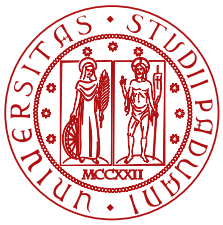


UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

DEPARTMENT OF INFORMATION ENGINEERING

BACHELOR DEGREE IN INFORMATION ENGINEERING

A Study on the Impact of Blockchain to Secure Federated Learning

Supervisor:
Prof. Giovanni Perin

Bachelor Candidate:
Rajatkant Nayak

Academic year 2024-2025

Date of Graduation: 10/03/2025

Abstract

Federated Learning (FL) enables collaborative model training without the need for centralizing data; however, traditional FL frameworks often suffer from significant security and trust challenges, including vulnerabilities to data poisoning, inference attacks, and depends on a central aggregator. To overcome these problems, this thesis proposes a Blockchain-Enabled Federated Learning (BCFL) framework that integrates blockchain technology to decentralize model update aggregation, thereby providing an immutable and transparent ledger for recording updates. The framework incorporates the InterPlanetary File System (IPFS) for decentralized storage of model weights and employs a simulation of Zero-Knowledge Proofs (ZKPs) to verify the integrity of submitted updates without compromising privacy.

The proposed BCFL framework is evaluated on three image-classification tasks (MNIST, FashionMNIST, and CIFAR-10). Experimental results demonstrate that while the integration of blockchain operations introduces a moderate computational overhead—typically increasing training time by approximately 4% to 24%—the final model accuracy remains comparable to that of standard FL. Our primary goal was to maintain the accuracy even after introducing blockchain, which we have achieved here. Moreover, the blockchain-based approach enhances security by ensuring update traceability and robustness against tampering, with an effective outlier detection mechanism further mitigating potential malicious contributions.

This research highlights a crucial trade-off between additional computational costs and the significant gains in transparency, security, and trust in distributed learning environments. The study also identifies limitations, including scalability challenges under real-world network conditions and the need for advanced cryptographic methods to replace the simplified ZKP simulation. Future research directions include the exploration of more efficient consensus mechanisms, the integration of advanced privacy-preserving techniques, and the development of interoperable BCFL systems capable of operating across heterogeneous blockchain networks..

Contents

List of Figures

Chapter 1

Introduction

Data-driven approaches have become very popular for modern world intelligent services, especially with the rapid growth of the Internet of Things (IoT) and Edge Computing, which generate a ton of data for machine learning (ML) models. In recent years, Federated Learning (FL) has come out as an innovative approach that enables multiple data owners or edge devices to train a shared global model without centralizing their raw data [1]. In short, we can also state that FL is a decentralized version of the traditional Machine Learning model. By storing the data locally and only exchanging learning parameters, FL improves privacy and efficient data utilization, making it more reasonable to use in sectors such as healthcare and consumer applications, where privacy is a key measure [2]. However, only FL does not fully resolve issues related to data integrity, security, and trust. A traditional FL system depends on a central server, which can become a vulnerability point. It could affect the training, manipulate model updates, or leak sensitive data, if compromised [3].

Along with FL, another popular technology called Blockchain has proven effective in ensuring decentralized trust, transparency, and tamper-resistant record-keeping [4]. Initially created for secure peer-to-peer cryptocurrency transactions, blockchain has evolved into a general-purpose distributed ledger that records and verifies system state transitions in an immutable way [5]. In scenarios where participants do not fully trust each other or require strong auditing capabilities, blockchain's decentralized nature eliminates single points of failure through distributed consensus. This leads to the emergence of Blockchain-Enabled Federated Learning (BCFL), where blockchain acts as a decentralized and auditable aggregator, while FL ensures data privacy by keeping raw data on local devices [6].

Although they are strongly compatible, integrating blockchain technology with FL comes

with multiple challenges such as high computational costs, increased communication overhead, and new security risks, which must be carefully addressed to ensure efficiency and reliability [7]. However, Blockchain-Enabled Federated Learning (BCFL) also brings a lot of opportunities, such as trustless data sharing, enhanced security through decentralized incentives, and improved tamper resistance. By removing the dependence on a central authority, BCFL creates a more resilient and transparent training process. This chapter delves into the FL framework, examines how blockchain can strengthen or replace the traditional FL coordinator, and explores how this fusion can lead to a more secure, privacy-preserving, and trustworthy machine learning system.

1.1 Overview of Federated Learning

Federated Learning (FL) is a method that moves away from traditional centralized model training. Instead of sending raw data to a central server, each client trains a model locally using its own private data. Then they share only the necessary model updates (such as gradients or weights) with a central coordinator, which combines them into a global model [1]. This process repeats in cycles until the model reaches a satisfactory performance level.

1.1.1 Why Federated Learning?

FL has become popular because it addresses several key problems related to data privacy, regulations, and efficiency:

- **Meeting privacy requirements:** Many industries, such as healthcare and finance, rely heavily on data privacy. Since FL keeps data on local devices instead of storing it in a central location, it allows organizations to train models collaboratively while still meeting their privacy requirements [2].
- **Data Privacy:** Because raw data never leaves the client's device, FL significantly reduces the risk of massive data breaches [4].
- **Bandwidth Efficiency:** In contrast to conventional methods, which require transferring large datasets, FL only sends model parameters, making it perfect for edge devices with limited bandwidth [1].
- **Personalization:** FL allows for local fine-tuning, meaning each device can adapt the model to its specific needs while still contributing to a shared, global model [8].

1.1.2 Challenges in Traditional Federated Learning

FL brings many advantages but also comes with several disadvantages:

- **Single Point of Failure (Aggregator):** Traditional FL depends on a central server to coordinate training. If this server fails, gets hacked, or is compromised, the entire system is affected [9].
- **Data Poisoning Attacks:** Malicious participants can inject corrupted updates into the training process, harming the global model's accuracy or even introducing hidden biases [6].
- **Inference Attacks:** Even though raw data stays local, an attacker can still analyze gradient updates to obtain private information from clients [2].
- **Lack of Transparency:** It's often unclear who contributed what to the final model. This lack of an audit trail makes it difficult to track errors, detect fraud, or resolve disputes [4].

These challenges state the need for a decentralized system, where one can verify, store, and audit all contributions without depending on a single authority. Blockchain technology is a strong candidate for filling this role [10].

1.2 Understanding Blockchain

A blockchain is a decentralized ledger that records information in linked blocks. Each block includes a cryptographic hash of the one before it, creating a secure and tamper-resistant structure [4]. This design ensures data integrity by making it extremely difficult to alter past records. Several features of blockchain make it a strong match for Federated Learning (FL):

- **Decentralized Trust:** Instead of depending on a single authority, blockchain uses consensus protocols across multiple nodes to validate and add new blocks, ensuring a more trustworthy and failure-resistant system [5].
- **Immutability & Transparency:** Once data (such as model updates) is stored on the blockchain, it cannot be easily altered. Additionally, all nodes in the network share a consistent view of the stored information, promoting accountability [1].
- **Smart Contracts:** Some blockchains support automated scripts (smart contracts) that can handle tasks like verifying model updates or distributing incentives without human intervention [6].
- **Security Against Malicious Actors:** Blockchain is designed to tolerate and expose dishonest behavior. Even if some participants attempt to manipulate the system, properly designed consensus mechanisms can detect and prevent such attacks [4].

1.3 How Blockchain and Federated Learning Work Together

Blockchain-Enabled Federated Learning (BCFL) integrates Federated Learning (FL) with blockchain's decentralized security and trust mechanisms [7]. Rather than relying on a central server to collect model updates, BCFL clients submit their updates to a blockchain network, ensuring fairness, transparency, and accountability throughout the training process.

- **No Central Aggregator:** Traditional FL depends on a single central server to collect and process model updates. BCFL eliminates this single point of failure by distributing this role across multiple blockchain nodes, making the system more resilient and decentralized [4].

- **Auditability:** Every model update is permanently recorded on the blockchain. This transparency makes it easy to trace contributions, detect anomalies, and even roll back harmful changes if needed [7].

- **Incentives for Participation:** Smart contracts can reward users for submitting accurate and useful model updates while punishing attempts to manipulate the model, encouraging ethical behavior [8].

- **Enhanced Security:** Since blockchain relies on cryptographic techniques, any attempt to tamper with model updates becomes easily detectable, making it much harder for attackers to secretly alter training data [2].

1.4 Challenges and Key Research Questions

Although BCFL has many advantages, it still faces some disadvantages and challenges that must be addressed. Some of these issues are:

- **Scalability Issues:** Blockchains often have limited processing capacity (throughput), meaning they may struggle to handle thousands of model updates at once without slowing down the system [10].

- **Increased Communication Overhead:** FL is already data-intensive, requiring frequent updates between clients and servers. Adding blockchain introduces extra steps for validation, block creation, and synchronization, which can further increase network congestion [4].

- **Balancing Security and Efficiency:** Some blockchain consensus protocols (like PBFT) offer strong security but slow down performance. BCFL needs to find the right balance between reliability and speed [7].

- **Privacy Concerns:** Although FL improves privacy by keeping raw data local, storing metadata or partial updates on a public blockchain could introduce new privacy risks. Techniques like zero-knowledge proofs or homomorphic encryption may help [6].

- **Good Incentive Models:** In open and decentralized networks, participants need clear rewards to contribute their computing power or data. Designing mechanisms to reward honest users and penalize malicious ones is a complex challenge [8].

These challenges drive ongoing research into better consensus mechanisms, cryptographic security methods, and hybrid blockchain architectures. Future innovations might include sidechains (smaller auxiliary blockchains) or aggregator sub-blockchains to manage model updates more efficiently. By solving these issues, BCFL could revolutionize secure and decentralized machine learning.

1.5 Real-World Applications of BCFL

Blockchain-Enabled Federated Learning (BCFL) has a lot of use cases in various industries by enhancing privacy, security, and collaboration in machine learning. Some key applications include:

- **Industrial IoT:** Factories rely on sensor data to monitor equipment and optimize production. With BCFL, industries can train local models while maintaining trust among different manufacturers, suppliers, and logistics providers, ensuring that no single party manipulates the shared data [9].

- **Healthcare:** Hospitals can train AI models for disease diagnosis without sharing sensitive patient data. Blockchain adds a decentralized and traceable ledger, allowing medical institutions to verify the entire training history and ensure no data has been tampered with [6].

- **Edge Intelligence:** Many real-time applications, like anomaly detection in sensor networks, rely on fast and efficient model training. BCFL enables low-latency AI processing at the edge while ensuring security through decentralized verification, preventing a single point of failure [2].

- **Smart Transportation:** AI-driven vehicle detection and traffic signal optimization can be enhanced by BCFL, where blockchain ensures that data manipulation at one location does not compromise the entire traffic management system [11].

1.6 Structure of This Thesis

This thesis is organized into four main chapters:

- **Chapter 2: Background**
  - Introduces Federated Learning (FL) and its current state-of-the-art techniques.
  - Discusses security risks, including data poisoning and inference attacks.
  - Reviews existing defense mechanisms used in FL.
- **Chapter 3: Implementation of Blockchain-Enabled Federated Learning**
  - Explains the technical implementation of BCFL, including code structure, smart contracts, and consensus mechanisms.
  - Presents experimental results, evaluating performance, security, and efficiency.
- **Chapter 4: Conclusions**
  - Summarizes the key insights from this study.
  - Discusses open challenges and future research directions, including scalability, efficiency, and real-world deployment.

Chapter 2

Background

Recently, Federated Learning (FL) has attracted a lot of interest mostly because it allows cooperatively training models without aggregating raw data in one single area [1]. First covering the FL idea and its most advanced techniques, this section next highlights possible security threats in FL—more especially, data poisoning and inference attacks—and lastly summarises current defences meant to maintain model dependability and client privacy.

2.1 Overview of Federated Learning

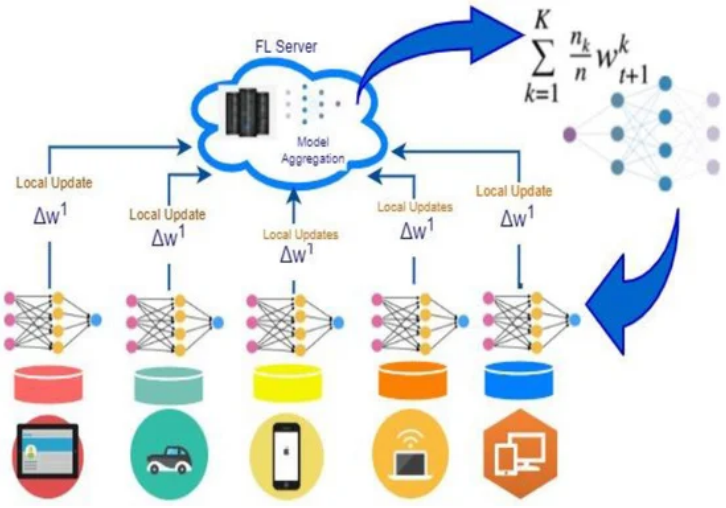


Figure 2.1: Federated Learning architecture, adapted from [12].

Often labelled as “clients,” Federated Learning (FL) is based on distributing the training task among numerous devices or organisational units without migrating their private data to a central server, hence eliminating centralisation [1], [2]. The key concept is that every client communicates locally computed parameters—for example, gradients—instead of raw datasets, therefore reducing the possibility of leaking personal data. After gathering these parameter updates, a server updates a global model and forwards it back to every client for more local training. This iterative cycle continues until the global model settles or some halting criteria is met [1], [4].

Unlike conventional centralised machine learning, which collects all data into a single server, FL maintains each client’s data within its own institution [8]. This technique is especially crucial in situations where data privacy laws like the General Data Protection Regulation (GDPR) prohibit data from being shared outside its original organisation [3].

Originally, FL was introduced to handle large-scale data on devices like smartphones. Transferring raw data is risky (privacy concerns), expensive (bandwidth usage), and complex (hardware differences) [1]. By allowing millions of devices to work together and train a shared model, FL ensures better generalisation across different data sources [9].

2.2 FL Partitioning and Classification

The Federated Learning (FL) categorisation depends on the distribution of data among the participants. There exist three primary forms of FL:

2.2.1 Homogeneous FL—Horizontal Federated Learning

When clients have separate user sets but data using the same feature space, horizontal FL is employed. Every client has the same dataset; the global model selects on regularly ordered updates.Users split datasets in horizontal FL horizontally; that is, each row indicates a different user, hence clients share common features but have different user records [4]. This guarantees that data stays private and increases the training sample count, hence improving model accuracy [4].

Example: Two mobile service providers in different locations may have different customer bases, but their user data structure and business practices are the same. Here, horizontal FL allows training of a shared prediction model [4] while maintaining user data privacy.

Real-world example: Google’s Android Keyboard (Gboard) trains text prediction models across millions of user devices without providing raw typing data to central servers [4].Although it may expose private information about some users, horizontal FL can still be vulnerable to privacy leaks

through gradient changes. Common techniques including homomorphic encryption, differential privacy, and safe aggregation [4] provide safe gradient sharing.

2.2.2 Vertical Heterogeneous Federated Learning

Vertical FL is used when clients share a user base but vary in feature sets. Every client thus has multiple types of information about the same persons, and data is spread using feature columns instead of user records [3]. When many businesses compile complementing data about the same users but cannot share raw data due to privacy rules, vertical FL is useful [3].
Example: Even if they have many of the same customers, stored data differs between a bank and an e-commerce company situated in the same city. The bank keeps credit scores and financial data, while the e-commerce website records purchase history and surfing habits [3]. Using Vertical FL, both businesses can work together to train a fraud detection model without exchanging actual data. Instead, shared encrypted feature embeddings enable improving expected accuracy [3].
Models of Machine Learning Suitable for Vertical FL Consist In:

- Decision Trees
- Neural Networks
- Statistical Regression Models

One of the main challenges in vertical FL is computational overhead, since many encryption methods—e.g., homomorphic encryption and safe multi-party computation—are required to protect important properties during model training [3].

2.2.3 Federated Transfer Learning (FTL)

Federated Transfer Learning (FTL) is applied when datasets comprise both unique users and independent features since user sets and feature spaces only partially overlap [4], [6].FTL is useful when two firms want to collaborate on a machine learning project yet have relatively different types of data [4].**Example:** Geographical distance and different data collection methods mean limited user overlap between a social network company in the United States and an e-commerce platform in China [6].FTL allows these organisations to pool data and create a better model utilising transfer learning techniques even with minimal common knowledge [6].FTL mostly helps to enable effective learning even in situations of limited data availability.**Example in Healthcare:** It is quite beneficial when hospitals maintain few medical records for uncommon diseases. By use of transfer learning, knowledge from linked medical datasets can help to enhance predictive models [6]. However, FTL does demand additional processing phases such as domain adaptation and feature alignment [6] to ensure that information transfer is meaningful.

