive fields of medicine, finance, and power load prediction (Orabi et al.; 2025; Aledhari et al.; 2020). FL addresses privacy risks posed by centralized systems by distributing model updates, as opposed to the direct data distribution of raw data accessibility (Yurdem et al.; 2024). However, FL has difficulties with the data integrity and resistance to adversarial attack (Jeong et al.; 2025). The use of malicious clients also has an impact on performance, as well as privacy, by introducing corrupt updates that threaten the operation of the model, and the decentralization of FL makes it challenging to detect and remedy attacks by malicious clients (Pillutla et al.; 2019).

Existing aggregation techniques such as Krum and Median are being used to address basic attacks but fail to resist highly advanced threats, such as colluding adversarial clients of the aggregation system (Jeong et al.; 2025; Fang et al.; 2025). Besides, the approaches fail to confirm the truthfulness and legitimacy of updates, making the application of these tools to be limited in situations where traceability is valuable. In spite of these difficulties, FL is a promising strategy, but its security issues should be addressed before it can be massively deployed in high-income uses (Ning; 2024).

2.2 Blockchain as a Solution for Security and Provenance

Federated Learning (FL) systems have been suggested to improve the security and accountability of blockchain technology engineering systems (Arya; 2025). It provides a decentralized immutable registry to record transaction, including model updates, with transparency and tamper resistant properties (Yurdem et al.; 2024). Through blockchain storage of updates to the models, the integrity of contributions made by the client can be cryptographically verified, safeguarding against unwanted modifications (Kang et al.; 2019; Lo et al.; 2021). Also, blockchain produces an accountability trail to monitor the identity of updates, dates, and verdict, which helps to identify and address evil behaviors (Jovanovic et al.; 2024).

The benefits of using blockchain in FL include enhancements of model training transparency and decentralized consensus system enabling verification of updates to the model (Albshaier et al.; 2025). It facilitates responsibility and honesty among the participants as well (Saleh; 2024). However, integration also comes with certain challenges, such as the redirected computational expenses as a result of cryptographic operations and maintenance of the distributed ledger (Li et al.; 2022). Also, scalability does not work well especially on large systems with a large number of clients (Shitharth et al.; 2023).

2.3 Adversarial Attacks and Defenses in Federated Learning

Federated Learning (FL) is vulnerable to severe risks of data and model poisoning (Tang et al.; 2024). These attacks are targeted to diminish the international model by adding fraudulent updates to it to decrease its accuracy and strength (Jeong et al.; 2025). The former is information poisoning or when the bad clients present bad data, whereas the latter is model poisoning or the transfer of corrupted gradients or weights to the training process (Pillutla et al.; 2019). These attacks can be reduced with existing defense, which

includes strong methods of aggregation and anomaly detection, based on filtering with statistics or modeling behavior analysis (Li et al.; 2023).

Nevertheless, they are frequently unsuccessful against sophisticated or organized attacks since they suppose that the majority of updates are legitimate, which does not hold in the hostile context with the high involvement of attackers in the process of actions (Sikandar et al.; 2023). Whereas techniques such as Krum and Trimmed Mean are useful in a small scale, they fail in terms of large-scale adversarial threat online predators (Jeong et al.; 2025). More sophisticated methods, e.g. detecting anomalies with deep learning, are promising but computationally intensive and cannot be applied to large systems. Thus, even though the defense against adversarial attacks in FL has been improved, there is still an issue of balancing robustness, scalability, and system performance without further impairment of adversarial defense in a FL system (Li et al.; 2022).

2.4 Blockchain-Driven Incentive Mechanisms and Trust Management in Federated Learning

One of the major problems with Federated Learning (FL) is the difficulty in encouraging clients to be truthful and provide a genuine update to it (Li et al.; 2023). Ancient FL systems do not have any mechanisms that can be used to provide clients with an incentive to work in the best interest of the system, and this results in free-riding or other malicious behavior of the client (Sikandar et al.; 2023). Incentive systems that are based on blockchains have been suggested to promote good behavior and discourage fraudsters (Kang et al.; 2019). The reward distribution can be automated by smart contracts, so clients are encouraged to provide valid updates to receive rewards (Lo et al.; 2021).

Nevertheless, blockchain-inspired incentives are also associated with problems such as the inefficiency of the execution of smart contracts, complexity of blockchain registry in resource-limited settings (Ouadrhiri and Abdelhadi; 2022). Also, it is not an easy task to design a fair and effective incentive system, as it demands a trade-off between rewards and penalties to promote honest behavior without undesirable consequences (Li et al.; 2022). Thus, although incentives based on blockchains can be beneficial to FL security, they should be implemented cautiously in order to guarantee.

2.5 Applications in Critical Infrastructure and Healthcare

In critical infrastructure, like energy and healthcare, Federated Learning (FL) and blockchain can play important roles (Nguyen et al.; 2021). FL can be applied to power load forecasting by having several grid operators collaborate in the definition of energy demand without having to affect the privacy of individual information. This is further advanced by blockchain which validates the updates in order to add only authenticated updates to the global model which enhances accurate prediction and reliability in predictions made by the model (Limbepe et al.; 2025). It is especially applicable to smart grids, where real time observability of load forecasting is essential to optimizing the distribution of energy and preventing systems collapse cases (Mao et al.; 2024).

Medicine FL with blockchain may be applied to teach models on sensitive information about patients across hospitals without infringing privacy. Blockchain also provides integrity and impossibility of manipulation of updates of the model (Limbepe et al.; 2025). However, its usage is hampered by factors like the need to integrate blockchain with the

current set of medical systems and the high computational costs of FL and blockchain (Zhang et al.; 2022). Irrespective of such difficulties, this holds promise of improving privacy, security and accountability in healthcare and energy systems, so this would be a crucial focal point of prospective research.

