computers

MDPI

Article

Dynamic Cohort Formation with Hierarchical Blockchain Using GDP for Enhanced FL

Sunila Fatima Ahmad [1,†], Zahra Abbas [1,†], Madiha Haider Syed [1,*,†], Adeel Anjum [1,†] and Semeen Rehman [2,*,†]

[1] Institute of Information Technology, Quaid-e-Azam University Islamabad, Islamabad 45320, Pakistan; sfatima@iit.qau.edu.pk (S.F.A.); zahraabbas@iit.qau.edu.pk (Z.A.); aanjum@qau.edu.pk (A.A.)
[2] Institute of Computer Technology, Technical University of Vienna (TU Wien), 1040 Vienna, Austria;
[*] Correspondence: madiha@qau.edu.pk (M.H.S.); semeen.rehman@tuwien.ac.at (S.R.)
[†] These authors contributed equally to this work.

Abstract: Federated Learning (FL) can be defined as an effective solution for using the benefits of machine learning (ML) in distributed systems, in which the data of the clients remain protected. However overlaid challenges affecting today's FL systems are centered on time optimization, scalability, and security. To these challenges, this paper responds with a new solution comprising the dynamic cohort creation via smart contracts and a hierarchical blockchain approach. Firstly, the research utilizes smart contracts for the dynamic tracking of cohorts in real time and addresses a two-tier blockchain structure for safety and efficiency in storage. In addition, Gaussian Differential Privacy (GDP) is used as a privacy-preserving mechanism that adds controlled noise to the data or model updates to protect individual data points from being inferred by adversaries. The method we are proposing includes four practical steps that include efficient update validation and aggregation; this will enhance training time, and increase model accuracy as well as data confidentiality. The standard dataset is used to show enhanced performance and scalability which validates this method. Based on the above investigations, it could be concluded that the proposed approach improves FL efficiency and creates a new direction in the development of secure, accurate, and scalable ML. The present study indicates that the implementation of blockchain with FL fortified by GDP will establish a novel innovation between intelligent and safe Artificial Intelligence (AI) architecture for safeguarding the privacy of ML system.

Keywords: federated learning; blockchain; smart contracts; dynamic cohort management; data privacy; Gaussian Differential Privacy

check for updates

Citation: Ahmad, S.F.; Abbas, Z.; Syed, M.H.; Anjum, A.; Rehman, S. Dynamic Cohort Formation with Hierarchical Blockchain Using GDP for Enhanced FL. *Computers* **2024**, *13*, 317. https://doi.org/10.3390/computers13120317

Academic Editors: Raman Singh and Shantanu Pal

Received: 8 October 2024
Revised: 19 November 2024
Accepted: 23 November 2024
Published: 28 November 2024

1. Introduction

Federated Learning (FL) has emerged as a transformative paradigm in machine learning (ML), enabling decentralized clients to collaboratively train a global model without exposing their local data. This approach is particularly significant in domains where data privacy is paramount, such as healthcare [1], finance [2], and government services. In these fields, stringent regulations like the General Data Protection Regulation (GDPR) [3,4] govern the use and sharing of sensitive data. However, despite its promising applications, FL encounters several critical challenges that hinder its scalability, efficiency, and robustness. One of the primary hurdles is the difficulty of aggregating updates from a large number of clients, leading to communication delays and network congestion [5–7]. Additionally, FL systems face vulnerabilities in the protection of model updates, leaving them susceptible to malicious actors attempting to compromise the system through attacks such as model poisoning [8,9]. Ensuring the integrity and efficiency of the learning process while maintaining robust privacy protections is essential for FL's broader adoption.

In FL, the decentralized nature of data introduces further complexities. Data among clients is often non-independently and identically distributed (non-IID), with significant variations in distribution, quality, and quantity. Traditional FL algorithms assume independently and identically distributed (IID) data, making them less effective in real-world applications where such assumptions rarely hold. Non-IID data can lead to imbalanced learning, biases, and slower convergence of the global model [10]. Addressing these challenges is critical to unlocking the full potential of FL.

To mitigate these limitations, Cohort-Based Federated Learning (CBFL) has been proposed as an enhancement to traditional FL. CBFL organizes clients into smaller, more manageable groups, or cohorts, based on shared characteristics such as data distribution, computational capacity, and network conditions [11,12]. Within each cohort, clients collaboratively train local models, which are then aggregated to form a global model. This hierarchical approach offers significant advantages in terms of scalability, efficiency, and resilience to non-IID data.

The architecture of CBFL provides several key components on both the client and server sides. Let's explore the components of basic CBFL architecture in detail. Figure 1 illustrates the basic architecture of CBFL, highlighting its modular components and hierarchical structure.