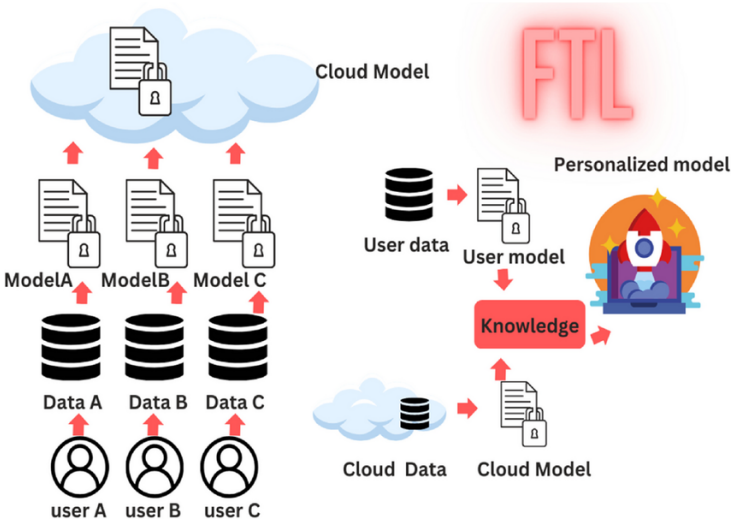


Figure 2.2: Architecture of Federated Transfer Learning, adapted from [13].

2.3 Modern FL Approaches

Federated Learning (FL) research has progressed to handle security issues, heterogeneous hardware, and communication overhead [1], [2], [10] since its inception. These are some innovative methods:

2.3.1 Communication: Effective Techniques

Compression Models

Model compression lowers the bandwidth needed for training via means of sparsification, quantisation, and sub-sampling, hence reducing the size of updates [1].Using difference model compression and pruning extra gradients help to lower the communication cost in FL [1].Further maximising model compression efficiency are dropout-based lossy compression and Golomb lossless coding [1].These techniques will help FL to maintain low computational demand on edge devices and attain steady communication efficiency [1].

Client Sampling and Planning

Servers choose a subset to expedite training rather than include every client in every training round [5].Adaptive client selection systems guarantee that only high-quality clients with enough computational capability participate, hence enhancing convergence rates [5].To dynamically select resource-efficient clients, the Federated Client Selection (FedCS) method has been presented, so improving training efficiency [5].This guarantees effective allocation of processing resources in FL [5] and helps to lower straggler effects.

Decentralised Policies

Like gossip networks, some FL models cut out the central server and rely on peer-to-peer updates [2].By removing a single point of failure, this approach increases FL's resilience against server attacks [2].Consensus-based updates and decentralised FL systems such as Blockchain-based FL help to avoid harmful client impact [2].Decentralisation increases complexity, though, which increases the computationally costly synchronising and aggregation [2].

2.3.2 Managing Partial Participation and System Heterogeneity

Asynchronous FR

Different processor rates and network availability found in FL devices cause straggler problems in conventional synchronous updates [4].Faster clients enabled by asynchronous FL can continue their education free from waiting for slower ones [4].Fairness is guaranteed and biased model updates in asynchronous FL [4] are avoided via weighted aggregation procedures.FL can reach faster convergence by using asynchronous updates, hence lowering training delays brought on by sluggish clients [4].

Adaptive Concentration

Based on client performance and data trends, FL servers change the merging of updates [9], [14].Dynamically changing aggregation rules guarantees that outlier updates do not adversely affect the model [9], [14] thereby enhancing its resilience.Gradient weighting systems balance updates from customers with significant volatility in data distributions [9], [14].Even in cases of diverse client devices [9], [14], FL guarantees optimal model updates by means of adaptive aggregation [9].

2.3.3 Blockchain-Enabled FL

Including blockchain with FL will remove central failures and offer tamper-proof logging [8], [10].The blockchain keeps an unchangeable ledger of model updates, thus avoiding hostile interventions and tampering [8], [10] so improving security.Blockchain-based FL helps sectors including supply chains, finance, and healthcare, where transparency and data integrity are paramount [7], [15].While punishing bad actors [7], [15], smart contract-based incentive systems help to reward honest customers.Blockchain adds additional delay because of consensus procedures, so real-time FL applications [8], [10] need further optimisation.

2.4 Security Concerning Federated Learning

Federated Learning (FL) is prone to assaults even if it does not exchange actual data. Data poisoning and inference assaults rank two most often occurring hazards [11], [16].

2.4.1 Data Poisoning Attacks

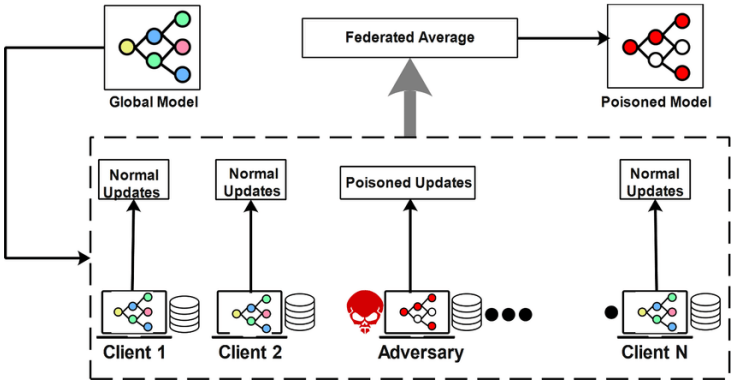


Figure 2.3: Poisoning attack in a federated learning environment, adapted from [17].

Attackers in data poisoning tamper with local data such that the global model gets corrupted. Before forwarding updates to the server, a hostile client might change training labels or introduce biassed data [6], [8].

Standard Data Poisoning Techniques

- Attackers falsify class labels (e.g. "cat" → "dog"), thus distorting the global model [18].
- Attackers provide particular prejudices while making sure the model runs as expected on other data. This approach hides backdoor triggers generating erroneous predictions in selected scenarios [19].

Since FL servers cannot directly validate the data, even a small number of corrupted updates can greatly compromise the performance of the model [9], [16].

2.4.2 Inference Attacks

In an effort to protect privacy, inference attacks retrieve information about a client's data set from distributed model updates [5].

Approaches of Inference Attacks

- Attackers identify whether a given client used a given data point, therefore revealing sensitive user involvement [2], [14].

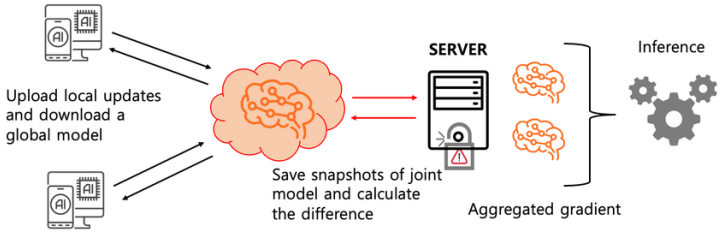


Figure 2.4: An illustration of an inference attack on federated learning, adapted from [20].

- Gradient Leakage: Sensitive data can be revealed in with one round of gradient modifications. Modern reconstruction methods can retrieve whole samples from gradients [5], [16].
- Monitoring updates over time helps attackers to reassemble text, pictures, or other sensitive data [1], [11].

Not often enough to stop privacy leaks is FL by itself. Attackers still can deduce information from gradient patterns [3], [5], [19].

Concluding thoughts

Still a hot topic for study is FL security issues. Rising sophisticated poisoning and inference attacks calls for stronger security procedures incorporating anomaly detection techniques [11], [16], and privacy-preserving processes via strong aggregation.

2.5 Current Defences in FL

To address security issues [9], [10], [14], researchers have proposed many protection strategies to guarantee privacy, resilience, and trust in the learning process in Federated Learning (FL), thereby counteracting security concerns [10]. We review the primary defensive strategies below.

2.5.1 Robust Aggregation Techniques

Data poisoning attacks control local model updates, so robust aggregation techniques concentrate on minimising their effects by lowering the influence of negative updates.

Krum & Multiple-Krum

Krum chooses the most trustworthy model updates from the clients after discounting likely anomalies. Rather than averaging all updates, it chooses one update that fits most client updates, hence lessening the impact of tainted contributions [9].Multi-Krum strengthens resistance against hostile clients by aggregating multiple selected updates over several rounds, extending this

approach [16].These methods greatly help to handle Byzantine failures and model poisoning attacks since they restrict the extent of influence a single hostile client can have on the global model.

Trimmed Median & Mean Aggregation

These methods remove extreme values before the final model update computation [11].Trimmed Mean eliminates a percentage of the highest and lowest gradient values to avoid hostile outliers from skewing aggregation [18].More robust to poisoned contributions, Median Aggregation considers the coordinate-wise median of updates rather than averaging them.In non-identically distributed (non-IID) data environments, where client updates vary greatly [11], these methods are particularly helpful.

2.5.2 Differential Privacy (DP)

Differential Privacy (DP) guarantees that even if an attacker accesses model updates, private information cannot be extracted from individual users [14].

DP Local

Before forwarding their own gradient updates to the aggregator [14], clients add controlled noise to them.This method ensures that even the central server cannot deduce specific details about any individual client's data [11].While good for privacy, too much noise could lower model accuracy, so privacy must be traded off with performance.

Global DP

Global DP protects all aggregated updates by applying noise at the server level instead of client-level modification of updates [3], [5].This stops outside attackers from deducing sensitive knowledge by examining global model updates.Although it preserves model integrity more than Local DP, depending on the degree of additional noise, it may still result in lower accuracy [14].

2.5.3 Homomorphic Encryption (HE)

A cryptographic method called Homomorphic Encryption (HE) allows computations to be performed on encrypted data without decrypting it [15].Before they leave the client's device, HE encrypts gradient updates, thereby guaranteeing privacy [7]. Without accessing raw values, the aggregator can still compute the sum of encrypted gradients and update the global model [15].

Although Fully Homomorphic Encryption (FHE) supports complex operations, it is computationally costly and difficult to apply in large-scale FL settings [7].Some simpler HE methods, such as Paillier encryption, offer efficient additive homomorphism, allowing federated learning models to manage encrypted gradients efficiently while maintaining privacy [15].

2.5.4 Blockchain with Secure Multiparty Computation (MPC)

Blockchain for FL Security

- Blockchain provides a distributed ledger where a transparent record of model updates helps to prevent illegal modification [8].
- Every FL client cryptographically signs its updates, thus guaranteeing immutability and verifiability [10].
- Applications requiring strong trust systems, such as finance and healthcare, benefit most from this approach [8].
- Blockchain also enables reward systems to incentivize honest participation in FL [10].

Secure Multiparty Computation (MPC)

- MPC allows multiple clients to jointly train a model without disclosing their raw data to each other [11].
- Unlike DP, which adds noise, MPC ensures that no one party gains access to complete updates during aggregation [11].
- While MPC is highly secure, it is difficult to scale in large FL networks because it requires significant communication overhead [11].

Federated Learning (FL) keeps raw data decentralized, offering significant privacy advantages. However, it is still vulnerable to major risks, including data poisoning and inference attacks.By incorporating strong aggregation, differential privacy, encryption, blockchain, and MPC, the techniques discussed here provide a multi-layered defense strategy. Each approach balances privacy, performance, and computational efficiency.Combining multiple defense techniques will help to optimize FL security, ensuring privacy, resilience, and efficiency. Future research aims to enhance these strategies to improve scalability and real-world adoption [9], [10], [14].

2.6 Existing Research on Blockchain-Based Federated Learning

Several researchers have investigated how adding a blockchain layer to Federated Learning (FL) can enhance privacy, security, and trust. For example, in [19], the authors show that Blockchain-Enabled Federated Learning (BCFL) can maintain user privacy without losing too much accuracy. They test various FL models and security risks like inference and poisoning attacks, and they also explore privacy-preserving techniques such as differential privacy (DP), homomorphic encryption (HE), and multiparty computation (MPC). Their key finding is that DP protects data but can lower model accuracy by 2–5%; HE makes training more secure but increases computational time by 2–4×; and using a blockchain layer adds trust and security but also raises latency and storage needs. In real-world experiments, they achieve up to 98.6% accuracy on medical imaging tasks, 0.93 F1-scores in smart city use cases, and demonstrate promise in Industry 5.0 applications, although higher energy consumption remains an issue.

Other work [2] emphasizes how blockchain can reduce the impact of data poisoning and inference attacks on FL. By using Ethereum smart contracts and Zero-Knowledge Proofs (ZKPs), the authors effectively verify model updates and encourage honest participant behavior. Their experiments, conducted on the Fashion-MNIST dataset, show that standard FL can lose nearly half of its accuracy under attack, whereas blockchain-based FL only drops by about 7%. While they achieve a high accuracy of 95% and prevent most inference threats, they note that overall computational costs rise due to blockchain operations. As a solution, they propose more efficient consensus mechanisms (e.g., Proof-of-Stake or hybrid blockchains) to limit overhead.

A broad survey [5] gives a high-level view of how blockchain can be integrated into FL. The authors focus on privacy, decentralized training, and incentive models, observing that blockchain can cut down on poisoning attacks by up to 80% and achieve 97% accuracy in IoT and healthcare settings. However, they also point out that Proof-of-Work (PoW) increases energy consumption by 2–3× and prolongs training through added latency (30–50%). In the same spirit, [10] introduces a system called FL Chain, tailored for Mobile Edge Computing (MEC). This approach reduces poisoning attacks by 85% and reaches 92% accuracy in vehicular applications, but experiences 20–60% longer latency depending on the consensus used. They suggest that switching to Proof-of-Stake reduces energy consumption by 30%.

In the Industrial IoT (IIoT) sphere, [14] proposes STFS, which uses a blockchain layer to safeguard data exchanges and securely manage model parameters in Federated Learning with FedXGBoost. The system experiences just a 1.5% accuracy drop compared to standard XGBoost, while effectively preventing inference attacks. Smart contracts also streamline ownership tracking and data access. Meanwhile, [18] targets Social Media (SM) 3.0 environments with a system

called DP-BFL, which blends blockchain, differential privacy, and FL. Their setup achieves about 97.8% accuracy on SM 3.0 data, blocking 85% of poisoning attacks. Although they initially rely on Proof-of-Work—leading to up to 12 malicious leaders—they find that a Quorum-Based Consensus (QBC) eliminates these attacks altogether.

Lastly, [8] proposes FL chain for Multi-Access Edge Computing (MEC), emphasizing secure, tamper-proof data logging via Merkle Patricia Trees. While Proof-of-Work can add substantial latency (40–60%), switching to Practical Byzantine Fault Tolerance (pBFT) speeds training. They also highlight that a channel-based network layout improves scalability by minimizing bottlenecks. Taken together, these works confirm that blockchain adds significant protection and transparency to FL but requires careful optimization of consensus mechanisms, energy usage, and latency to be truly practical at scale.

Table 2.1: Summary of Key BCFL (Blockchain-Based Federated Learning) Works

Reference	Focus/Approach	Key Findings	Challenges/Future Work
[19]	BCFL for privacy & efficiency	• DP lowers accuracy by 2–5% • HE increases compute by 2–4× • 98.6% accuracy on medical tasks	• Optimize blockchain consensus • Adopt hybrid encryption
[2]	Multi-layer security using ZKPs and Ethereum	• Standard FL drops 47.51% under attack • BCFL reduces loss to 7.24% • Achieves 95% accuracy with inference protection	• High computational cost • Suggest PoS or hybrid consensus
[5]	Survey of BC and FL integration	• BCFL cuts poisoning by up to 80% • PoW raises energy 2–3× • IoT/health use cases reach 97% accuracy	• PoS or committee-based consensus • Balancing scalability and security
[10]	FL Chain for Mobile Edge Computing	• Reduces poisoning by 85% • Increases latency by 20–60% • PoS cuts energy usage by 30%	• Refine consensus mechanism • Explore hybrid architectures
[14]	STFS for secure IIoT data sharing	• FedXGBoost + TSS (Threshold Signature Scheme) • Only 1.5% accuracy drop vs. standard XGBoost • ∼1s contract lookups	• Improve scalability • Reduce power consumption
[18]	DP-BFL for Social Media 3.0	• 97.87% accuracy • Blocks 85% of poisoning attacks • QBC prevents malicious leaders	• Increase blockchain throughput • Validate in real-world SM 3.0
[8]	FLchain for Multi-Access Edge Computing (MEC)	• Uses Merkle Patricia Trees to secure updates • PoW adds 40–60% latency • pBFT enables faster training	• Enhance consensus protocols • Develop incentive models

Chapter 3

Implementation of Blockchain-enabled Federated Learning

3.1 Introduction

From cryptocurrency to data sharing in Internet of Things (IoT) systems, blockchain technology has become a transforming distributed ledger system supporting many modern uses. Over the past decade, blockchain has developed from a niche solution for digital currency to a comprehensive paradigm that enables decentralised, secure, and transparent systems [5]. In federated learning, blockchain is being used to get around problems like single-point failures, a lack of trust, and changing data [6], [10]. Including its basic ideas, main architectural components, consensus systems, and role in distributed applications, this chapter offers a thorough introduction to blockchain technology.