Although there has been improvement on the adoption of blockchain in FL, there are still gaps. Even the latest aggregation practices in FL are susceptible to high-level adversarial attacks, and more powerful defense strategies are needed (Fang et al.; 2025). Also, the computation cost of blockchain and scalability of blockchain-based FL systems should be addressed further as well (Li et al.; 2022; Kang et al.; 2019). The following table shows the systematic literature review as;

Title	Models Used	Limitations	How the Proposed Study Will Cover
Federated Learning: A Survey on Enabling Technologies, Protocols, and Applications	Federated Learning, Collaborative AI, P2P Networks	Does not address how federated learning can be integrated with blockchain for security enhancements	The proposed study will integrate blockchain to provide secure, privacy-preserving federated learning systems.
Blockchain-Based Privacy-Enhancing Federated Learning in Smart Healthcare: A Survey	Blockchain, Federated Learning	Focuses only on privacy in healthcare	The proposed study will extend the idea to other industries, such as cloud-load forecasting, and enhance transparency through blockchain auditability.
Federated Learning Optimization: A Computational Blockchain Process with Offloading Analysis	Federated Learning, Blockchain Process, Optimization	Limited practical implementation details	The proposed study will implement the blockchain-based optimization process in a real-world decentralized load forecasting application.
BGFL: A Blockchain-Enabled Group Federated Learning at Wireless Industrial Edges	Blockchain, Federated Learning	Lacks a clear verification mechanism for model updates	This study will incorporate a verification layer that filters poisoned updates using trust-based criteria and loss validation before aggregation.

Title	Models Used	Limitations	How the Proposed Study Will Cover
A Blockchain-Based Framework for Federated Learning with Privacy Preservation in Power Load Forecasting	Blockchain, Federated Learning, Power Load Forecasting	Limited to privacy preservation, no emphasis on model robustness or attack mitigation	The proposed study will add a tamper-evident blockchain ledger for secure and transparent aggregation of federated updates, improving model robustness.

3 Methodology

The study follows a quantitative experimental research design as shown in Figure 1 in order to examine the claim that the inclusion of a verification-and-ledger layer can maintain the performance of Federated Learning (FL) models in the presence of data integrity attacks. The research contains three controlled regimes namely: (i) *Clean FL*, (ii) *Attacked FL*, and (iii) *Attacked + Verified FL*, all of which train using the same dataset, model architecture and hyperparameters. RMSE, MAE, MAPE, and R^2 are used to evaluate performance and robustness measures are used to assess change in accuracy compared to the clean condition. All the experiment, logs and ledger records are saved into disk in order to be reproducible and auditable.

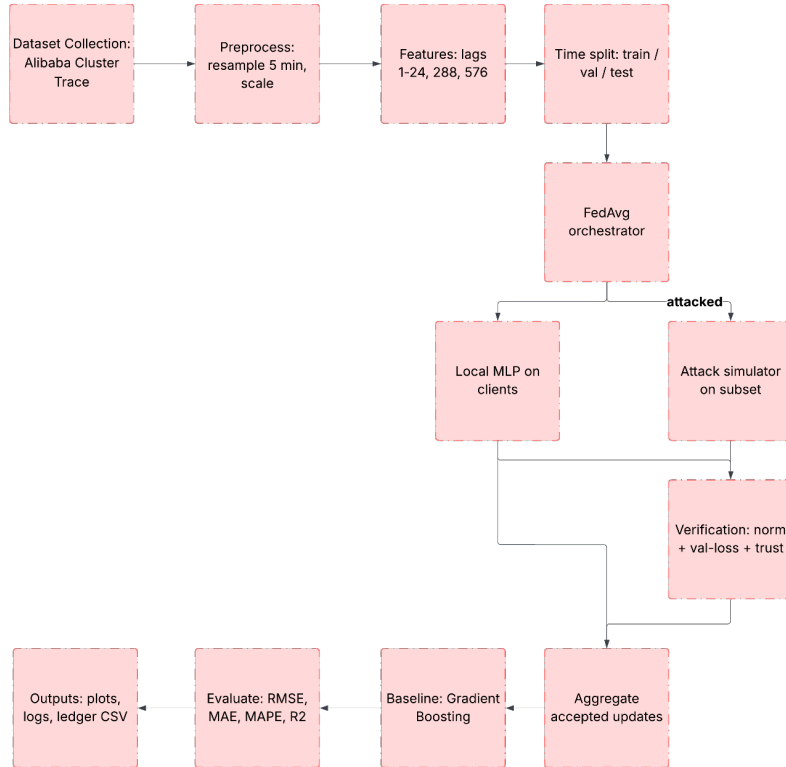


Figure 1: Proposed Methodology for FL with Blockchain Layer

3.1 Source of the Data and Cleaning the Data

The dataset used is the Alibaba Cluster Trace (GPU v2020), obtained from Kaggle. It has usage logs of thousands of GPU-enabled machines running in a production cloud environment. The raw data is being processed in a systematic way. Timestamps are changed to UTC to attain time standardization and machine-level time series are extracted. The resampling of the data is done at a time period of 5 minutes, with the gaps filled in short sequences to maintain continuity. A single load indicator is made up of the normalization and average of utilization values (e.g., CPU, GPU, memory load) on the scale of $[0, 1]$. Time-aware splits (Train / Validation / Test) are used in order to prevent the leakage across time. Machines with either too short or too many missing data are eliminated. This is to make sure that the dataset is clean, uniformly sampled, and can be used in federated time-series prediction.

3.2 Feature Engineering

In feature engineering, attributes with little or no useful value are discarded and condensed into a smaller subgroup to model short-term dynamics, daily periodic variations and fine-grain variability of local resource demands. The last feature set is short-term lags (1 to 24 past timesteps), daily and bi-daily periodic lags (288, 576 timesteps of 24-hour and 48-hour cycles), rolling (mean and standard deviation window) statistics, calendar encodings (hour of day and day of week with sine/cosine transformations), and one-step-ahead alignment of the forecast (features and target shifted to predict at time $t+1$). This structured feature set enables both short scale trend identification and periodic load cycle modeling.

3.3 Client Formation and Sampling

Cloud machines are treated to behave as separate FL clients so that decentralized and non-IID conditions of data can be maintained in a natural manner. The training is to be configured in the following way. There is a set of machines with a fixed number (e.g., 12) which constitute the total FL client set. The number of clients picked to take part in a federated round is about two-thirds in every round. Sampling randomness is seeded and recorded so that it can be reproduced. Such an arrangement resembles actual operational federated settings, in which there is varied participation of clients.

3.4 Model Training

There are two models that are trained for comparison:

Centralized Baseline Model (GBR):

- A Gradient Boosting Regressor is trained on concatenated data. It can be used as a complete-data benchmark to put performance of the FL systems into perspective.

Federated Learning Model (MLP):

- Each picked client locally trains a compact MLP on its data. Local model aggregation is through sample-weighted FedAvg. Learning rate, batch size, and epochs are fixed across clients.

This enables to determine the impact of federated decentralization on the accuracy of the forecast.