3.2 Overview of Blockchain Technology

Blockchain is a distributed ledger with an immutable and safe transaction recording mechanism. Fundamentally, blockchain depends on a distributed network of nodes whereby no single entity manages the whole system [5]. Critical issues in distributed learning systems, such as data breaches and single points of failure associated with centralised architectures, are mitigated by this distributed approach [1], [10].

Originally developed by Bitcoin, the idea of blockchain has recently found applications in

supply chain management, healthcare, and industrial IoT [5]. In blockchain-enabled federated learning, technology is leveraged to ensure that model updates are securely recorded, verified, and aggregated without relying on a trusted central server [6]. This not only improves security but also provides an immutable audit record that increases confidence among participating nodes [10].

3.3 Core Components of Blockchain Architecture

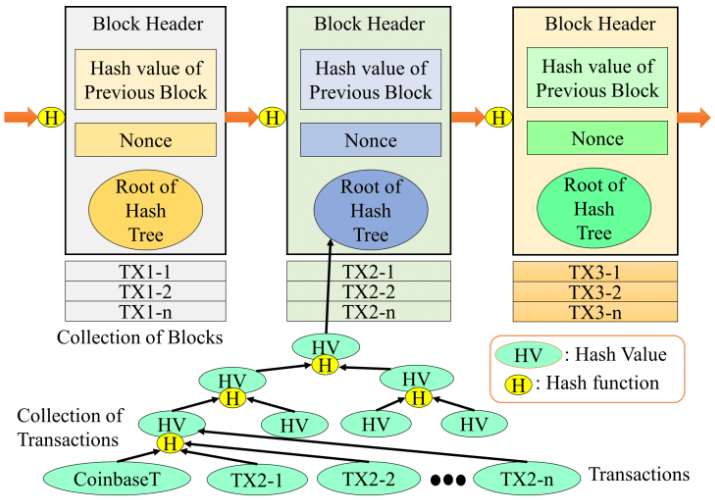


Figure 3.1: Structure of blockchain, adapted from [21].

Blockchain systems are composed of several fundamental components that work in concert to provide a transparent, secure, and decentralised data storage and transaction processing solution. These elements are especially important in federated learning, where trust, privacy, and immutability are critical. In this section, we present each element of blockchain architecture as reported in the literature [5], [6], [9], [10], [15], [18].

3.3.1 Blocks and the Ledger Structure

At the core of any blockchain is the ledger—a distributed, append-only database composed of sequentially linked blocks. Each block consists of two primary parts:

- **Block Header:** This portion contains metadata essential for maintaining the chain's integrity. The header includes a timestamp, a unique block identifier, a reference (in the form of a cryptographic hash) to the previous block, and—depending on the consensus mechanism—a nonce used in proof-of-work algorithms [5]. The inclusion of the previous block's hash in every block creates an immutable chain; any alteration to a block's contents would change its hash and break the chain's continuity [5].

- **Block Body:** The body contains the list of transactions or data records. In federated learning applications, these records might include model updates, training round information, or interactions triggered by smart contracts [9]. Organising data into blocks allows for efficient verification through the validation of hash links between consecutive blocks [5].

This chained structure not only provides tamper-evidence but also distributes the transaction record among all network participants, thereby ensuring transparency and trust [5].

3.3.2 Cryptographic Foundations

The security and reliability of a blockchain rely on robust cryptographic techniques. Key primitives include:

- **Hash Functions:** Each block's content is processed using cryptographic hash functions (such as SHA-256) to produce a fixed-length, unique fingerprint. Since each block header is linked to its predecessor via its hash, any change in a block's data generates a new hash, making unauthorized modifications easily detectable [5], [10].
- **Digital Signatures:** Public-key cryptography is used to generate digital signatures for transactions. These signatures verify the origin of transactions and ensure that the data has not been altered, thereby confirming that only authorized participants can submit valid updates [6].
- **Encryption:** In addition to signatures and hashes, encryption protects sensitive data stored or transmitted via the blockchain. In federated learning—especially in industrial IoT applications where privacy is paramount—advanced techniques such as homomorphic encryption enable computations on encrypted data without exposing the raw values [15].

Together, these cryptographic methods create an environment in which data integrity and authenticity are maintained, and unauthorized modifications become computationally infeasible [5], [15].

3.3.3 Consensus Mechanisms

A major challenge in any distributed system is ensuring that all nodes agree on the state of the ledger. Blockchain achieves consensus through protocols that allow the network to reach agreement even in the presence of faulty or malicious nodes. Common consensus mechanisms include:

- **Proof-of-Work (PoW):** In PoW systems, miners compete to solve complex mathematical puzzles. The first node to solve the puzzle earns the right to add a new block to the chain.

While PoW is highly secure because of its computational intensity, it is also resource-intensive and may not be optimal for all applications [5], [18].

- **Proof-of-Stake (PoS):** PoS selects validators based on the amount of cryptocurrency they hold and stake as collateral, making it more energy-efficient than PoW. This consensus method is increasingly popular in modern blockchain implementations [18].

- **Other Protocols (DPoS, BFT):** Delegated Proof-of-Stake (DPoS) and Byzantine Fault Tolerance (BFT) are additional protocols that offer scalability and lower latency. These protocols are particularly useful in edge computing scenarios where rapid consensus is critical for time-sensitive operations, such as federated learning model aggregation [18].

The consensus mechanism selected has a significant impact on network performance, scalability, and resistance to attacks. In federated learning, the consensus process must ensure that only valid and trusted model updates are recorded on the blockchain, thereby preserving the integrity of the global model [18].

3.3.4 Nodes and Network Participation

A blockchain network comprises various types of nodes that work together to ensure overall system functionality:

- **Full Nodes:** Full nodes verify and validate transactions while maintaining a complete copy of the blockchain. They play a key role in decentralising the network and ensuring that every participant has access to the entire ledger history [5]. In federated learning, full nodes can serve as auditors of model updates and robust aggregators [6].

- **Light Nodes:** Also known as Simplified Payment Verification (SPV) nodes, light nodes do not store the entire blockchain but rely on full nodes for transaction verification. This approach is especially beneficial for IoT and mobile devices that have limited storage and processing power [5].

- **Miners/Validators:** Depending on the consensus mechanism, these nodes are responsible for constructing new blocks. In PoW, miners perform resource-intensive computations, whereas in PoS, validators are chosen based on their staked value. These nodes are critical for maintaining blockchain continuity by validating transactions and adding new blocks [6], [18].

3.3.5 Smart Contracts and the Execution Layer

Smart contracts are self-executing programs stored on the blockchain that automatically enforce the terms of an agreement when predetermined conditions are met [5]. They provide the execution layer in blockchain architectures and offer several benefits:

- **Automation:** In federated learning systems, smart contracts can automate the aggregation of model updates, transaction validation, and distribution of incentives. This integration eliminates the need for a central authority or human intervention [9].
- **Transparency and Trust:** Because smart contracts are stored on the blockchain and their execution is recorded immutably, their operations are publicly verifiable, which fosters trust among network participants [10].
- **Security:** Smart contracts ensure that only legitimate transactions are executed, enforcing policies such as proper incentive distribution and verifying that only authorized model modifications are accepted [9], [10].

3.3.6 Network Layers and Interoperability

Modern blockchain systems are often conceptualised as layered architectures. This layered model helps clarify the responsibilities of various components:

- **Infrastructure Layer:** Comprises the physical hardware, blockchain network, communication channels, and networking protocols that form the foundation for all other layers [5].
- **Network Layer:** Responsible for peer-to-peer communication, this layer enables nodes to exchange data and broadcast transactions, thereby upholding the decentralised nature of the system [5].
- **Consensus Layer:** Sits atop the network layer and implements the consensus protocols (e.g., PoW, PoS, BFT) that ensure all nodes agree on the blockchain's state [18].
- **Application Layer:** The top layer where smart contracts and decentralized applications (DApps) reside. This layer provides end-user functionalities and integrates blockchain with other systems, such as federated learning platforms [9], [10].

This layered approach enhances interoperability and scalability, allowing blockchain networks to adapt to a wide range of applications—including those in industrial IoT and federated learning [9], [10].In summary, the core components of blockchain architecture—ranging from the structure

of blocks and the ledger, through cryptographic foundations, consensus mechanisms, network participation, and smart contract execution—collectively form a robust, decentralised, and secure system. This architecture not only guarantees data integrity and transparency but also provides the essential framework for advanced applications such as blockchain-enabled federated learning [5], [6], [9], [10], [15], [18].

3.4 Types of Blockchain Networks

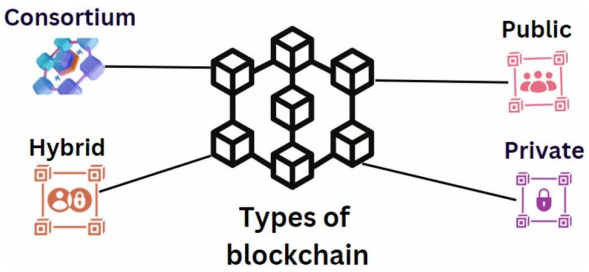


Figure 3.2: Types of Blockchain, adapted from [22].

Depending on its accessibility, governance, and operational traits, blockchain networks can be generally divided into numerous groups. Every kind has unique benefits and trade-offs that fit various uses, including edge computing, healthcare settings, and federated learning in industrial IoT. We enumerate the four primary types—public, private, consortium, and hybrid blockchains—in the sections that follow.

3.4.1 Public Blockchains

Public blockchains are fully decentralized and open to anyone who wishes to participate. They operate on a permissionless basis, meaning that any node can join the network, contribute to the consensus process, and access the ledger without restrictions. The problem with public blockchains is their high level of transparency and robustness against censorship or central points of failure.

Decentralization and Transparency: Public blockchains maintain a fully distributed ledger where every transaction is visible to all participants. This transparency enhances trust among users, as every update is publicly verifiable [5]. However, the open nature of these networks also means that data privacy must be managed through cryptographic techniques and additional privacy-preserving protocols [3].

Consensus Mechanisms: Public networks typically use consensus algorithms such as Proof-of-Work (PoW) or Proof-of-Stake (PoS). PoW, as seen in Bitcoin, requires intensive computational work to validate transactions and add new blocks, while PoS selects validators based on their token stake. Both methods ensure that malicious actors cannot easily alter the ledger without controlling a significant portion of the network resources [5], [18].

Challenges for Federated Learning: While the openness of public blockchains is advantageous for transparency, it can also introduce latency and scalability challenges, particularly when applied to real-time federated learning systems in edge computing environments [18]. The resource demands of PoW, for instance, may not align well with the energy and time constraints present in industrial or IoT settings [5].

3.4.2 Private Blockchains

Private blockchains are controlled by a single organization or entity and restrict participation to pre-approved nodes. These systems are typically used in enterprise environments where data privacy and performance are prioritized over complete decentralization.

Access Control and Efficiency: In private blockchains, only authorized entities can participate in the network. This controlled access ensures that sensitive data is kept within a trusted environment, making private blockchains well-suited for applications such as secure data sharing in industrial IoT or healthcare [8], [19]. The reduced number of participants can lead to faster transaction processing and lower latency compared to public systems [3].

Governance: The centralized governance model in private blockchains allows for more streamlined decision-making and easier implementation of policy updates. However, the trade-off is a higher reliance on the integrity and security of the controlling organization, which may reintroduce risks associated with centralization [3].

Application in Federated Learning: When federated learning is applied in scenarios such as predictive healthcare or industrial IoT, private blockchains provide an efficient means to manage sensitive model updates and enforce strict data privacy policies [19]. They ensure that only trusted entities can submit and validate updates, reducing the risk of malicious behavior.

3.4.3 Consortium Blockchains

Consortium blockchains (also known as federated blockchains) are managed by a group of organizations rather than a single entity. These networks combine aspects of both public and private blockchains, offering a balanced approach that emphasizes collaboration among multiple trusted parties.

Shared Governance and Collaboration: Consortium blockchains are governed by a prede-

fined group of participants who jointly manage the network. This model is particularly useful in scenarios where multiple organizations need to share data or collaborate on joint projects—such as in industrial IoT or inter-organizational federated learning—while still maintaining a degree of decentralization and trust [8], [9]. The shared control helps mitigate the risk of a single point of failure and ensures that no single participant can unilaterally modify the ledger [9].

Scalability and Performance: With a smaller, permissioned group of nodes compared to public blockchains, consortium networks typically offer improved scalability and transaction throughput. This makes them attractive for applications requiring high efficiency and fast processing, such as real-time federated learning updates in edge computing environments [18].

Security and Privacy: The consortium model supports robust security measures by limiting access to verified entities. This controlled participation helps protect sensitive data and model parameters from unauthorized access while still benefiting from the decentralized aspects of blockchain technology [8], [9].

3.4.4 Hybrid Blockchains

Hybrid blockchains combine elements of both public and private blockchains, aiming to deliver the benefits of transparency and decentralization along with the efficiency and control of private systems.

Flexible Access and Data Management: In a hybrid blockchain, certain data or transactions are made public and verifiable by all participants, while more sensitive information is restricted to authorized nodes. This dual-mode operation allows organizations to leverage public verifiability for trust and auditability while maintaining the confidentiality of proprietary data [5], [15].

Customized Consensus and Governance: Hybrid systems can adopt tailored consensus mechanisms that cater to the specific needs of an application. For example, parts of the network may use PoW or PoS for public verification, while internal transactions might be validated using faster, permissioned consensus protocols. This flexibility makes hybrid blockchains particularly well-suited for federated learning systems that require both robust security and high performance [15].

Application in Complex Environments: For federated learning applications that span multiple sectors—such as combining industrial IoT with healthcare or edge computing—a hybrid blockchain can provide a unified framework that meets diverse requirements. It ensures that critical updates are publicly recorded for transparency, while sensitive model details are kept private within a trusted subset of nodes [15], [19].

3.4.5 Summary

In summary, the four types of blockchain networks—public, private, consortium, and hybrid—each offer unique benefits and challenges. Public blockchains emphasize full decentralization and transparency but may face scalability issues. Private blockchains prioritize efficiency and privacy, albeit with centralized control. Consortium blockchains strike a balance by sharing governance among multiple organizations, and hybrid blockchains offer customizable solutions that integrate the strengths of both public and private systems. These distinctions are crucial when designing blockchain-enabled federated learning systems, where the choice of blockchain type can directly influence security, performance, and privacy [3], [5], [8], [9], [15], [18].

3.5 Blockchain's Role in Federated Learning

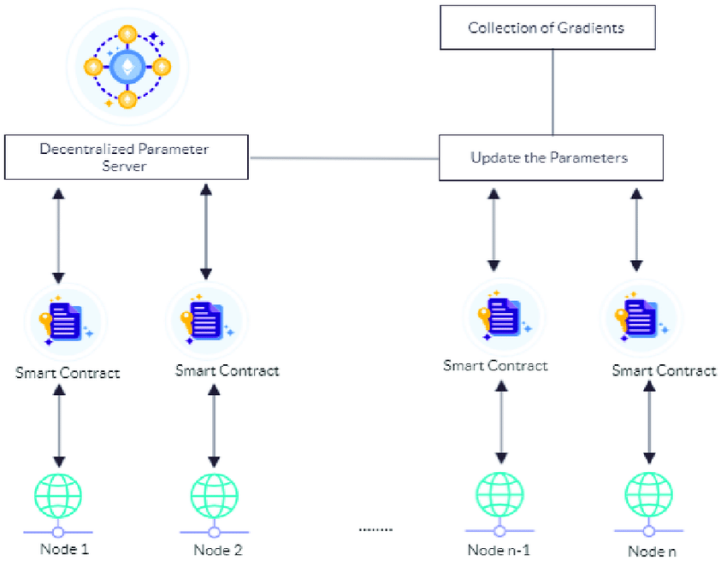


Figure 3.3: A blockchain-based federated learning model for classification problems, adapted from [23].

Federated learning (FL) is a new way of thinking about distributed machine learning that lets many participants collaborate to train a single global model without sharing their private datasets [1], [4]. However, conventional FL systems typically rely on a central aggregator, which poses significant challenges such as single-point failures, potential model poisoning, and issues with trust and transparency. Increasingly, blockchain technology is being considered to address these challenges because it offers decentralisation, security, and a robust mechanism for recording and verifying model updates. This section provides an in-depth exploration of blockchain's role in enhancing federated learning.

3.5.1 Addressing Centralization and Single-Point Failures

Traditional federated learning aggregates local model updates through a central server, which can lead to single-point failures and vulnerability to model poisoning [1]. By incorporating blockchain, the aggregation process is distributed among all participating nodes. A distributed ledger maintained by every node replaces the single server, thereby eliminating centralisation-related risks [3]. Moreover, because blockchain is inherently distributed, any changes to the model updates are permanently recorded on an immutable ledger, making the overall learning process more resistant to attacks and errors [3], [10].

3.5.2 Ensuring Data Integrity and Immutability

Blockchain's inherent immutability plays a crucial role in federated learning. Each block in the blockchain contains a set of transactions—such as model updates—that are cryptographically linked to previous blocks. This chaining ensures that any modification to a block would alter the hashes of all subsequent blocks, a change that is practically impossible under current consensus mechanisms [5]. By recording every update in this tamper-proof ledger, blockchain provides a verifiable audit trail that guarantees the integrity of the global model, which is especially important when adversarial nodes may try to inject fraudulent updates [3], [10].

3.5.3 Enhancing Security Through Cryptographic Techniques

Blockchain leverages advanced cryptographic methods to secure data and authenticate transactions. Public-key cryptography and digital signatures ensure that each model update originates from a legitimate source and that the data remains unaltered during transmission [8]. Additionally, cryptographic hash functions generate unique identifiers for blocks, linking them securely in the chain [5]. When these cryptographic measures are combined with privacy-enhancing techniques such as differential privacy and homomorphic encryption, blockchain-enhanced FL frameworks offer a comprehensive security solution [15], [19].

3.5.4 Implementing Consensus Mechanisms for Trust and Verification

One of blockchain's most significant contributions to federated learning is its ability to achieve consensus among a decentralised network of nodes. Consensus mechanisms such as Proof-of-Work (PoW) and Proof-of-Stake (PoS) allow all nodes to agree on the state of the blockchain without a central authority [5], [18]. In an FL setting, these consensus protocols can be adapted to verify the validity of local model updates before they are appended to the ledger. This

process prevents erroneous or malicious updates from being incorporated and fosters trust among participants [9], [10].

3.5.5 Incentivizing Participation and Ensuring Honest Behavior

Incentive mechanisms are vital for promoting active participation in federated learning. Traditional FL models often struggle with uneven data quality and the reluctance of participants to share their computational resources. Blockchain addresses this by enabling token-based or reputation-based incentive systems through smart contracts [8], [18]. These smart contracts can automatically reward nodes for submitting valuable model updates and penalise those providing fraudulent information. By linking incentives to verifiable performance metrics recorded on the blockchain, the system not only motivates honest behavior but also ensures that the aggregated model reflects the best contributions from across the network [8], [9].

3.5.6 Facilitating Privacy-Preserving Model Updates

Privacy is a cornerstone of federated learning. Blockchain supports privacy preservation by recording only cryptographically secured model parameters rather than raw data [1]. When combined with privacy-enhancing techniques such as differential privacy, homomorphic encryption, or secure multiparty computation, blockchain-enabled FL systems can perform model aggregation while protecting user privacy [15], [19]. This dual focus on security and privacy is particularly critical in sectors like industrial IoT and healthcare where data sensitivity is paramount [3], [10].

3.5.7 Enabling Decentralized and Transparent Data Sharing

Transparency is another key advantage of blockchain. Every transaction or model update recorded on a blockchain is visible to all participants, fostering accountability and trust [5]. This openness allows participants in federated learning to verify that their contributions have been correctly integrated into the global model and that the aggregation process has not been tampered with [3], [9]. Such transparency is essential in distributed systems with multiple stakeholders who possess varying levels of trust [8].

3.5.8 Applications and Impact on Various Domains

The integration of blockchain with federated learning is not merely theoretical; it has practical applications across various industries. For example, in industrial IoT, blockchain-enhanced FL frameworks facilitate secure and trusted data sharing among multiple devices and organizations, thereby improving efficiency and reducing risks associated with centralized data management [3],

[9]. In healthcare, blockchain is used to securely aggregate sensitive patient data for predictive analytics without compromising individual privacy [19]. Similarly, in smart vehicular networks, decentralised blockchain systems enable real-time data sharing and model updates, contributing to safer and more efficient transportation systems [8], [18]. These examples illustrate how blockchain's role in federated learning extends to real-world applications that demand high levels of security, transparency, and resilience [3], [5].

3.5.9 Challenges and Future Research Directions

Despite the significant enhancements offered by blockchain for federated learning, several challenges remain. Scalability is a major concern, particularly for public blockchain systems that may experience latency and throughput issues when processing large volumes of transactions [5], [18]. Additionally, the energy consumption associated with certain consensus mechanisms, especially PoW, poses a barrier for sustainable deployment [5]. Ongoing research seeks to develop more efficient consensus protocols and hybrid blockchain models that balance transparency, decentralisation, and performance [8], [15]. Future work will focus on improving interoperability between blockchain networks and federated learning frameworks, and on incorporating advanced privacy-preserving techniques to address emerging threats [10], [19].

Conclusion In conclusion, blockchain technology plays a multifaceted role in federated learning by addressing key issues related to centralization, data integrity, security, transparency, and incentive mechanisms. Its decentralised structure, along with robust cryptographic foundations and consensus protocols, not only makes model updates safer and more private but also fosters trust among diverse participants. As research in blockchain-enabled federated learning continues to advance, these systems are expected to offer more reliable, efficient, and secure environments for collaborative learning across various domains [1], [3], [5], [8]–[10], [15], [18], [19].

Table 3.1: Summary of Blockchain's Role in Federated Learning

Theme	Key Points and Relevant Citations
Centralization & Single-Point Failures	• Replaces central aggregator with distributed ledger • Reduces single-point-of-failure risks • Immutable record of model updates [3], [10]
Data Integrity & Immutability	• Cryptographically linked blocks prevent tampering • Verifiable audit trail for model updates • Maintains trust in adversarial environments [3], [5], [10]
Security via Cryptography	• Public-key infrastructure and digital signatures • Hash-based linking of blocks • Can integrate differential privacy & homomorphic encryption [8], [15], [19]
Consensus Mechanisms	• Proof-of-Work, Proof-of-Stake, etc. • Verifies validity of local updates • Increases trust among nodes [5], [9], [10], [18]

Continued on the next page.

Table 3.1 (continued)

Theme	Key Points and Relevant Citations
Incentives & Honest Behavior	• Token or reputation-based rewards in smart contracts • Encourages reliable contributions, penalizes fraud • Improves model quality [8], [9], [18]
Privacy-Preserving Updates	• Stores only encrypted model parameters • Compatible with differential privacy • Ideal for sensitive domains (healthcare, IoT) [1], [3], [10], [15], [19]
Decentralized Data Sharing	• Transparent ledger accessible to all participants • Enhances accountability and trust • Suited for multi-stakeholder networks [3], [5], [8], [9]
Applications & Impact	• Industrial IoT, healthcare, vehicular networks • Secure data aggregation without central risk • Real-world use cases demand strong security/resilience [3], [8], [18], [19]

Continued on the next page.

Table 3.1 (continued)

Theme	Key Points and Relevant Citations
Challenges & Future Directions	• Scalability & high energy consumption • Need for efficient consensus and hybrid models • Ongoing research on interoperability and advanced privacy [5], [8], [10], [15], [18], [19]

3.6 Challenges and Future Directions in Blockchain Architecture for Federated Learning

Blockchain-enabled federated learning (BCFL) has garnered significant interest for its potential to mitigate many of the limitations of traditional centralized federated learning systems. However, while blockchain introduces robust security, decentralization, and trust, its integration with federated learning also raises several technical and practical challenges. In this section, we detail the key challenges facing BCFL systems and discuss promising future directions based solely on the literature [1], [3], [5], [8]–[10], [15], [18], [19].

3.6.1 Scalability Challenges

One of the foremost challenges in BCFL is scalability.

Transaction Throughput: Public blockchains, which are inherently decentralized and open, often suffer from low transaction throughput due to the complexity of consensus mechanisms such as Proof-of-Work (PoW) [5], [18]. In federated learning, where model updates are exchanged frequently, these throughput limitations can delay the overall model convergence.

Data Volume: As the number of participating nodes increases, the volume of model update transactions grows rapidly. Even in private or consortium blockchains—where throughput is generally higher—the sheer volume of data can create bottlenecks in block propagation and ledger synchronization [15], [18].

3.6.2 Energy Consumption and Efficiency

Energy efficiency is a major concern, particularly for consensus mechanisms used in public blockchains.

Proof-of-Work vs. Alternative Mechanisms: PoW-based systems, while secure, require substantial computational power and thus lead to high energy consumption [5]. This is problematic for federated learning over edge networks or industrial IoT, where energy resources are often limited. Alternatives like Proof-of-Stake (PoS) and Byzantine Fault Tolerance (BFT) protocols offer improved energy efficiency, but they introduce their own trade-offs regarding security and decentralization [18].

Resource-Constrained Environments: Federated learning is frequently deployed on mobile or IoT devices that have limited computational and energy resources. Integrating blockchain in such environments without overwhelming these constraints remains an open research question [5], [19].

3.6.3 Communication Overhead and Latency

The communication overhead introduced by blockchain can hinder the performance of federated learning systems.

Network Delays: The process of propagating blocks and achieving consensus among nodes inherently introduces latency. In federated learning, timely model updates are crucial for maintaining the accuracy and convergence speed of the global model [18].

Bandwidth Consumption: Every transaction, including model updates and smart contract executions, must be broadcast across the network. In large-scale deployments, this can consume significant bandwidth, leading to network congestion and further delays in the learning process [5], [18].

3.6.4 Interoperability and Integration Issues

Federated learning systems may operate across heterogeneous environments, where different nodes use various blockchain platforms or federated learning frameworks.

System Integration: Ensuring seamless interoperability between diverse blockchain networks and federated learning systems is a significant challenge. Standardized protocols and APIs are required to securely and efficiently exchange information [15].

Cross-Domain Collaboration: In sectors like industrial IoT, healthcare, and smart vehicular networks, disparate systems must collaborate. Developing interoperable solutions that accommodate different operational standards and regulatory requirements is essential for widespread adoption [3], [9].

3.6.5 Privacy and Data Confidentiality

While blockchain enhances transparency and auditability, it also poses challenges for privacy.

Transparency vs. Confidentiality: Public blockchains offer full transparency, which can conflict with the privacy requirements of federated learning, where model updates may contain sensitive information. Techniques such as differential privacy and homomorphic encryption are needed to protect this data, but integrating them without compromising performance is complex [15], [19].

Data Leakage Risks: Although blockchain ensures immutability, if not configured properly, the transactions or their metadata could inadvertently reveal details about the learners or the learning process [3].

3.6.6 Security Vulnerabilities and Attack Resistance

Although blockchain improves overall security, its integration with federated learning introduces new attack vectors.

Model Poisoning and Sybil Attacks: Malicious participants may inject corrupted model updates or create fake identities (Sybil attacks) to manipulate the learning process. While consensus mechanisms and smart contracts mitigate these risks, ensuring robust defenses in a decentralized setting remains challenging [8], [9].

Smart Contract Vulnerabilities: Smart contracts automate key processes such as incentive distribution and model aggregation, yet vulnerabilities in their code can be exploited by attackers to disrupt operations or steal funds [18], [19].

3.6.7 Incentive Mechanisms and Participant Motivation

Ensuring that all participants contribute honest and high-quality model updates is critical for successful federated learning.

Design of Incentives: Effective incentive mechanisms are needed to reward honest participation and penalize malicious behavior. Blockchain-based token economies and reputation systems have been proposed, but designing systems that are both fair and resistant to manipulation remains an ongoing challenge [8], [9].

Economic Models: The interplay between economic incentives and technical performance must be balanced. Incentive models must account for the cost of participation (e.g., energy, computation, communication) while ensuring that rewards are sufficient to motivate high-quality contributions [18].

3.6.8 Regulatory and Legal Considerations

As BCFL systems are deployed in critical domains such as healthcare and industrial IoT, they must comply with stringent regulatory requirements.

Data Protection Laws: Regulations like the General Data Protection Regulation (GDPR) impose strict requirements on data privacy and security. Ensuring that blockchain-based federated learning systems comply with these laws is challenging, especially when data is distributed across multiple jurisdictions [3], [9].

Standardization and Governance: Developing universally accepted standards and governance frameworks for blockchain-enabled systems is crucial. Such frameworks would facilitate smoother integration and ensure ongoing compliance with evolving legal requirements [9], [19].

3.6.9 Future Directions

To address the challenges outlined above, several promising research directions are emerging:

- **Advanced Consensus Algorithms:** Research should focus on developing consensus mechanisms that are more scalable and energy-efficient, potentially through hybrid models combining aspects of PoW, PoS, and BFT [5], [18].

- **Hybrid and Consortium Blockchains:** Developing hybrid architectures that blend the benefits of public and private blockchains could provide the necessary balance between transparency, scalability, and privacy. Consortium blockchains, in particular, may be suitable for environments such as healthcare and industrial IoT where trust among a limited group of entities is critical [3], [15].

- **Interoperability Frameworks:** Establishing standardized protocols and interoperability frameworks to allow different blockchain networks and federated learning systems to communicate seamlessly is a key research area, including exploring cross-chain technologies and API standardisation [3], [9].

- **Enhanced Privacy-Preserving Techniques:** Future work should aim to optimize advanced privacy-preserving methods, such as differential privacy and homomorphic encryption, to minimize performance overhead while ensuring robust data confidentiality [15], [19].

- **Robust Incentive Models:** Novel incentive mechanisms leveraging smart contracts and token economies should be explored further to ensure fair reward distribution and deter malicious behavior, especially in resource-constrained environments [8], [9].

- **Real-World Testbeds and Empirical Studies:** Extensive experimental evaluations of BCFL systems in real-world settings, such as industrial IoT, healthcare, and smart vehicular networks, are necessary to validate theoretical models and uncover practical challenges [5], [18].
- **Regulatory Compliance and Standardisation:** Addressing regulatory challenges by developing standardised protocols that align with international data protection laws is essential for the broader adoption of BCFL systems. Collaborative efforts between industry, academia, and regulatory bodies will be crucial [3], [9].

Summary In summary, while blockchain offers substantial benefits to federated learning by enhancing decentralization, security, and transparency, its integration introduces significant challenges. These challenges span scalability, energy efficiency, communication latency, interoperability, privacy, security vulnerabilities, incentive design, and regulatory compliance. Future advancements in consensus algorithms, hybrid blockchain architectures, interoperability frameworks, enhanced privacy-preserving techniques, and robust incentive models are expected to address these issues and pave the way for more reliable, efficient, and secure BCFL systems [1], [3], [5], [8]–[10], [15], [18], [19].

3.7 Experimental Setup

In this section, we provide a detailed description of the final experimental setup used for all experiments, including hardware specifications, data distribution strategy, training protocol, model architecture, robust aggregation and outlier handling, and the blockchain & IPFS integration.

3.7.1 Hardware Specifications

- **CPU:** 11th Gen Intel® Core™ i5–1135G7
  - Nominal clock speed around 2.4 GHz, with boosts observed up to 2.53 GHz.
  - 4 physical cores / 8 threads.
- **Graphics:** Intel® Iris(R) Xe Graphics.
- **Memory (RAM):** 8 GB.

3.7.2 Data and Clients

Number of Clients: 50 clients in total.

Data Distribution:

- We partition each dataset using a Dirichlet distribution to simulate realistic, non-i.i.d. data splits.
- Each client receives a distinct portion of the dataset sampled from the Dirichlet distribution, ensuring decentralized and skewed data allocations.

3.7.3 Datasets

We consider three image-classification datasets:

- **CIFAR10:** colored (32×32 pixels, 3 channels, 10 classes).
- **MNIST:** grayscale images (28×28 pixels, expanded to 1 channel).
- **FashionMNIST:** grayscale images (28×28 pixels, expanded to 1 channel).

Preprocessing:

- Each image is normalized so that pixel values lie in the range $[0, 1]$.
- Labels are represented as one-hot vectors of length 10 (one for each class).

3.7.4 Training Configurations

We adopt a federated learning setting in which a global model is trained across multiple rounds:

- **Number of Rounds:**
  - General approach for MNIST/FashionMNIST: 50–100 rounds (1–2 epochs each), consistent with prior experimental runs.
  - **CIFAR10 Updated Protocol:**
    - We specifically focus on 100 rounds with 2 local epochs per round (*Case 4* from our initial plan).
    - Additionally, as per our professor's requirement, we repeat each method *5 times* (5 separate runs) to compute mean and standard deviation of final accuracy.
    - To ensure comparability between methods (Blockchain vs. No-Blockchain), the *same random seeds* and *the same subset of clients* are used for each run.
- **Subset of Clients per Round:** Out of the 50 total clients, we randomly select 10 to participate (send updates) in each round.