3.5 Threat Model (Label Poisoning Attack)

The attackers are assigned a fraction of the clients to simulate adversarial behavior. Their local target labels are intentionally corrupted with scaling and random noises. The intensity and fraction of the attack are held constant during each run of the experiment. The attack resembles malicious reporting systems manipulation distorting the global model training.

3.6 Verification Layer

The verification mechanism is only activated in the state *Attacked + Verified*. The process of every client update is through three filters:

- Norm-Gate Check: Rejects updates with an abnormally high parameter shift as compared to other clients.
- Validation-Loss Gate: Obstacles updates that aggravate verifying performance after an adherence threshold.
- Rolling Trust Score: The clients have trust scores, and they decrease when they are rejected and increase when they are accepted.

t is only an update of models that passes all gates that add to global aggregation.

3.7 Tamper-Evident Ledger

A ledger database that draws inspiration of a blockchain keeps track of provenance. Each round logs shows client ID, acceptance verdict, rejection reason, timestamp, and SHA-256 update hash. Stored as CSV (and optionally JSON) for integrity, transparency, and audit trails. This can be detected by manipulation or retrospective tampering.

3.8 Training Procedure and Results Storage

The training process involves predefined federated rounds. After each round, the number of selected and accepted clients is logged, and validation metrics are computed and stored. Upon training completion, baseline metrics are saved in a dedicated output directory, FL log files for each regime are stored, and ledger files are generated to report all verification results.

3.9 Evaluation Metrics

In evaluating forecasting performance across experimental regimes (Clean, Attacked, and Attacked + Verified), the following regression metrics are calculated: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and Coefficient of Determination (R^2). Let y_t denote the true load at time t , and $\hat{y}_t$ denote the model prediction at time t .

4 Design Specification

The system proposed combines a federated neural forecasting model together with a verification-and-ledger trust layer to achieve resistance to data integrity attacks in decentralized learning systems. The architecture divides into three main parts, namely: (i) data and feature processing pipeline, (ii) federated model training based on adversarial simulation, verification constraints, and (iii) evaluation and robustness analysis. Figure 2 describes the interactions between components.

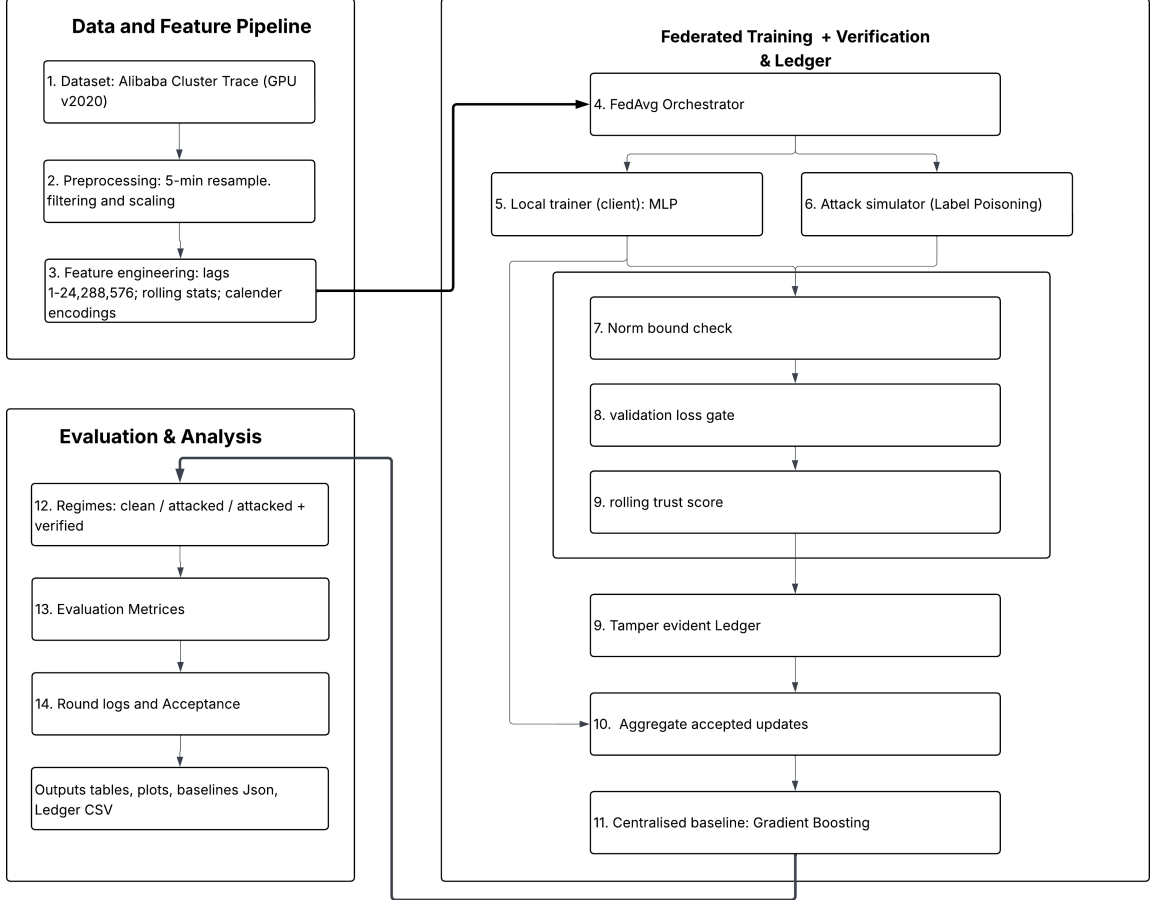


Figure 2: Proposed Architectural Design for FL with Blockchain Layer

4.1 Data and Feature Pipeline

The system uses the Alibaba Cluster Trace (GPU 2020) that has timestamped utilization samples on a sizable group of compute nodes. The raw telemetry stream of each machine is then chronologically sorted and resampled to a consistent 5-minute time interval in order to remove clock skew and reporting anomalies. Missing gaps (at the level of ≤ 20 minutes) are forward-filled to retain signal continuity, whereas longer gaps cause machine exclusion to impose time continuity.

There are several feature transformations, which are used to produce a supervised forecasting dataset. One-step to 24-step lagged embedded time dependence One-step to 24-step lagged embeddings are used to capture short-range temporal dependence. Long range periodicity is coded by lag offsets of 288 and 576 steps, respectively the lag of recycling 24-hours and 48-hours respectively at a sampling rate of 5 minutes. Rolling

mean and variance filters are used to include the local trend dynamics. Sinecosine positional encodings are used to encode temporal cyclical trends (hour-of-day and day-of-week modulations) to eliminate discontinuities of integer categorical values.