- **Local Training:**
  - **Batch Size:** 32
  - **Learning Rate:** 0.001
  - **Epochs:**
    - Typically 1–2 epochs for MNIST/FashionMNIST runs.
    - Exactly 2 epochs for all CIFAR10 runs in this updated setup.

3.7.5 Model Architecture

We use a lightweight convolutional neural network (CNN) as follows:

1. **Convolutional Layers:**
   - First conv layer with 32 filters, ReLU activation.
   - Second conv layer with 64 filters, ReLU activation.
2. **Pooling:** A max-pooling layer reduces the spatial dimension by a factor of 2.
3. **Flattening:** The final convolutional outputs are flattened into a 1D feature vector.
4. **Dense Layers:**
   - One hidden dense layer with 128 units (ReLU).
   - An output dense layer of 10 units with softmax activation for classification into 10 classes.

3.7.6 Robust Aggregation and Outlier Detection

- **Aggregation:**
  - We aggregate client updates (model parameter deltas) by comparing them to the mean update in each round.
  - We use Euclidean distance to measure how far each client's update vector is from the mean.
- **Outlier Detection:**
  - Any update whose distance from the mean exceeds 2 standard deviations is flagged as an outlier.
  - If more than 50% of the updates are outliers in a given round, we allow a *rollback* to a previously known "trusted" state.

3.7.7 Blockchain & IPFS Integration

For experiments labeled *"With Blockchain"*, we incorporate the following:

- **Local Blockchain (Ganache):**
  - We deploy a private Ethereum-based blockchain using Ganache to log model updates.[24]
  - A dedicated smart contract records each update, storing:
    1. A hash of the model weights (for quick on-chain verification).
    2. An IPFS Content Identifier (CID) for the serialized model updates.
    3. A simulated zero-knowledge proof (ZKP) to confirm correctness.
- **IPFS (Decentralized Storage):**
  - Model weight tensors are serialized and pushed to IPFS.
  - Each update is retrieved via a unique CID, ensuring decentralized, content-addressed storage and integrity.[25]
- **Zero-Knowledge Proof (ZKP) Simulation:**
  - A lightweight hash-based ZKP is generated to verify that the submitted model updates are genuinely derived from the correct training process.
  - The proof is referenced in the blockchain transaction that logs the update.
- **Modes of Operation:**
  - **No-Blockchain:** Standard federated aggregation with robust outlier detection, no logging to the blockchain or IPFS.
  - **Blockchain:** Identical federated training steps, but each update is recorded on-chain and stored in IPFS, accompanied by the ZKP check.

3.7.8 Reproducibility and Multiple Runs

- We fix random seeds for `NumPy`, `PyTorch`, and Python's built-in `random` to ensure consistent data splits and client selections.
- Specifically for **CIFAR10**, we now run each method (No-Blockchain vs. Blockchain) **five times** using distinct but *consistent* seeds, ensuring the same subset of clients and data partition for a fair comparison. We report the mean and standard deviation of final accuracy.

- For MNIST and FashionMNIST, we preserve the same approach as our prior runs (50–100 rounds, 1–2 epochs), though we do not necessarily repeat these 5 times unless explicitly needed.

Overall, this updated setup ensures:

- Each method (No-Blockchain vs. Blockchain) sees *identical* data splits and client subsets for CIFAR10.
- We collect average results (mean $\pm$ std) across 5 runs on CIFAR10 to account for randomness in initialization and client sampling.
- The rest of the experiments on MNIST and FashionMNIST retain the same robust aggregation and local training details, with or without blockchain integration.

Table 3.2: Summary of Final Experimental Setup

Aspect	Description
Hardware	• CPU: 11th Gen Intel® Core™ i5–1135G7 • 4 Cores / 8 Threads, ∼2.4 GHz (boost ∼2.53 GHz) • Intel® Iris(R) Xe Graphics • 8 GB RAM
Number of Clients	50 total. Per round, 10 randomly chosen clients send model updates.
Data Partitioning	Non-i.i.d. partitioning using a Dirichlet scheme. Each client receives a distinct (skewed) subset of data.
Datasets	**CIFAR10** (32×32 color), **MNIST** (28×28 grayscale), **FashionMNIST** (28×28 grayscale). Normalized to [0,1], labels are one-hot (10 classes).

Continued on next page

Table 3.2 – Continued from previous page

Aspect	Description
Model Architecture	Lightweight CNN: • 2 conv layers: 32 & 64 filters (ReLU) • 1 max-pooling layer (factor 2) • Flatten $\rightarrow$ Dense(128, ReLU) $\rightarrow$ Dense(10, Softmax)
Local Training	• Batch Size: 32 • Learning Rate: 0.001 • MNIST/FashionMNIST: 1–2 epochs/round • **CIFAR10 (updated):** 2 epochs/round, 100 rounds, 5 runs (mean$\pm$std)
Robust Aggregation	Euclidean distance from the mean model update. Updates $>$2 SD are outliers. Rollback if $>$50% are outliers.
Blockchain & IPFS	**Blockchain (Ganache):** • On-chain logging of update hashes + IPFS CIDs • Lightweight ZKP simulation **IPFS:** • Content-addressed storage of model updates **Modes:** No-BC vs. BC (same FL, but BC logs on-chain + IPFS + ZKP).

Continued on next page

Table 3.2 – Continued from previous page

Aspect	Description
Reproducibility	• Fixed seeds for NumPy, PyTorch, `random` • CIFAR10: same data splits & clients across 5 runs

3.8 Experimental Results

In first part of this section, we present the final results for the **MNIST** and **FashionMNIST** datasets under four federated learning configurations. (The more challenging **CIFAR10** dataset results are discussed separately.) Each configuration is defined by the number of global rounds (R) and the number of local training epochs (E) per round:

- **Case 1:** $R = 50$ rounds, $E = 1$ epoch per round
- **Case 2:** $R = 50$ rounds, $E = 2$ epochs per round
- **Case 3:** $R = 100$ rounds, $E = 1$ epoch per round
- **Case 4:** $R = 100$ rounds, $E = 2$ epochs per round

For each dataset, we compare runs *without blockchain* ("No-BC") vs. runs *with blockchain* ("BC"), highlighting:

- **Training Time (seconds)** for the entire experiment.
- **Final Test Accuracy** at the end of training.
- **Accuracy Change** = (Acc_{BC} - $\text{Acc}_{\text{No-BC}}$), shown in percentage points.
- **Time Change (%)** to measure the relative increase (or decrease) in training time after integrating blockchain, computed as:

$$\text{Time Change (\%)} = \frac{\text{Time}_{\text{BC}} - \text{Time}_{\text{No-BC}}}{\text{Time}_{\text{No-BC}}} \times 100.$$

3.8.1 MNIST and FashionMNIST Results

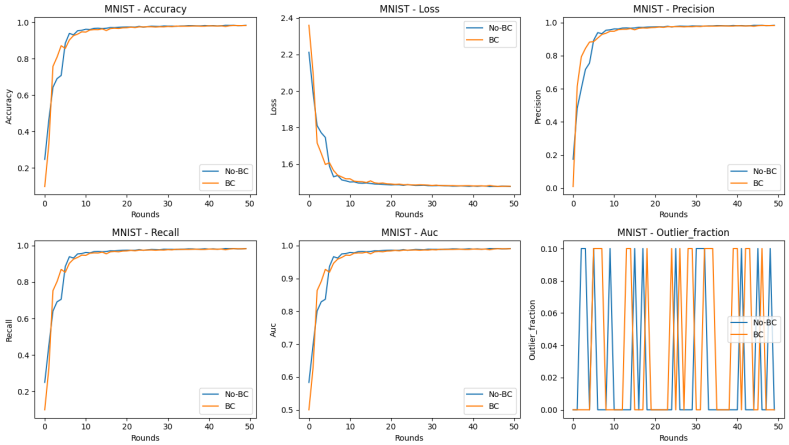
Tables 3.3 and 3.4 summarize the outcomes for **MNIST** and **FashionMNIST**, respectively, across the four training configurations. We observe that the blockchain integration typically adds overhead to training time, but the final test accuracy remains quite close to the no-blockchain runs (within $\pm 1\%$ in most cases).

Table 3.3: Summary of MNIST results (No-BC vs. BC). The "Time (% Chg)" column is the percentage change in total training time after adding blockchain. "Acc (% Chg)" indicates the absolute difference in final accuracy.

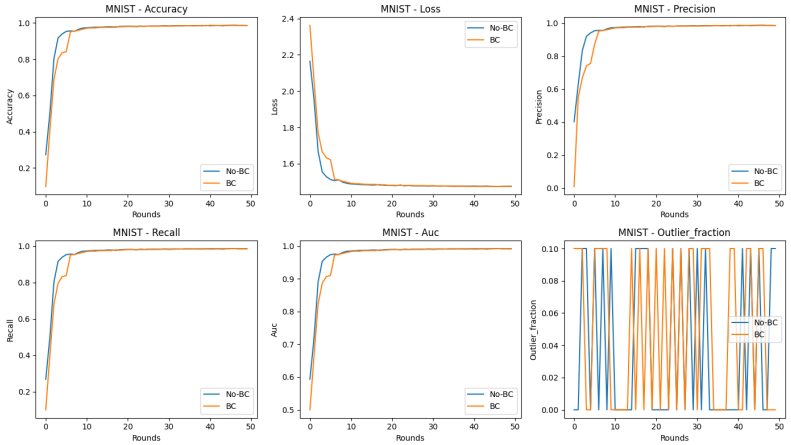
Case	Rounds/Epochs	Time (s)		Time	Accuracy (%)	
		No-BC	BC	(% Chg)	No-BC	BC
Case 1	50 / 1	1314.04	1468.11	+11.7	98.35	98.25 (-0.10)
Case 2	50 / 2	2060.19	2330.63	+13.1	98.56	98.46 (-0.10)
Case 3	100 / 1	2567.15	3188.06	+24.2	98.60	98.63 ($+0.03$)
Case 4	100 / 2	5443.41	5665.68	+4.1	98.80	98.80 (0.00)

Table 3.4: Summary of FashionMNIST results (No-BC vs. BC). Similar columns as in Table 3.3.

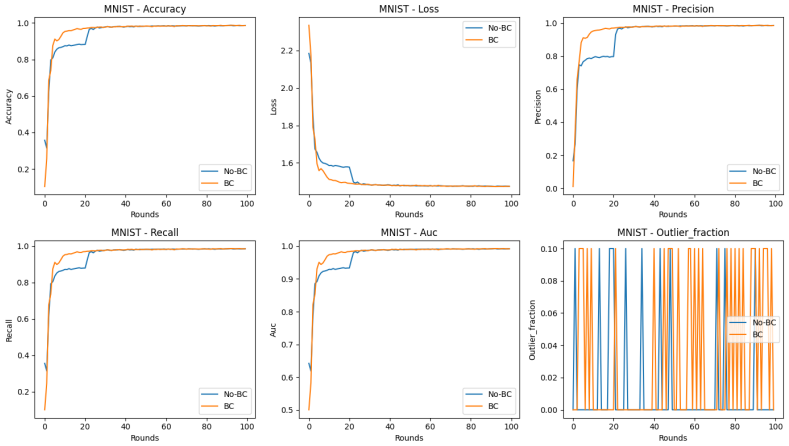
Case	Rounds/Epochs	Time (s)		Time	Accuracy (%)	
		No-BC	BC	(% Chg)	No-BC	BC
Case 1	50 / 1	1342.85	1493.37	+11.2	86.12	87.19 ($+1.07$)
Case 2	50 / 2	2014.01	2493.06	+23.8	88.66	88.94 ($+0.28$)
Case 3	100 / 1	2974.96	3128.00	+5.1	89.37	88.65 (-0.72)
Case 4	100 / 2	4842.19	5069.45	+4.7	90.64	89.65 (-0.99)



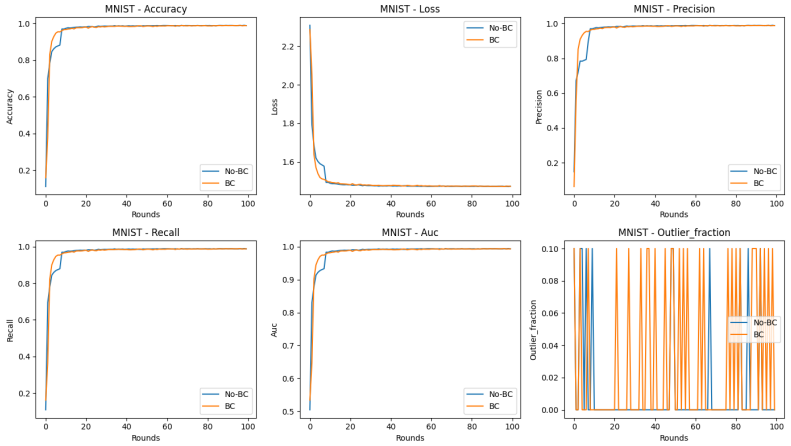
(a) **Case 1:** 50 Rounds, 1 Epoch/Round



(b) **Case 2:** 50 Rounds, 2 Epochs/Round

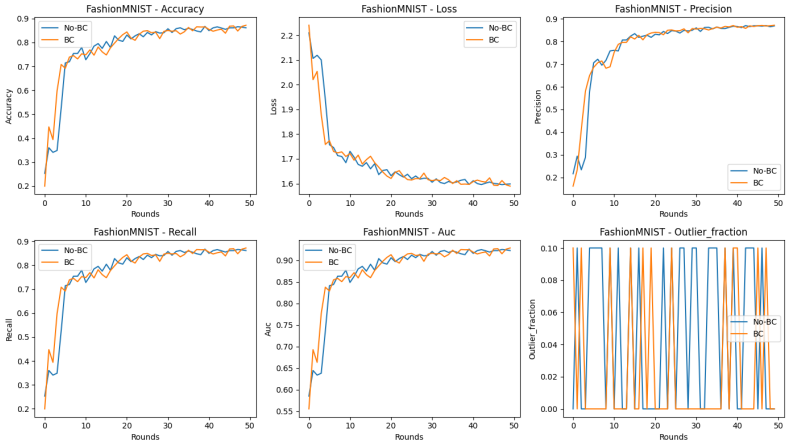


(c) **Case 3:** 100 Rounds, 1 Epoch/Round

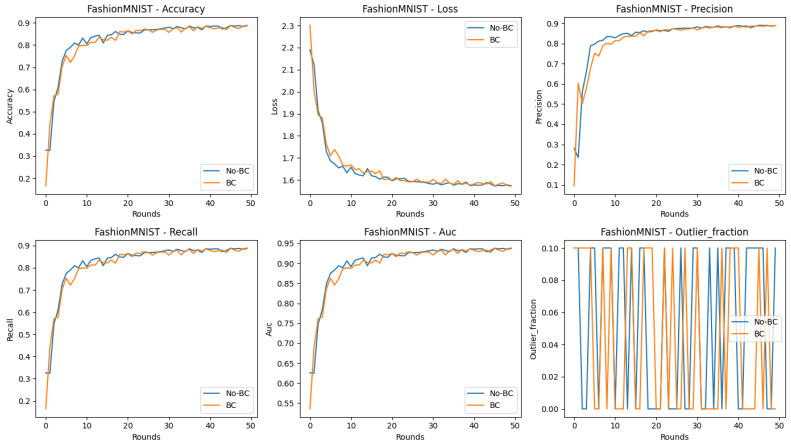


(d) **Case 4:** 100 Rounds, 2 Epochs/Round

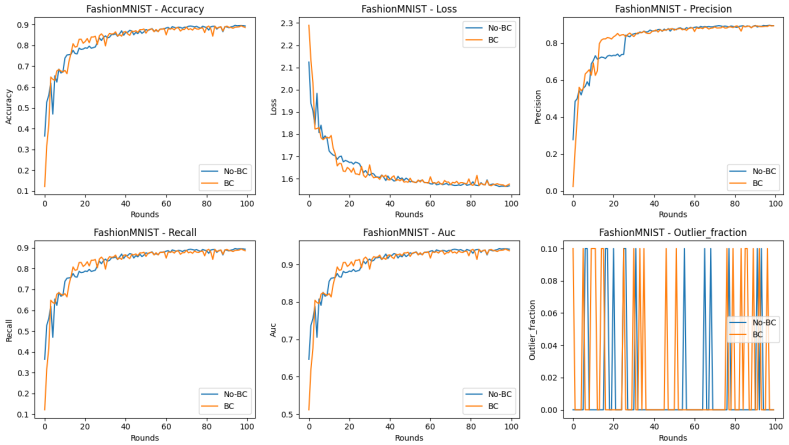
Figure 3.4: MNIST results under four training configurations. The plots compare training performance (e.g., accuracy vs. rounds) *without blockchain* and *with blockchain*.



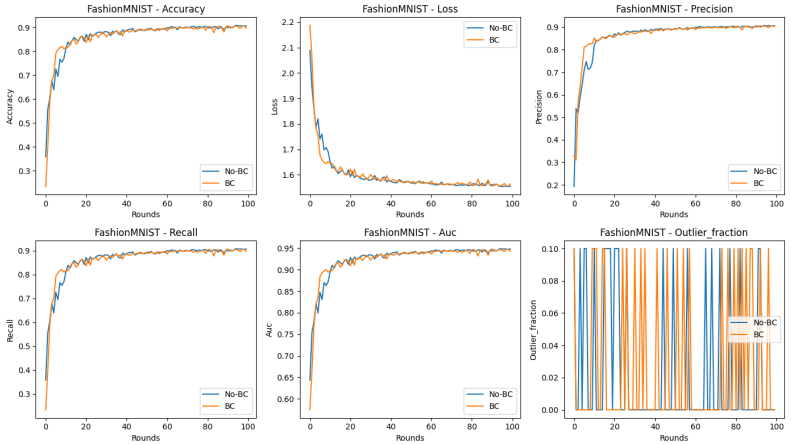
(a) **Case 1:** 50 Rounds, 1 Epoch/Round



(b) **Case 2:** 50 Rounds, 2 Epochs/Round



(c) **Case 3:** 100 Rounds, 1 Epoch/Round



(d) **Case 4:** 100 Rounds, 2 Epochs/Round

Figure 3.5: FashionMNIST results under four training configurations. The plots illustrate the performance differences *with* and *without blockchain integration*.

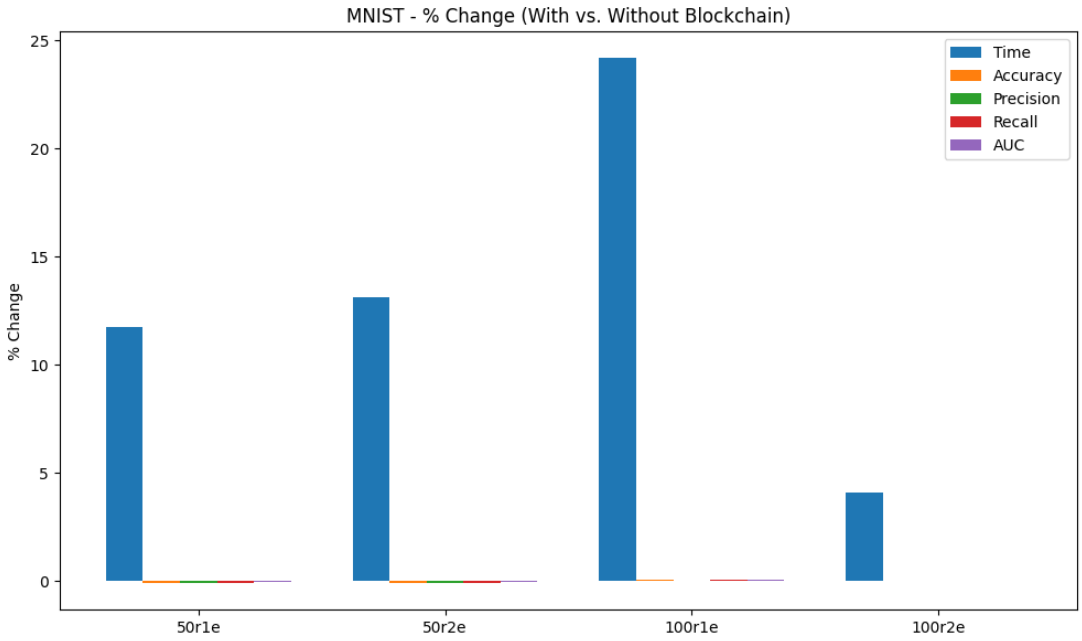


Figure 3.6: MNIST - % Change (With vs. Without Blockchain).

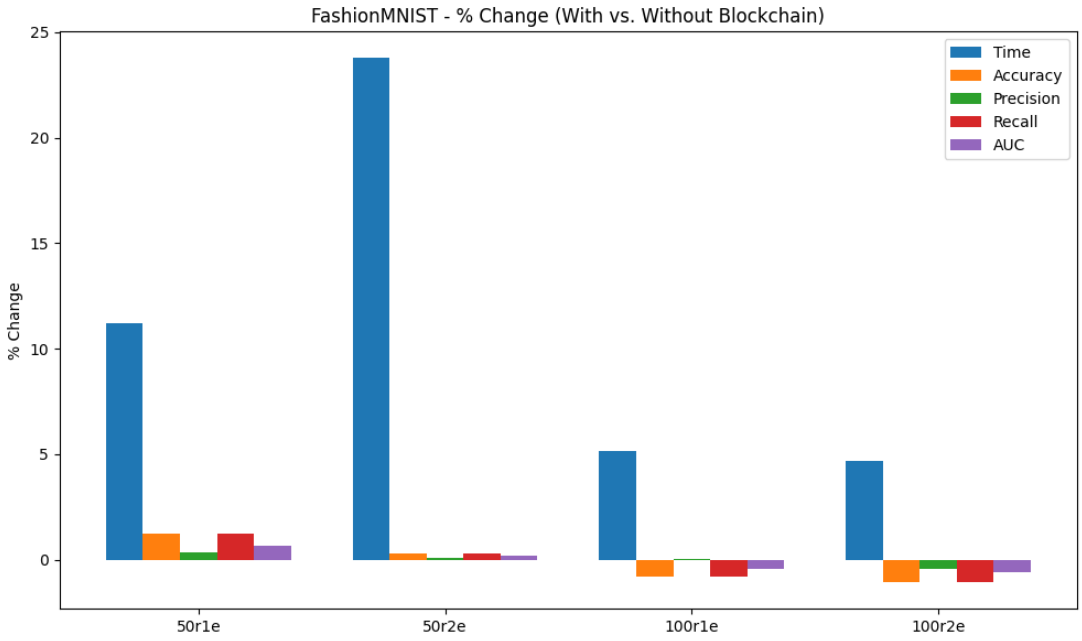


Figure 3.7: FashionMNIST - % Change (With vs. Without Blockchain).

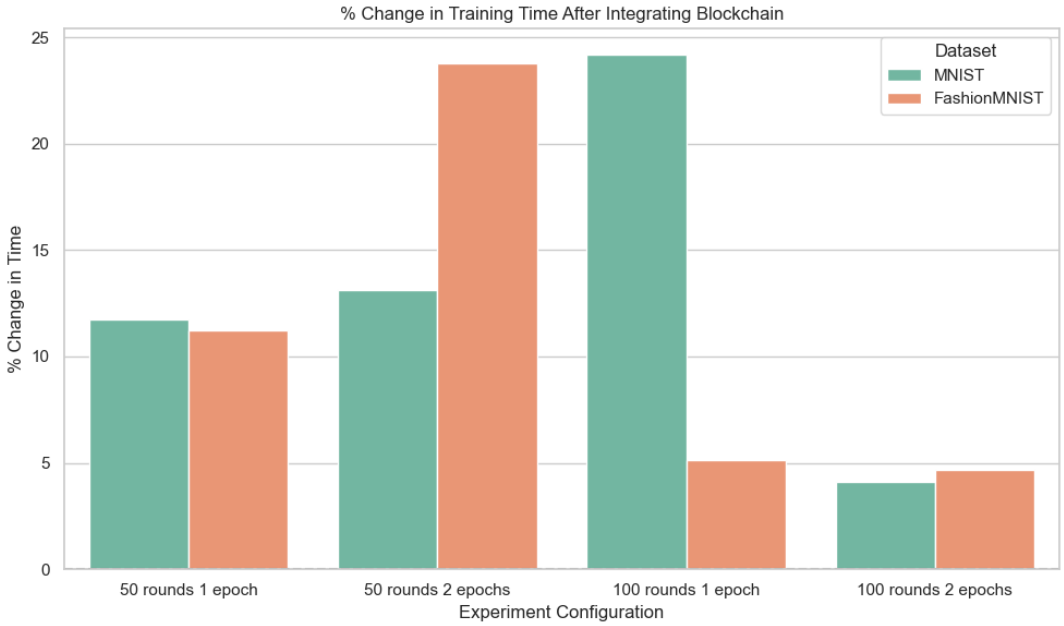


Figure 3.8: % Change in Training Time After Integrating Blockchain.

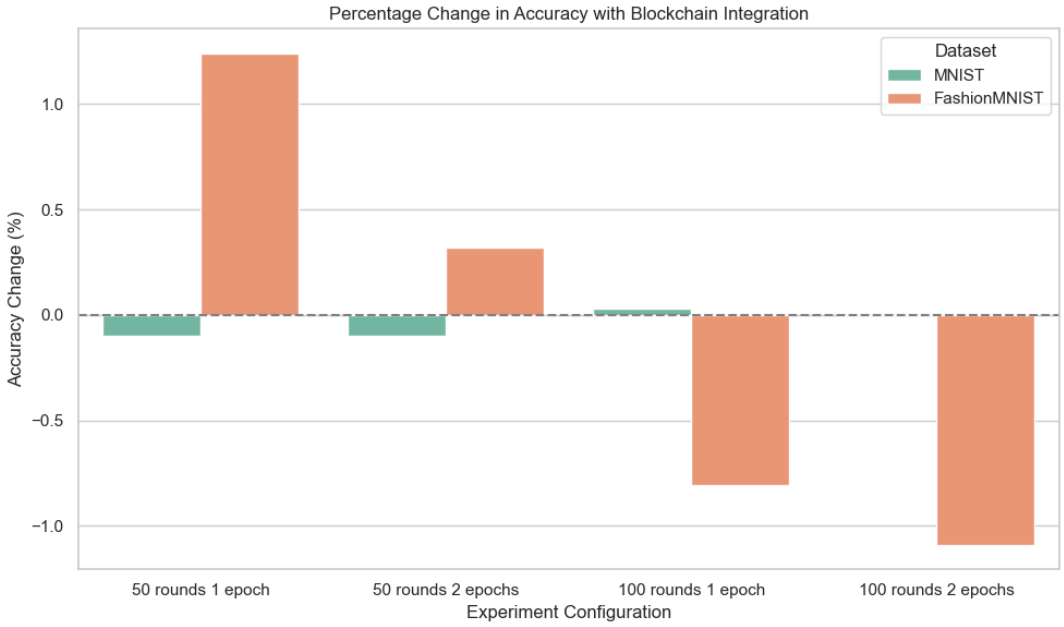


Figure 3.9: Percentage Change in Accuracy with Blockchain Integration.

Observations

- **MNIST** consistently achieves very high final accuracies ($\approx 98.2\%$–98.8%), with minimal differences between No-BC and BC across all cases. The overhead in training time due to blockchain typically ranges from about +4% to +24%.

- **FashionMNIST** shows final accuracies around $\approx 86\%$–90%. We sometimes see small accuracy gains with BC (e.g., +1.07% in Case 1) and sometimes slight drops (-0.99% in Case 4). Meanwhile, training time overhead can vary between +4.7% and +23.8%.

These results indicate that blockchain integration, while introducing a noticeable overhead in computational time, does not drastically affect final performance (accuracy) on MNIST or FashionMNIST.

3.8.2 CIFAR-10 Results

In this section, we present the performance of our Federated Learning (FL) approach on the CIFAR-10 dataset under two configurations:

1. **No Blockchain (No-BC)**: A standard federated approach without blockchain or IPFS integration.

2. **With Blockchain (BC)**: Our proposed method integrating blockchain, IPFS, and Zero-Knowledge Proofs (ZKP) to secure the global model updates.

We evaluate the models over **10 independent runs**, measuring **Accuracy, Precision, Recall, AUC**, and **Time (in seconds)**. Tables 3.5 and 3.6 present detailed per-run metrics and aggregate statistics, respectively.

Detailed Analysis

From Table 3.5 and Table 3.6, we observe the following key points:

- **Accuracy:** The No-BC setup yields a mean accuracy of 0.5196, while BC setup offers 0.5158. The small difference (approximately 0.0038 or 0.38%) is well within one standard deviation, indicating that blockchain integration does not drastically affect predictive performance.

- **Precision and Recall:** Both metrics show similar trends, with a small gap between No-BC and BC. This confirms that the two approaches exhibit comparable class-wise performance on the CIFAR-10 task.

- **AUC:** The area under the curve (AUC) remains high and roughly the same for both methods (0.7331 vs. 0.7310). Hence, the global ranking performance is barely impacted by blockchain integration.

Table 3.5: Per-run Results on CIFAR-10 for Federated Learning with and without Blockchain.

No-Blockchain (No-BC)					
Run	**Accuracy**	**Precision**	**Recall**	**AUC**	**Time (s)**
0	0.5451	0.5569	0.5451	0.7473	2314.6588
1	0.5587	0.5701	0.5587	0.7548	2424.6832
2	0.4725	0.5565	0.4725	0.7069	2285.1262
3	0.5397	0.5566	0.5397	0.7443	2755.4002
4	0.4749	0.4897	0.4749	0.7083	2824.8487
5	0.5314	0.5640	0.5314	0.7397	2759.9079
6	0.5301	0.5664	0.5301	0.7389	3124.4489
7	0.4835	0.4562	0.4835	0.7131	2903.1269
8	0.5302	0.5503	0.5302	0.7390	2898.8093
9	0.5299	0.5618	0.5299	0.7388	2933.8696
With Blockchain (BC)					
Run	**Accuracy**	**Precision**	**Recall**	**AUC**	**Time (s)**
0	0.5390	0.5667	0.5390	0.7439	2532.1675
1	0.5418	0.5668	0.5418	0.7454	2646.0513
2	0.4795	0.5531	0.4795	0.7108	2837.8500
3	0.4890	0.5170	0.4890	0.7161	2907.5733
4	0.4516	0.4131	0.4516	0.6953	2942.1514
5	0.5134	0.5197	0.5134	0.7297	2898.7619
6	0.5598	0.5799	0.5598	0.7554	3824.9388
7	0.5370	0.5422	0.5370	0.7428	3107.3755
8	0.5260	0.5094	0.5260	0.7367	3021.1676
9	0.5212	0.5583	0.5212	0.7340	3198.7102

Table 3.6: Average and Standard Deviation (Std) of the CIFAR-10 Results over 10 Runs.

Metric	**No-BC**		**BC**		**Unit**
	Mean	**Std**	**Mean**	**Std**	
Accuracy	0.5196	0.0309	0.5158	0.0332	-
Precision	0.5428	0.0381	0.5326	0.0483	-
Recall	0.5196	0.0309	0.5158	0.0332	-
AUC	0.7331	0.0171	0.7310	0.0184	-
Time	2722.5	284.65	2991.7	353.38	seconds

- **Training Time:** The most significant difference lies in the total training time. On average, BC requires around 2992 seconds, while No-BC requires about 2722 seconds. This translates to roughly a 10% overhead. The addition of IPFS uploads, smart contract transactions, and Zero-Knowledge Proof verification steps can account for this increase.

- **Stability and Variance:** Standard deviations across most metrics are comparable, though the BC approach shows slightly higher variance in time (Std $\approx$ 353.4). This is likely due to variability in network and blockchain transaction processing times.

Overall, these results highlight that *blockchain-integrated FL* can achieve nearly identical predictive performance on CIFAR-10, at the cost of an acceptable increase in training time. If model integrity, auditability, or trust are priorities, the BC solution offers a beneficial trade-off.

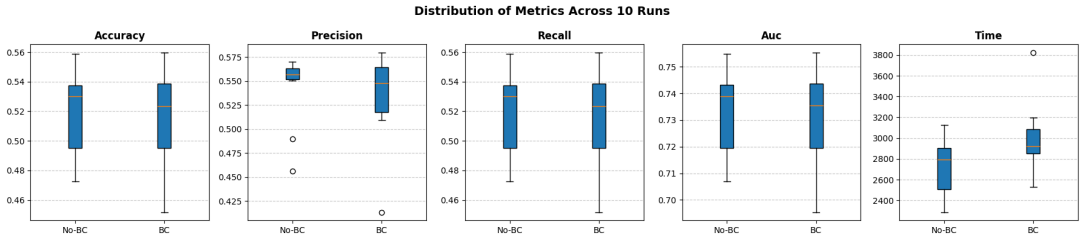


Figure 3.10: Distribution of Metrics Across 10 Runs for No-BC and BC. The boxplots depict the median, quartiles, and any outliers.