A global StandardScaler is computed on uniformly sampled data of features in all client datasets in order to provide cross client comparability to federated optimization. The scaler is sent to all the participants so that the distributions of feature magnitudes match and gradient instability caused by unequal scaling cannot occur. The forecastable variable is as follows:

$$y_{t+1} = \text{load}(t + 1),$$

and is moved as a result to prevent information leakage during training.

4.2 Federated Training and Verification Ledger

The federated training is fed with the FedAvg paradigm: the central coordinator has the global model, which they distribute to a random sample of clients during each communication round. Every client uses a small multilayer perceptron (MLP) with the following set-up on its local data to run stochastic gradient descent on E epochs.

MLP: $\mathbb{R}^d \rightarrow \mathbb{R}^1$, with hidden layers $[H, H/2]$, ReLU activations, and dropout p .

It is expressed through the formula client update:

$$\Delta W_i^{(r)} = W_i^{(r)} - W^{(r-1)},$$

where $W_i^{(r)}$ is the locally trained model at round r . In regimes that imitate adversarial behavior, the labels of selective clients are perturbed by:

$$\tilde{y} = \alpha y + \epsilon,$$

where α represents scaling distortion factor and ϵ represents Gaussian noise, and hence targets targeted integrity violation.

In order to avoid the corrupted updates affecting the global model, the regime of attained and verified imposes the policy of verification in three stages:

1. **Norm Gate:** Magnitude of update is restricted using a cohort relative bound:

$$\|\Delta W_i^{(r)}\|_2 \leq \tau \cdot \text{median}_j(\|\Delta W_j^{(r)}\|_2),$$

where τ is a deviation threshold that is adapted.

2. **Validation Loss Gate:** Global consistency is used by refusing to accept updates that worsen predictive performance on local validation partitions:

$$\ell_{\text{val}}(W_i^{(r)}) - \ell_{\text{val}}(W^{(r-1)}) > \delta,$$

where δ is a tolerance hyperparameter.

3. **Trust Score Dynamics:** Each client i maintains a scalar trust value:

$$T_i^{(r+1)} = \begin{cases} T_i^{(r)} + \eta(1 - T_i^{(r)}) & \text{if accepted,} \\ (1 - \gamma)T_i^{(r)} & \text{if rejected,} \end{cases}$$

where η controls recovery and γ governs decay. Updates from clients where $T_i^{(r)} < T_{\min}$ are automatically discarded.

Sample-weighted averaging is only used to combine updates that satisfy the following three gates:

$$W^{(r)} = \frac{\sum_{i \in \mathcal{A}^{(r)}} n_i W_i^{(r)}}{\sum_{i \in \mathcal{A}^{(r)}} n_i},$$

where, the accepted clients are denoted by the set $\mathcal{A}^{(r)}$, and n_i the respective local dataset size.

To be able to audit, every verification decision is attached to a tamper-evident ledger. Every record of a ledger contains: `round`, `client_id`, `verdict`, `reason_code`, a SHA-256 hash of the parameters submitted and a UTC time. The ledger acts as a blockchain-like integrity trail that can be applied to be post-hoc accountable in addition to forensically examining adversarial behaviors.

4.3 Evaluation and Analysis

A held-out global test set is used to estimate model performance in terms of four forecasting metrics:

$$\begin{aligned} \text{MAE} &= \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|, & \text{RMSE} &= \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}, \\ \text{MAPE} &= \frac{100}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right|, & R^2 &= 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}. \end{aligned}$$

The statistics of ledger, including the acceptance rates and the rejection causes distribution are examined and used to describe the stabilization of the trust and the adversarial suppression with time. All the outputs of the experiments (baselines, round logs, aggregated metrics, ledger files and trained model checkpoints) are archived to enable reproducibility, model reloading and downstream comparative benchmarking.

5 Implementation

5.1 Data Processing and Preprocessing

The data employed in the present research is the Alibaba Cluster Trace (GPU v2020), the data contains time-series information about machine performance (CPU, GPU, and memory usage). The preprocessing facility begins by changing timestamps into the UTC condition and remapping to 5 minutes intervals. Forward filling is used to fill up short gaps to avoid the gaps between data.

Different measures of utilization (e.g., CPU I/O wait, GPU utilization, etc.) are interpolated and scaled between $[0, 1]$ to generate one machine load target. Incomplete or excessive missing data after resampling are eliminated to ensure integrity of the data in the machines.

The data will be divided into training, validation, and test and attention will be taken to prevent data leakage. Lag features (1-24 steps of short-term lag, 288 and 576 steps of daily lag) are constructed and rolling mean and standard deviation computed and calendar features encoded based on sine and cosine transformations to capture periodicity.

5.1.1 Feature Engineering

In the case of feature engineering, multiple lag features are developed based on the past values of the load target to forecast the future values. These characteristics consist of short-term (1-24 steps) and long-term (288 and 576 steps) lags. Also, the mean and standard deviation data are rolled on a 24 time-steps window to determine short-term trends and variations. The hour of the day, day of the week and calendar-based feature e.g. hour of the day are converted into the model as sine and cosine coefficients to maintain periodicity. All characteristics are standardized through the use of standard normalization to have similar input in the models.

5.2 Federated Training and Verification Ledger

5.2.1 Federated Learning Setup

The federated learning (FL) system is built by training a series of local models on each client (machines) and a global one which is aggregated by Federated Averaging (FedAvg). The model of every client is a Multi-Layer Perceptron (MLP) and it is trained with the help of local data and then aggregated to compose the global model. Clients are randomly chosen in every round of training, and the global model weighting is computed using the weighted mean of the local models weighting.

When dealing with malicious clients trying to make updates with poison, a verification layer is added. It has three gates that it uses to filter updates: update norm bounded (rejects updates with large parameter changes), validation loss gate (rejects updates that grow validation loss) and rolling trust score (rejects updates by clients with low trust). This is to make sure that only valid updates are added to the global model and avoid the adverse impacts of adversarial behavior.