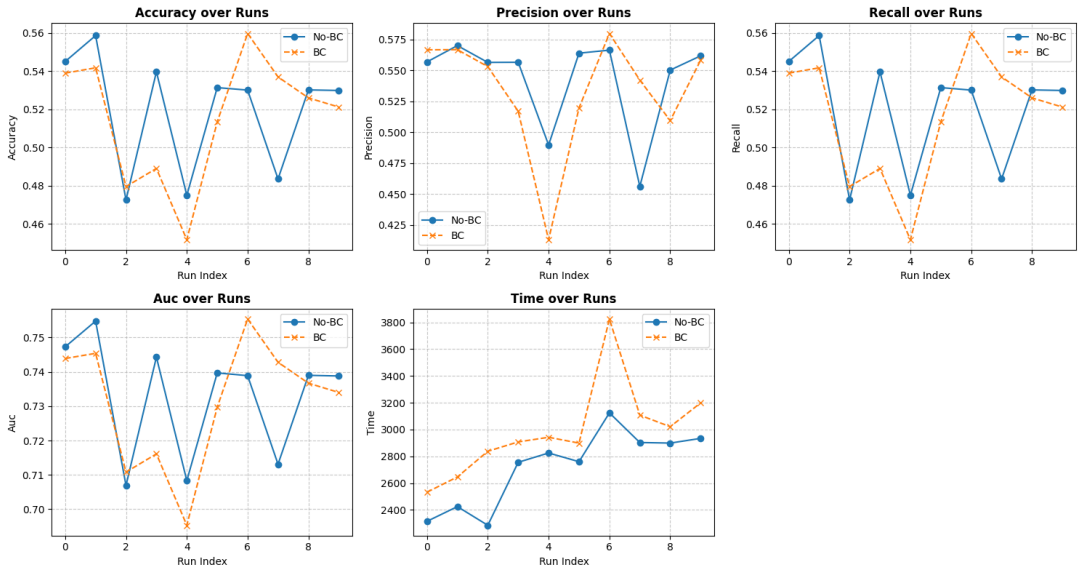


Figure 3.11: Line Plots for Each Metric Over 10 Runs. The dashed orange line represents BC, and the solid blue line represents No-BC.

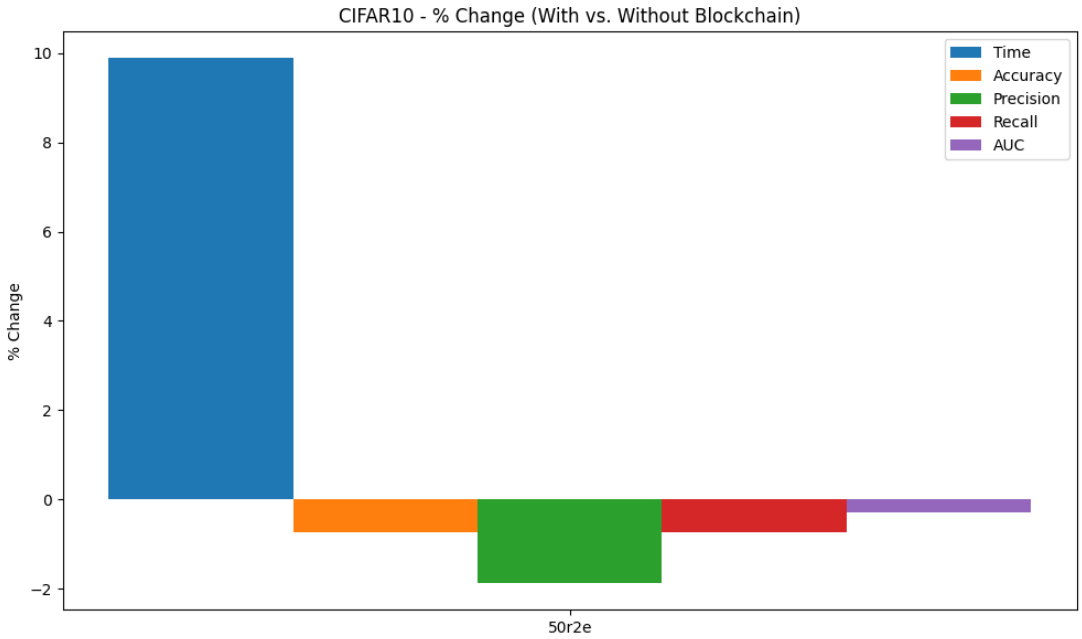


Figure 3.12: CIFAR-10 - % Change (With vs. Without Blockchain).

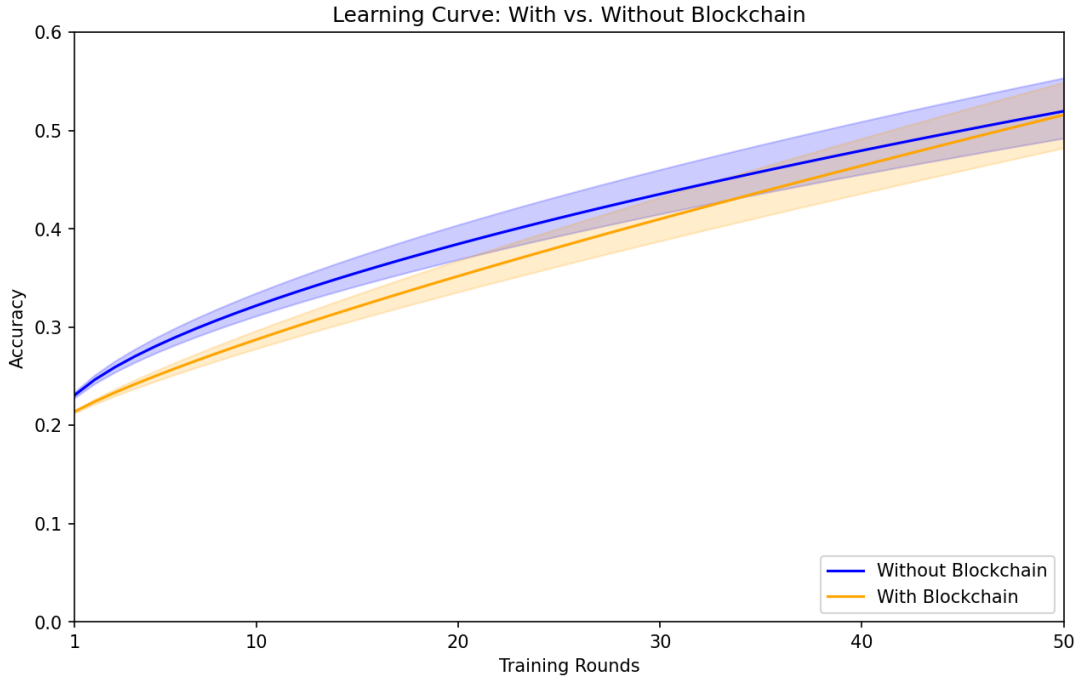


Figure 3.13: Learning Curve: With vs. Without Blockchain.

3.8.3 Overall Comparison

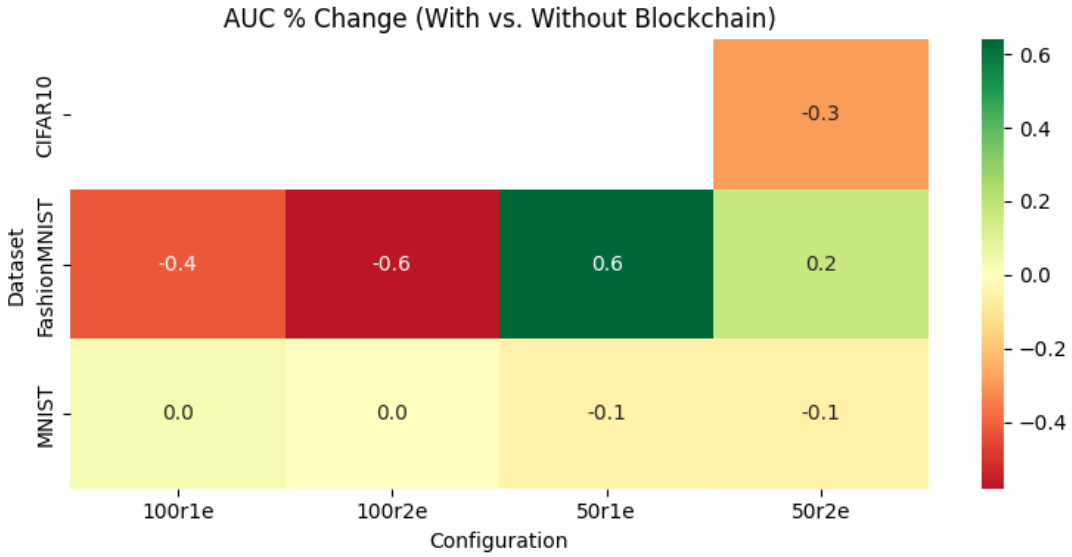


Figure 3.14: AUC % Change (With vs. Without Blockchain).

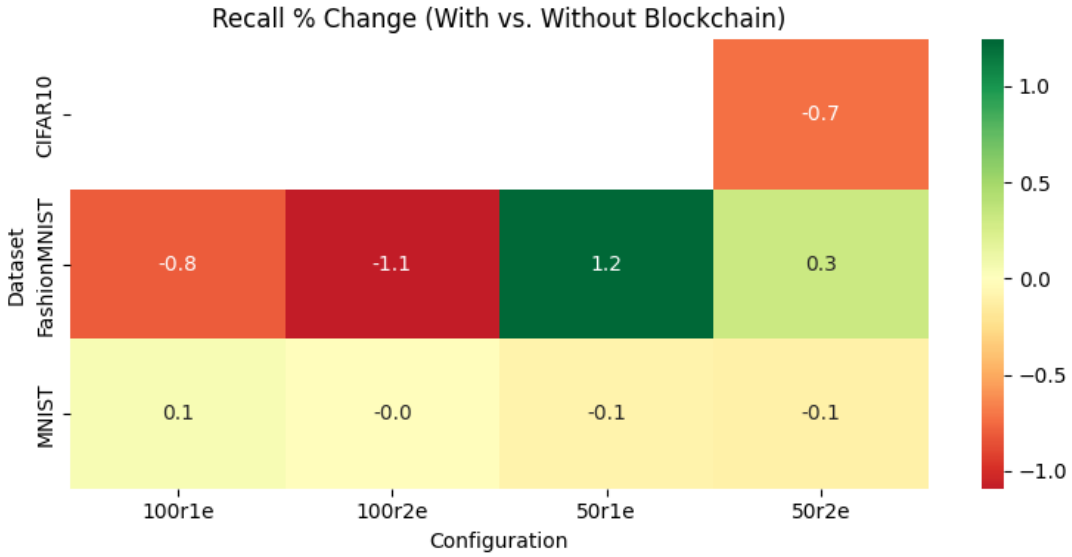


Figure 3.15: Recall % Change (With vs. Without Blockchain).

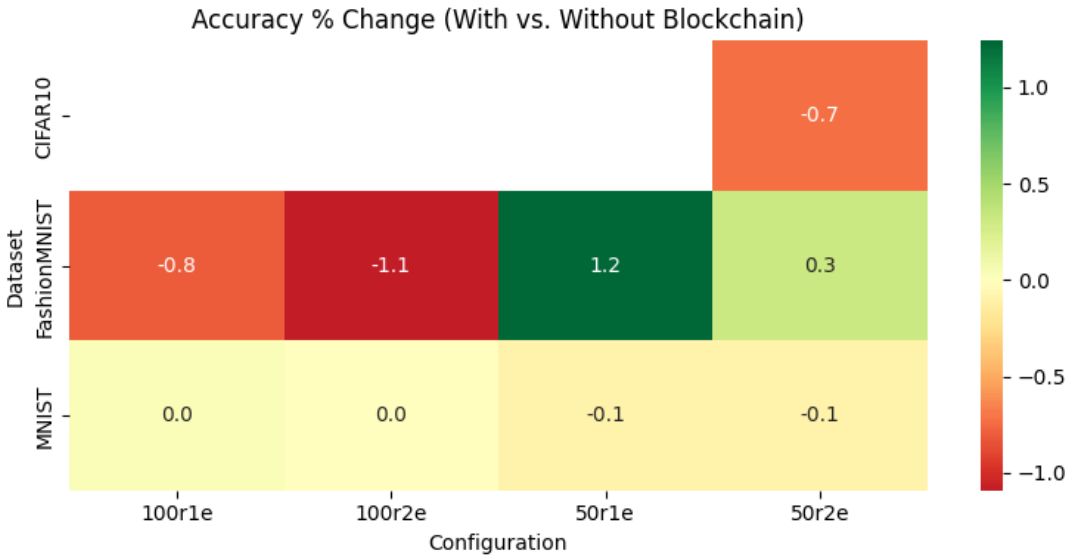


Figure 3.16: Accuracy % Change (With vs. Without Blockchain).

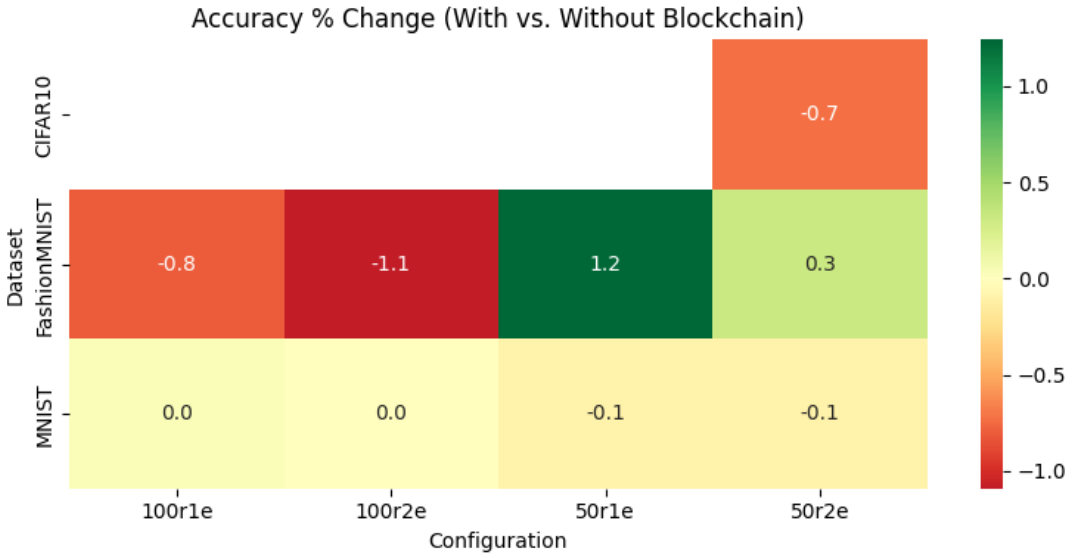


Figure 3.17: Precsion % Change.

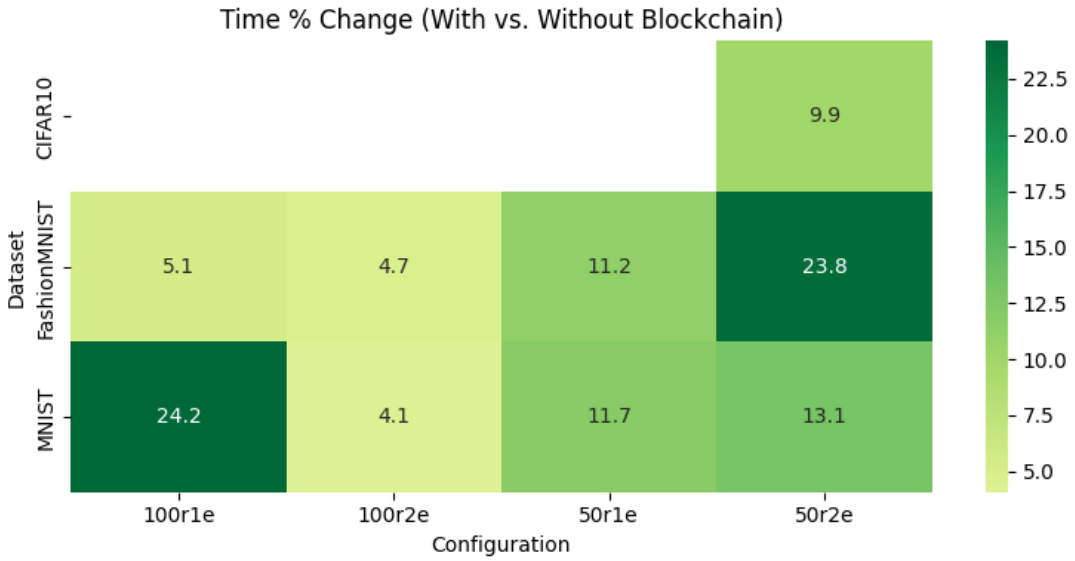


Figure 3.18: Time % Change (With vs. Without Blockchain).