5.2.2 Verification Layer

The integrity of the federated learning system is very important to the verification layer. It implements three filters for validation checks on malicious/ erroneous updates. Update-norm Bounded gate tests whether the norm of the change in parameters is greater than a specified threshold, which is an indicator of an outlier update. Validation-loss Gate is a comparison of the pre and post update of the validation loss. In case the loss on post-update is more than a certain tolerance, the update is rejected. Rolling Trust Score shows each client has a trust score, which decays during time should the client make rejected updates. Clients who have below a minimum score in the trust scale are barred to contribute updates.

5.2.3 Tamper-Evident Ledger

A ledger that is tamper-evident is used to ensure transparency and traceability. This ledger lists all the updates done by all clients, the round number, client ID, possible reason for rejection (none possible), a hash of the update using the SHA-256 function, and a time stamp. Such a ledger is in CSV format that can be easily audited and verified on the status of every update. The ledger gives a permanent history which can be looked at to trace the process of decision making in the federated learning system.

5.3 Evaluation and Analysis

The analysis results show that the proposed verification layer is effective in improving the model with regard to adversarial attacks. In case of the baseline model that is trained on Gradient Boosting Regressor (GBR), the test measures are as follows: RMSE = 0.2517, MAE = 0.2038, MAPE = 21.65%, and $R^2 = 0.1483$. In the federated learning arrangement, the outcome is much better. In the case of Clean Federated Model, the evaluation metrics belong to RMSE = 0.1112, MAE = 0.0898, MAPE = 46.81%, and $R^2 = 0.9507$. Nevertheless, when RMSE = 0.2934, MAE = 0.2353, MAPE = 85.70% and $R^2 = 0.6565$, the performance is low, which pertains to the Attacked Federated Model. The performance rates again after verifying the layer in the Attacked + Verified Federated Model with RMSE = 0.1173, MAE = 0.0914, MAPE = 53.30 percent, and $R^2 = 0.9451$.

The comparison indicates that the increase of the RMSE of the attacked model is very large (163.95% increase relative to the clean model) and this means that the federated learning systems are susceptible to adversarial attacks. The attacked and checked model however, appears to be significantly improved with the RMSE increase lowered to only 5.52% as compared to the clean model. This is the affirmation that the verification layer will be effective in the reduction of the impacts of attacks.

5.4 AWS S3 Integration for Data Storage and Management

In this research, Amazon S3 was selected with the aim of using it as the main storage facility to handle the data produced during the federated learning process. An S3 bucket was also created in the Region of eu-north-1 (Stockholm), where all the experiment output, model files, logs, and performance measurements are safely maintained. In order to achieve controlled access, an AWS-Identity and Access Management (IAM) user and role were to be set with limited permissions at most, such that only the authorized parties would be able to access the storage. AWS SigV4 authentication allows access to a programmatic access so that each and every access interaction with the S3 bucket is signed and authenticated.

Automation of data was introduced in order to have sell and uniform uploads to the S3 bucket by the local environment. It is implemented with AWS SDK (boto3) to store the files recursively keeping the original folder hierarchy. The data of each experiment is sorted by unique prefixes with UTC times, which provides a proper distinction between consecutive experiments. Once the data has been uploaded, the integrity of the data is checked by listing the objects on the bucket and the data has been successfully persisted. The integration offers scalable, long-lasting, and reliable system of handling the high volumes of data produced during the federated learning experiments.

6 Evaluation

The purpose of this section is to provide a comprehensive analysis of the results and main findings of the study, as well as the implications of these findings, both from academic and practitioner perspectives. These findings are discussed to evaluate the performance of federated learning (FL) on various regimes, clean state, attacked state, and verified (with attacks) state. These models are also compared with the baseline Gradient Boosting Regressor (GBR) in the course of the evaluation to get knowledge on the effect of adversarial attacks and verification layer on the performance of models. The analysis

metrics applied in this analysis are RMSE, MAE, MAPE, as well as R^2 , these are also frequently used to assess the performance of machine learning models when it comes to forecasting tasks. The experimental results are presented using visual aids such as graphs, plots, and tables to support the analysis.

6.1 Experiment 1: Baseline Model Performance

This study used centralized Gradient Boosting Regressor (GBR) as the baseline model which was trained on the globally aggregated dataset. It was used to compare the performance of the federated learning models in not only clean, but also attacked and attacked + verified cases.

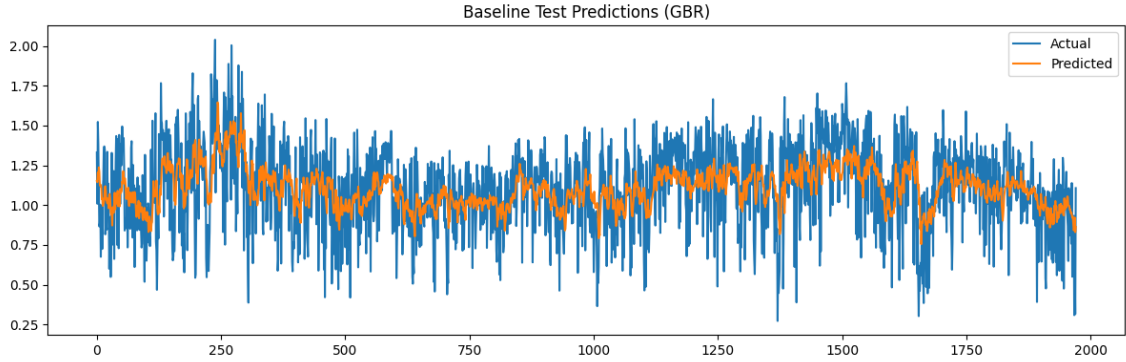


Figure 3: Baseline Test Prediction

The baseline model’s performance was assessed using RMSE, MAE, MAPE, and R^2 . The RMSE of 0.2517 and MAE of 0.2038 indicate moderate prediction error. The MAPE represents normal errors in making time-series predictions at 21.65%. The low R^2 value of 0.1483 is used as an indicator to compare more advanced models. Although the performance of the baseline model cannot be regarded as particularly accurate, it can be used as an excellent guideline to gauge the performance of federated learning models, especially when it comes to processing decentralized data and possible adversarial attacks. The performance of the baseline sets the stage of the knowledge of the improvements that federated models provide and the role of the verification layer in mitigation of data.

6.2 Experiment 2: Clean Federated Model Performance

The model was trained without any adversarial attack in the clean federated learning configuration, and only the valid data of the involved clients were used. The findings demonstrate that the federated learning capabilities of early-stage data processing and privacy protection are strong which is reflected by the improvements that are made to baseline Gradient Boosting Regressor (GBR).

The clean federated model results are: RMSE of 0.1112, MAE of 0.0898, MAPE of 46.81%, and $R^2 = 0.9507$. These findings reveal that federated learning works in ensuring high accuracy through the utilization of decentralized data. The significant decrease in RMSE and MAE implies that federated learning is able to effectively model the intricate trends of the data, which gives a significant enhancement of the baseline model.

Such results indicate that federated learning is able to make privacy-preserving predictions without affecting model performance. The better R^2 of clean federated model

further confirms that it is strong enough to manipulate a wide range of data sources as well as, guarantee predictive power is not compromised in a decentralized setting.

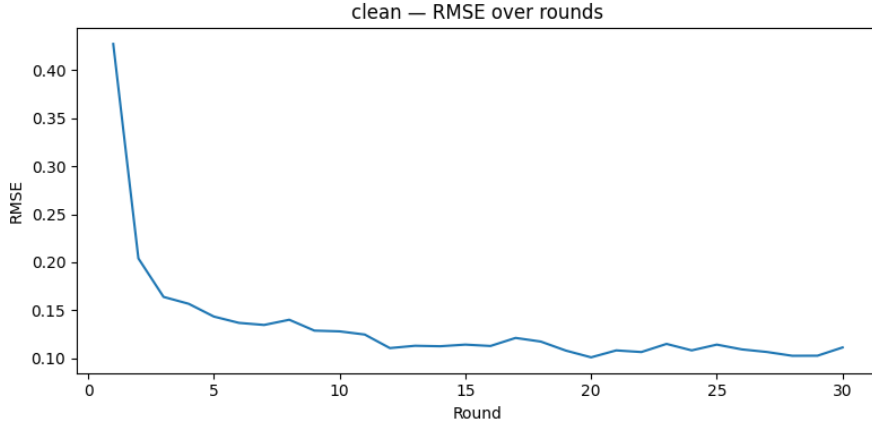


Figure 4: Clean Federated Model – RMSE Over Rounds

6.3 Experiment 3: Attacked Federated Model Performance

In the federated learning under attack, adversarial attacks are introduced intentionally into the training itself, to recreate the existence of non-beneficial clients. The objective of these attacks was to interfere with the performance of the model by feeding it with faulty or inaccurate data thereby interfering with the overall predictive accuracy. The following are the performance measures of the attacked federated model: RMSE of 0.2934, MAE of 0.2353, MAPE of 85.70%, and R^2 of 0.6565.

The significant growth in RMSE and MAE indicates that federated learning is susceptible to data integrity attacks. It was found that the model could not make predictions with good accuracy, which highlights why it is so difficult to sustain high model performance when dealing with adversarial clients. The high MAPE also shows the inaccuracy of forecasting being minimal in the malicious conditions. These findings highlight the importance of having strong anti-attack mechanisms to protect against such attacks in federated learning environments.

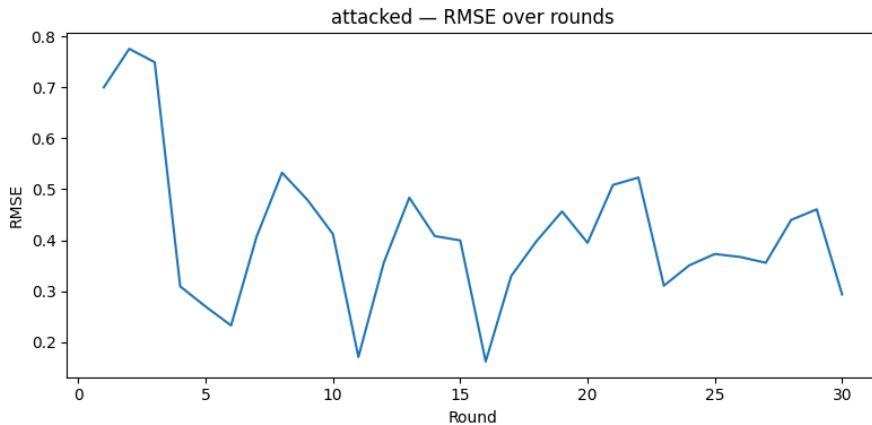


Figure 5: Attacked Federated Model – RMSE Over Rounds

6.4 Experiment 4: Attacked + Verified Federated Model Performance

The vulnerability that were revealed in the attacked federated model was corrected by introducing a verification layer which would filter the potentially malicious updates before aggregation. This layer provides a protective mechanism to meet only legitimate updates are added to the global model. The performance metrics of the attacked + verified federated model are as follows: RMSE of 0.1173, MAE of 0.0914, MAPE of 53.30%, and R^2 of 0.9451.

The attacked + verified model was significantly improved as compared to the attacked federated model. The verification layer reduced the effects of adversarial attack significantly, putting the performance of the model closer to that of the clean federated model. The decrease of RMSE and MAE proves the efficiency of the verification mechanism to avoid the corruption of the model accuracy in case of malicious updates.

These findings highlight the usefulness of verification layer in selecting model integrity and enhancing the robustness of federated learning especially in adversarial aspects. The enhancement of the performance of the verified model justifies the significance of the verification mechanisms adopted in federated systems because they are essential in ensuring that predictions made by them are reliable even when potentially malicious data contributions exist.

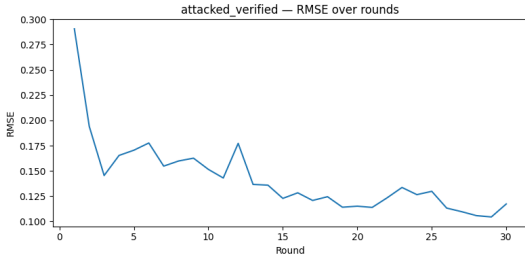


Figure 6: Attacked + Verified Federated Model – RMSE Over Rounds



Figure 7: Attacked + Verified Federated Model – Accepted Client Updates

6.5 Experiment 5: AWS S3 Integration and Evaluation

The main aim of this experiment was to determine the efficiency of the Amazon Web Services (AWS) S3 integration with the purpose of storing and managing the results of the federated learning process. The experiment was to identify the level on which S3 was effective in managing large sets of data created in the training process in such a way that data would be persisted, can be accessed securely and these data would be easily retrieved. AWS S3 was selected as it is scalable and durable, has the capability to store large volumes of logs, model files and training outputs in a centralized location. The integration was also put to test by uploading model artifacts and assessing the time required it takes to make data transfers between the local compute environment and the cloud so that the synchronization becomes timely and sustainable.