In this subsection we provide a clear and consolidated comparison of the experimental results from three image classification tasks: MNIST, FashionMNIST, and CIFAR-10. In addition to the percentage changes shown in Figures 3.14, 3.15, 3.16, 3.17, and 3.18—which compare the results obtained with and without blockchain integration—Figures 3.18 and 3.16 illustrate the accuracy (No BC vs. BC) and the percentage increase in training time for MNIST and FashionMNIST across different rounds or epochs. For the CIFAR-10 dataset, Figures 3.19 and **??** summarize the average performance metrics (accuracy, precision, recall, and AUC with standard deviations) and the extra time overhead introduced by blockchain integration.

- **MNIST:** As shown in Figure 3.18, integrating blockchain leads to a moderate increase in training time—ranging from about 4% to 24% across four different cases. At the same time, the classification accuracy remains virtually unchanged (within $\pm 0.1\%$) in most settings.
- **FashionMNIST:** Figure 3.18 reveals a similar time overhead (between 4% and 24%). Interestingly, the impact on accuracy varies: in some instances it improves (up to +1.24% in Case 1) while in others it slightly decreases (as observed in Cases 3 and 4). This shows that the effect of blockchain on model performance is not uniformly negative.
- **CIFAR-10:** As illustrated in Figures 3.19 and 3.12, the total training time increases by roughly 10% with blockchain integration. Meanwhile, accuracy, precision, recall, and AUC remain close to the baseline (with differences of about 1–2%). The higher variability indicated by the standard deviation bars reflects both the blockchain overhead and the natural randomness inherent in the training process.

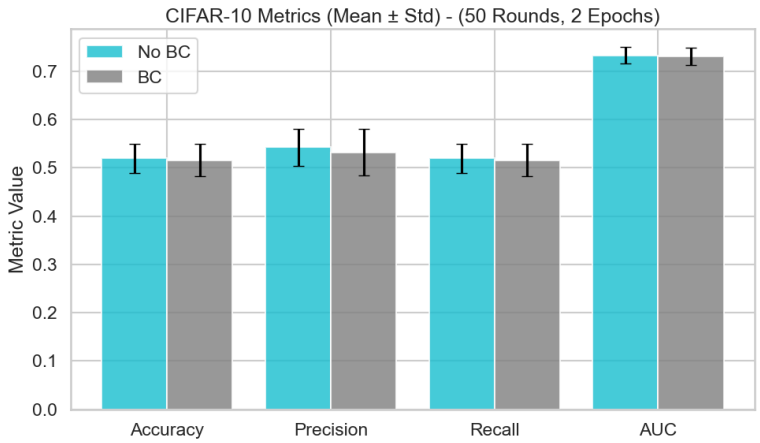


Figure 3.19: CIFAR-10: Mean Accuracy, Precision, Recall, and AUC (with Std. Dev.) for No BC vs. BC.

Overall, these results indicate that while blockchain integration consistently increases training time (by about 4% to 24% for MNIST/FashionMNIST and around 10% for CIFAR-10), the impact on accuracy and other classification metrics is minimal. This additional computational cost may be acceptable in scenarios where the benefits of enhanced trust, auditability, or security in blockchain-based federated learning outweigh the modest time overhead.

3.8.4 Security Point-of-view

The blockchain-based federated learning system incorporates several security mechanisms to ensure data integrity, traceability, and robustness against tampering. Key security features include:

- **Immutability & Auditability:** Every client update is hashed (via `Web3.keccak`) and stored on-chain along with its corresponding IPFS CID and ZKP. This guarantees an immutable audit trail where historical updates cannot be altered without detection.

- **Data Integrity via IPFS:** The decentralized storage of model weights via IPFS ensures that any alteration in the data results in a different CID. Storing the hash of the CID on-chain adds an extra layer of integrity verification.

- **Zero-Knowledge Proof (ZKP) Simulation:** Although the ZKP here is simulated using simple hash functions, it provides a baseline method for verifying that model updates have not been tampered with. In practical settings, more sophisticated ZKP methods (such as SNARKs or STARKs) can be implemented.

- **Rollback Mechanism:** The smart contract includes a rollback function, which can revert the global model to a previous state if a high percentage of updates are identified as outliers. This mechanism adds a safety layer against widespread malicious updates, although it was not triggered in our experiments.
- **Robust Aggregation:** By comparing updates based on the Euclidean distance from the mean and excluding outliers, the system remains resilient against potential poisoning attacks. This robust aggregation is critical in maintaining the overall quality of the global model.

Chapter 4

Conclusions

This chapter concludes the thesis by summarizing the key contributions, findings, and insights from the previous chapters. We also highlight the limitations of our work and suggested future directions for advancing blockchain-enabled federated learning (BCFL) systems.

4.1 Summary of Contributions

In this thesis, we have shown how blockchain technology can be integrated into federated learning (FL) to address challenges related to security, trust, and resisting malicious behaviors. The main contributions are summarized below:

- **Comprehensive Background (Chapter 2).** We presented an overview of FL, its core principles, and common attacks such as data poisoning and inference attacks. We also reviewed existing defense strategies and identified gaps that could be addressed by leveraging blockchain-based solutions.

- **Blockchain-Enabled Federated Learning Implementation (Chapter 3).** We proposed and implemented a BCFL framework that combines:

    1. A *private/consortium blockchain* to maintain a trustworthy and immutable record of training updates.
    2. *InterPlanetary File System (IPFS)* for decentralized storage of model weight updates, ensuring data integrity and verifiability.[25]

3. *Zero-Knowledge Proof (ZKP) simulation* to verify the correctness of submitted updates without revealing sensitive information.
4. A *robust model aggregation* scheme with outlier detection to mitigate data poisoning and malicious updates.

The framework was evaluated on three image-classification tasks (MNIST, FashionMNIST, and CIFAR-10), highlighting its capability to preserve model accuracy while adding a moderate computational overhead.

- **Experimental Evaluation and Security Analysis (Chapter 3).** We conducted some experiments to test accuracy, training time, and other performance metrics in both blockchain and non-blockchain settings. Our results indicate that although blockchain integration increases training time by approximately 4%–24% (depending on the dataset and configuration), it has a negligible or minor impact on final model accuracy. We also showed that the ledger-based audit trail and decentralized storage significantly enhance trust, traceability, and resilience to tampering attempts.

4.2 Key Findings and Insights

1. **Trust and Auditability.** One of the primary motivations for using blockchain in FL is to create a transparent and auditable record of model updates. Our experiments confirmed that once updates are hashed and stored on-chain (with references to IPFS), any malicious attempt to alter past contributions becomes evident, thereby promoting trust among participating clients.

2. **Moderate Overhead vs. Enhanced Security.** The extra time needed to interact with the blockchain, store data on IPFS, and simulate ZKPs introduced a measurable but acceptable overhead in training. This overhead did not exceed roughly 20% for most experiments and averaged around 10% for the larger CIFAR-10 tasks. In practice, this trade-off may be justified in scenarios where the integrity and security of model updates are mission-critical.

3. **Robust Aggregation.** We integrated an outlier detection mechanism that identifies suspicious updates (based on Euclidean distance to the mean) and allows for potential rollbacks if a majority of updates appear malicious. Although our experiments did not trigger this rollback, it provides a valuable safeguard for industrial or adversarial settings.

4. **Consistent Accuracy Across Datasets.** Despite the inherent complexity of blockchain operations, our BCFL framework maintained comparable or slightly lower accuracy compared to standard FL on MNIST, FashionMNIST, and CIFAR-10. In some cases (e.g.,

FashionMNIST), we observed minor improvements in test accuracy under blockchain-based settings, possibly due to more disciplined update verification and data integrity.

4.3 Limitations and Challenges

Although our proposed framework demonstrated the viability of blockchain-based FL, several limitations and open challenges remain:

- **Local Testbed vs. Real-World Network.** Our experiments were conducted on a local Ethereum blockchain (Ganache) and a local IPFS node, which may not fully reflect real-world latency, network partitions, or large-scale node participation. Deploying the BCFL system on a public or consortium blockchain in geographically distributed environments would reveal more practical performance and scalability concerns.[25] [24]

- **Simplified Cryptographic Proofs.** Our current ZKP simulation relies on simple hash-based checks. Integrating advanced zero-knowledge systems (e.g., zk-SNARKs or STARKs) could offer stronger privacy guarantees and more robust proofs of correctness; however, these methods typically impose higher computational overhead.[26] [27]

- **Scalability and Resource Constraints.** We used up to 50 clients in the experiments. In industrial and commercial scenarios, FL may involve thousands or even millions of devices with limited battery and compute resources. Future research must optimize blockchain consensus (e.g., using Proof-of-Stake or Byzantine Fault Tolerant protocols) and data storage strategies to handle massive networks efficiently.

- **Security Threat Models.** Although our framework addresses model poisoning and tampering, there are other sophisticated attacks such as collusion, Sybil attacks, and adaptive adversarial methods. A robust theoretical security analysis or formal verification of the smart contracts and cryptographic mechanisms could further strengthen the system's defense.

- **Need for Regulatory Compliance.** Domains such as healthcare, finance, and critical infrastructure impose strict regulations on data privacy and management. Ensuring compliance (e.g., with GDPR) and implementing formal governance mechanisms for consortium chains will be necessary for real-world adoption.

4.4 Future Research Directions

Building upon this work, the following avenues merit further investigation:

- **Advanced Consensus and Hybrid Chains.** Researching alternative consensus algorithms that are more energy-efficient and faster (e.g., Proof-of-Stake, Delegated Proof-of-Stake, or hybrid approaches) can reduce the overall overhead of BCFL and improve scalability.
- **Adaptive Privacy-Preserving Techniques.** Incorporating advanced privacy schemes such as fully homomorphic encryption or secure multi-party computation, combined with robust on-chain verification, could enhance confidentiality without sacrificing accuracy.
- **Interoperability and Cross-Chain Collaboration.** As different organizations and consortia may adopt distinct blockchain platforms, designing BCFL frameworks that interoperate across heterogeneous chains is an open challenge.
- **Formal Security and Game-Theoretic Incentives.** Future studies could delve deeper into formal security proofs to quantify the resilience against various attack models, or develop incentive mechanisms (e.g., token-based rewards) that motivate honest participation and data quality.
- **Real-World Deployment.** Finally, large-scale pilots in edge computing or IoT applications (e.g., smart transportation, healthcare analytics) would validate the practical benefits of BCFL in securing sensitive data and model updates.

4.5 Final Remarks

The integration of blockchain technology into federated learning frameworks marks a promising step toward secure, trustworthy, and collaborative machine learning. Our work demonstrates that a carefully designed BCFL system can retain high model performance while benefiting from decentralized governance, immutable record-keeping, and robust outlier detection. As the field continues to advance, we anticipate that future research and real-world deployments will further refine the balance between efficiency, security, and scalability, ultimately unlocking the full potential of BCFL for a wide range of critical data-driven applications.

Bibliography

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

[2] E. Guerra, F. Wilhelmi, M. Miozzo, and P. Dini, "The cost of training machine learning models over distributed data sources," *IEEE Open Journal of Communications Society*, vol. 4, 2023.

[3] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, 2020.

[4] C. Zhang, Y. Xie, H. Bai, B. Yu, and W. Li, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106 775, 2021.

[5] Y. Qu, M. P. Uddin, C. Gan, Y. Xiang, L. Gao, and J. Yearwood, *Blockchain-Enabled Federated Learning: A Survey*, *arXiv preprint arXiv:2301.01234*, 2023.

[6] C. Ma, J. Li, L. Shi, *et al.*, "When federated learning meets blockchain: A new distributed learning paradigm," *IEEE Computational Intelligence Magazine*, vol. 17, no. 4, 2022.

[7] H. Chai, S. Leng, Y. Chen, and K. Zhang, "A hierarchical blockchain-enabled federated learning algorithm for knowledge sharing in internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, 2021.

[8] U. Majeed and C. S. Hong, "FLchain: Federated Learning via MEC-enabled Blockchain Network," in *Proceedings of the 20th Asia-Pacific Network Operations and Management Symposium*, 2019.

[9] Y. Qu, S. R. Pokhrel, S. Garg, L. Gao, and Y. Xiang, "A blockchained federated learning framework for cognitive computing in industry 4.0 networks," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, 2021.

[10] D. C. Nguyen, M. Ding, Q.-V. Pham, *et al.*, "Federated learning meets blockchain in edge computing: Opportunities and challenges," *IEEE Internet of Things Journal*, vol. 8, no. 16, 2021.

[11] M. J. Baucas, P. Spachos, and K. N. Plataniotis, "Federated learning and blockchain-enabled fog-iot platform for wearables in predictive healthcare," *IEEE Transactions on Computational Social Systems*, vol. 10, no. 4, 2023.

[12] M. Shaheen, M. S. Farooq, T. Umer, and B.-S. Kim, "Applications of federated learning: Taxonomy, challenges, and research trends," *Electronics*, vol. 11, no. 4, 2022.

[13] N. Malarby, S. Harish Kumar G, R. Sriram, and N. Jebson Immanual Raj, "Federated transfer learning for intrusion detection system in industrial iot 4.0," *Multimedia Tools and Applications*, vol. 83, no. 1–29, 2024.

[14] Z. Zhou, Y. Tian, J. Xiong, J. Ma, and C. Peng, "Blockchain-enabled secure and trusted federated data sharing in iiot," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 5, 2023.

[15] B. Jia, X. Zhang, J. Liu, Y. Zhang, K. Huang, and Y. Liang, "Blockchain-enabled federated learning data protection aggregation scheme with differential privacy and homomorphic encryption in iiot," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 6, 2022.

[16] Z. Zhou, Y. Li, J. Li, R. Xie, and Y. Dai, "Fedcare: Towards interactive diagnosis of federated learning systems," *Frontiers of Computer Science*, vol. 17, no. 2, 2023.

[17] A. Qammar, J. Ding, and H. Ning, "Federated learning attack surface: Taxonomy, cyber defences, challenges, and future directions," *Artificial Intelligence Review*, vol. 55, no. 1, p. 5501, 2022.

[18] S. Salim, B. Turnbull, and N. Moustafa, "A blockchain-enabled explainable federated learning for securing internet-of-things-based social media 3.0 networks," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 4, 2023.

[19] C. Dhasaratha, M. K. Hasan, S. Islam, *et al.*, "Data privacy model using blockchain reinforcement federated learning approach for scalable internet of medical things," *CAAI Transactions on Intelligence Technology*, 2023.

[20] E. Kim and E.-K. Lee, "Evaluating the impact of mobility on differentially private federated learning," *Applied Sciences*, vol. 14, no. 12, p. 5245, 2024.

[21] D. Puthal, N. Malik, S. Mohanty, E. Kougianos, and G. Das, "Everything You Wanted to Know About the Blockchain: Its Promise, Components, Processes, and Problems," *IEEE Consumer Electronics Magazine*, vol. 7, pp. 6–14, Jul. 2018. doi: 10.1109/MCE.2018.2816299.

[22] Y. Y. Sghari, T. Mazhar, T. Shahzad, *et al.*, "The role of blockchain to secure internet of medical things," *Scientific Reports*, vol. 14, p. 18 422, 2024, Published 08 August 2024.

[23] Z. Mahmood and V. Jusas, "Implementation framework for a blockchain-based federated learning model for classification problems," *Symmetry*, vol. 13, no. 7, p. 1116, 2021.

[24] G. Mathur, "GANACHE: A Robust Framework for Efficient and Secure Storage of Data on Private Ethereum Blockchains," *Research Square (Preprint)*, 2023, Posted Date: October 30th, 2023. Licensed under CC BY 4.0. doi: 10.21203/rs.3.rs-3495549/v1.

[25] A. M. Athreya, A. A. Kumar, S. M. Nagarajath, *et al.*, "Peer-to-Peer Distributed Storage Using InterPlanetary File System," in *Advances in Artificial Intelligence and Data Engineering*, ser. Advances in Intelligent Systems and Computing, Springer, Singapore, 2021, pp. 711–721, isbn: 978-981-15-3513-0. doi: 10.1007/978-981-15-3514-7_54.

[26] T. Chen, H. Lu, T. Kunpitaya, and A. Luo, *A Review of zk-SNARKs*, arXiv preprint arXiv:2302.10877, Cryptography and Security (cs.CR), 2023. [Online]. Available: https://arxiv.org/abs/2302.10877.

[27] A. Berentsen, J. Lenzi, and R. Nyffenegger, *A Walk-through of a Simple Zk-STARK Proof*, SSRN preprint No. 4308637, Available at SSRN: https://ssrn.com/abstract=4308637, 2022.