The test also evaluated the level of security of data management by using AWS Identity and Access Management (IAM) roles and applying SigV4 authentication. This guaranteed that any communication with the S3 bucket was signed with high security and only authorized parties could get access. Besides, the efficacy of the organization strategy, prefixing the UTC timestamp in the beginning of each experiment, was tested to ensure

clear division and convenient access to the results stored. Verification of each experiment was performed after the experiment to ensure that all the data was uploaded and intact and object listing was done to confirm the presence and integrity of all the files. The findings proved that AWS S3 was used to increase data management and also offered a secure, scalable and reliable solution in managing the vast amount of data generated throughout the federated learning experiments.

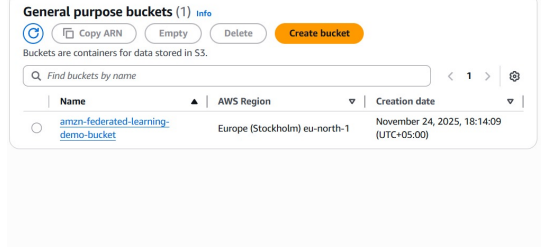


Figure 8: Stored Experiment Outputs in AWS S3 bucket

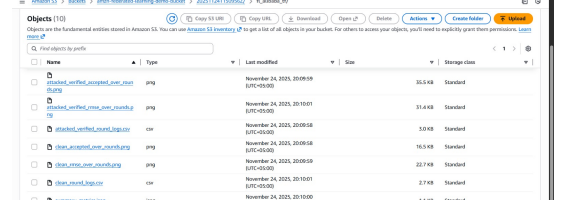


Figure 9: AWS S3 bucket Created for Federated Learning Data

6.6 Discussion

The results of the experiment indicate the usefulness of Federated Learning (FL) in load forecasting, namely in the aspects of data integrity and adversarial resistance. The clean, attacked, and attacked + verified models have their strengths and weaknesses as discussed when comparing the models. The clean federated model performed better than the baseline model with less RMSE of 0.1112 (compared to 0.2517) which shows that FL can manage decentralized load forecasting tasks without introducing malicious interference. The clean federated model also had less MAE of 0.0898 than the 0.2038 of the baseline demonstrating that FL can give better predictions but not affect privacy. A value of R^2 of 0.9507 indicates that this model is likely to be applicable to unseen data and this fact demonstrates the ability of FL to learn various complex patterns in time series.

These findings are in line with earlier literature, such as Xu et al. (2021), which demonstrated FL's superior performance over centralized models in privacy-sensitive tasks, and Yurdem et al. (2024), which showed FL's scalability in decentralized datasets. FL enables the identical performance of the centralized models and maintains the privacy of data, which is essential in the area of cloud computing and energy prognostics. On the contrary, the federated model attacked demonstrated a large performance drop, and RMSE and MAE were 0.2934 and 0.2353, respectively. The large MAPE of 85.70% each time proves that FL does not resist adversarial attacks, in which harmful customers have the opportunity to poison the global model by uploading malicious updates. The value of $R^2 = 0.6565$ does not indicate good predictability.

This reduction in performance can be compared to such works as Li et al. (2023) and Fang et al. (2025), which emphasize the vulnerability of FL to poisoning attacks, so effective defense mechanisms against the malicious interference must be developed. A much greater improvement was notable on the attacked + verified federated model, and by adding a verification layer, RMSE was lowered to 0.1173, and MAE was lowered to 0.0914. The R^2 of 0.9451 is approximated to that of the clean federated model implying that the layer of verification can be utilised to reduce adversarial effects effectively. Even the MAPE of 53.30% is still better than the clean model of 46.81%, by a considerable

margin compared to the attacked model of 85.70%. This can be verified by literature such as Shitharth et al. (2023), which demonstrated that adversarial attacks can be guarded against by verification mechanisms such as trust scores and validation-loss gates to secure FL systems. Also, in reference to the paper by Jovanovic et al. (2024) and Kang et al. (2019), the use of blockchain and security systems to enhance the stability and functionality of federated models is highlighted, particularly when dealing with adversarial contexts.

7 Conclusion and Future Work

This study examined the question of whether a verification layer using blockchain could ensure the accuracy of federated cloud-load predictions when attacked by data integrity attacks, and offer an auditable training history. With the Alibaba Cluster Trace and three experimental regimes (clean, attacked, and attacked + verified) the results demonstrated that the clean FL model was highly accurate in the accuracy (RMSE = 0.1112, R^2 = 0.9507), but the attacked model was extremely degraded (RMSE = 0.2934, R^2 = 0.6565). The verification layer was able to achieve much more active performance (RMSE = 0.1173, R^2 = 0.9451), and it was able to mitigate the presence of poisoned updates and enhance robustness. The tamper-evident ledger also provided a sense of transparency by maintaining records of verdicts, reasons, hash and timestamps of every client update. The areas of work in the future include adaptive verification thresholds, expansion of threat models, and application with incentives, or smart contracts; and implementation in a real-world cloud or smart-grid system. In general, the suggested verification-and-ledger framework was able to meet the research goals and improve the level of security and auditability in federated learning.

References

- Albshaier, L., Almarri, S. and Albuali, A. (2025). Federated learning for cloud and edge security: A systematic review of challenges and ai opportunities, *Electronics (Switzerland)* **14**: 1019.
URL: <https://www.mdpi.com/2079-9292/14/5/1019/html>
<https://www.mdpi.com/2079-9292/14/5/1019>
- Aledhari, M., Razzak, R., Parizi, R. M. and Saeed, F. (2020). Federated learning: A survey on enabling technologies, protocols, and applications, *IEEE Access* **8**: 140699–140725.
- Arya, A. (2025). Combining federated learning and blockchain to enhance cloud storage security, *European Journal of Applied Science, Engineering and Technology* **3**: 4–16.
URL: <https://ejaset.com/index.php/journal/article/view/164>
- Fang, M., Nabavirazavi, S., Liu, Z., Sun, W., Iyengar, S. S. and Yang, H. (2025). Do we really need to design new byzantine-robust aggregation rules?
URL: <https://arxiv.org/pdf/2501.17381>
- Jeong, H., Son, H., Lee, S., Hyun, J. and Chung, T.-M. (2025). Fedcc: Robust federated learning against model poisoning attacks.
URL: <https://arxiv.org/pdf/2212.01976v2>

- Jiang, Y., Ma, B., Wang, X., Yu, P., Yu, G., Wang, Z., Ni, W., Liu, R. P. and Federated, B. (2023). Blockchain-based federated learning for internet of things: A comprehensive survey, **1**: 1.
URL: <https://arxiv.org/pdf/2305.04513>
- Jovanovic, Z., Hou, Z., Biswas, K. and Muthukkumarasamy, V. (2024). Robust integration of blockchain and explainable federated learning for automated credit scoring, *Computer Networks* **243**: 110303.
URL: <https://www.sciencedirect.com/science/article/pii/S138912862400135X>
- Kang, J., Xiong, Z., Niyato, D., Xie, S. and Zhang, J. (2019). Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory, *IEEE Internet of Things Journal* **6**: 10700–10714.
URL: <https://asu.elsevierpure.com/en/publications/incentive-mechanism-for-reliable-federated-learning-a-joint-optim>
- Li, D., Luo, Z. and Cao, B. (2022). Blockchain-based federated learning methodologies in smart environments, *Cluster Computing* **25**: 2585–2599.
URL: <https://link.springer.com/article/10.1007/s10586-021-03424-y>
- Li, J., Sun, Q. and Sun, F. (2023). Enhancing privacy-preserving intrusion detection in blockchain-based networks with deep learning, *Data Science Journal* **22**.
- Liang, J., Wang, R., Feng, C. and Chang, C.-C. (2023). A survey on federated learning poisoning attacks and defenses.
URL: <https://arxiv.org/pdf/2306.03397>
- Limbepe, Z. N., Gai, K. and Yu, J. (2025). Blockchain-based privacy-enhancing federated learning in smart healthcare: A survey, *Blockchains 2025, Vol. 3, Page 1* **3**: 1.
URL: <https://www.mdpi.com/2813-5288/3/1/1/htm> <https://www.mdpi.com/2813-5288/3/1/1>
- Lo, S. K., Liu, Y., Lu, Q., Wang, C., Xu, X., Paik, H.-Y. and Zhu, L. (2021). Blockchain-based trustworthy federated learning architecture.
URL: <https://arxiv.org/pdf/2108.06912>
- Mao, Q., Wang, L., Long, Y., Han, L., Wang, Z. and Chen, K. (2024). A blockchain-based framework for federated learning with privacy preservation in power load forecasting, *Knowledge-Based Systems* **284**.
URL: <https://dl.acm.org/doi/10.1016/j.knosys.2023.111338>
- Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J. and Poor, H. V. (2021). Federated learning for internet of things: A comprehensive survey, *IEEE Communications Surveys and Tutorials* **23**: 1622–1658.
- Ning, Weiguang, Z. Y. (2024). Blockchain-based federated learning: A survey and new perspectives, *Applied Sciences 2024, Vol. 14*, **14**.
URL: <https://www.mdpi.com/2076-3417/14/20/9459>
- Orabi, M. M., Emam, O. and Fahmy, H. (2025). Adapting security and decentralized knowledge enhancement in federated learning using blockchain technology: literature review, *Journal of Big Data* **12**: 1–24.

- URL:** <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-025-01099-5>
- Ouadrhiri, A. E. and Abdelhadi, A. (2022). Differential privacy for deep and federated learning: A survey, *IEEE Access* **10**: 22359–22380.
- Pillutla, K., Kakade, S. M. and Harchaoui, Z. (2019). Robust aggregation for federated learning, *IEEE Transactions on Signal Processing* **70**: 1142–1154.
URL: <https://arxiv.org/pdf/1912.13445>
- Saleh, A. M. S. (2024). Blockchain for secure and decentralized artificial intelligence in cybersecurity: A comprehensive review, *Blockchain: Research and Applications* **5**.
URL: <https://academix.com/doi/10.1016/j.bcra.2024.100193>
- Shabbir, A., Manzoor, H. U., Manzoor, M. N., Hussain, S. and Zoha, A. (2024). Robustness against data integrity attacks in decentralized federated load forecasting, *Electronics* 2024, Vol. 13, Page 4803 **13**: 4803.
URL: <https://www.mdpi.com/2079-9292/13/23/4803/htm>
<https://www.mdpi.com/2079-9292/13/23/4803>
- Shitharth, S., Manoharan, H., Shankar, A., Alsowail, R. A., Pandiaraj, S., Edalatpanah, S. A. and Viriyasitavat, W. (2023). Federated learning optimization: A computational blockchain process with offloading analysis to enhance security, *Egyptian Informatics Journal* **24**: 100406.
URL: <https://www.sciencedirect.com/science/article/pii/S1110866523000622>
- Sikandar, H. S., Waheed, H., Tahir, S., Malik, S. U. R., Rafique, W., Sikandar, H. S., Waheed, H., Tahir, S., Malik, S. U. R. and Rafique, W. (2023). A detailed survey on federated learning attacks and defenses, *Electronics* 2023, Vol. 12, **12**.
URL: <https://www.mdpi.com/2079-9292/12/2/260>
- Tang, Y., Zhang, Y., Niu, T., Li, Z., Zhang, Z., Chen, H. and Zhang, L. (2024). A survey on blockchain-based federated learning: Categorization, application and analysis, *CMES - Computer Modeling in Engineering and Sciences* **139**: 2451–2477.
URL: https://www.sciencedirect.com/org/science/article/pii/S1526149224001152?utm_source=chatgpt.com
- Xu, Y., Jiang, C., Zheng, Z., Yang, B. and Zhu, N. (2021). Lstm short-term residential load forecasting based on federated learning, *ACM International Conference Proceeding Series* pp. 217–221.
URL: [/doi/pdf/10.1145/3518781.3519181?download=true](https://doi/pdf/10.1145/3518781.3519181?download=true)
- Yurdem, B., Kuzlu, M., Gullu, M. K., Catak, F. O. and Tabassum, M. (2024). Federated learning: Overview, strategies, applications, tools and future directions, *Heliyon* **10**: e38137.
URL: <https://www.sciencedirect.com/science/article/pii/S2405844024141680>
- Zhang, G., Zhu, S., Bai, X., Zhang, G., Zhu, S. and Bai, X. (2022). Federated learning-based multi-energy load forecasting method using cnn-attention-lstm model, *Sustainability* 2022, Vol. 14, **14**.
URL: <https://www.mdpi.com/2071-1050/14/19/12843>