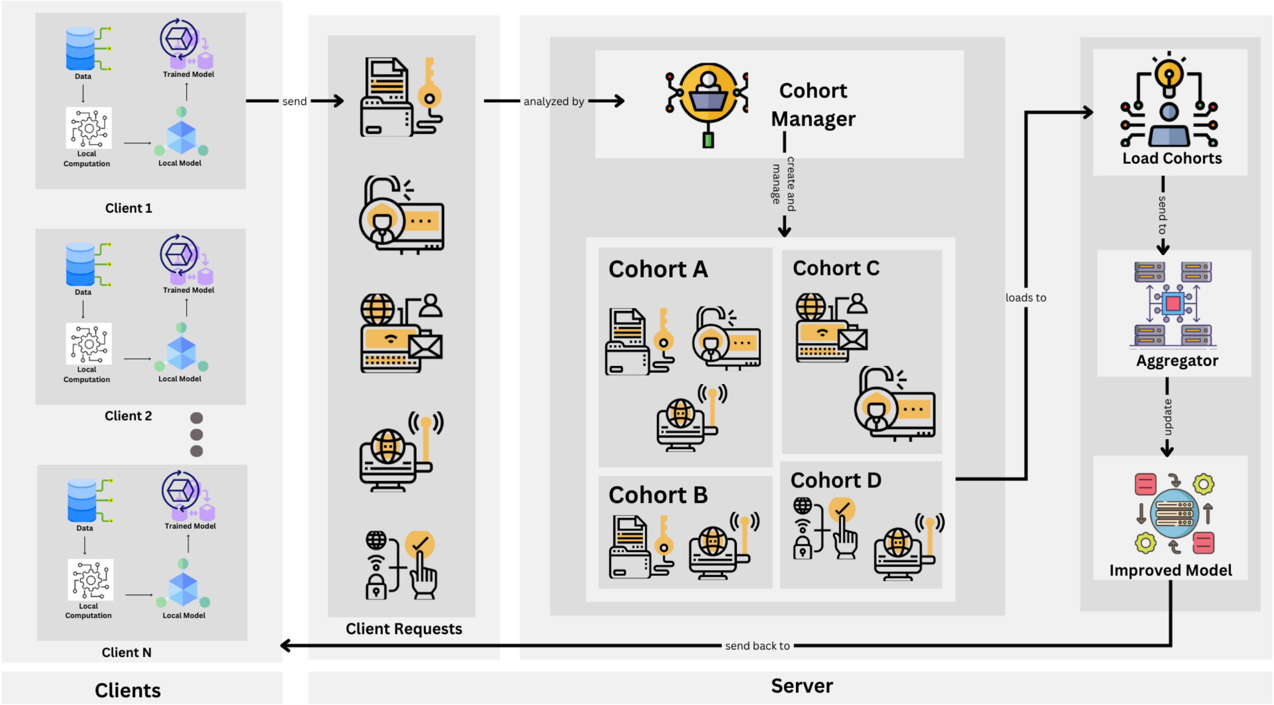

Figure 1. Architectural Diagram of CBFL.

1.1. Client-Side Components

1. Data: Each client in the CBFL architecture retains its dataset locally. This data undergoes preprocessing to ensure it is suitable for effective model training. The local storage and processing of data are crucial for maintaining privacy, ensuring sensitive information never leaves the client's environment.
2. Local Computation and Local Model: ML algorithms are applied to the data of individuals on their local models. That process involves feature selection, training, and validation in the client infrastructure. It also ensures the privacy of the data by sharing only the resulting model parameter.
3. Locally Trained Model: The local computation on the local model produces a locally trained model which serves as an update to the patterns and insights from the client model based on its data. The data is then shared with the server and continues to run without revealing the down values.

4. Client Requests: The server will securely receive the client's trained models, together with the information required for the process of training and managing the cohorts. It is essential for data privacy maintenance during transmission.

1.2. Server-Side Components

1. Cohort Manager: These cohorts are subgroups of models that the Cohort Manager sorts based on factors such as data distribution and computational capacity. This organization assists in mitigating difficulties inherent in non-IID data in FL settings.
2. Cohorts: Cohorts are client model groups that are in similar categories. Condensing models into cohorts is therefore a major strength of CBFL training as it means that the training is specialized and focused.

1.3. Aggregation and Model Refinement

1. Load Cohorts: During this phase, the system gathers cohorts for aggregation. Cohort accumulation and categorization are important tasks before the synthesis of a global model.
2. Aggregator: The Aggregator is the most important tool within the CBFL process. It uses the updates of each of the cohorts to build an even better global model of learning, including the formation of the aggregation process to capture the improvement of the collective learning while at the same time protecting the privacy of the client's data.
3. Improved Model: The final output of this aggregation process is a better global model containing all that has been learned by all the cohorts. FL systems continue to encounter significant challenges in enhancing model accuracy and generalization, particularly due to issues such as data heterogeneity, communication inefficiencies, and privacy concerns [13–16].

While CBFL introduces notable improvements, it is not without its challenges. FL systems, including CBFL, still face communication overheads, scalability, and security issues. Aggregating updates efficiently, especially with many clients, remains a significant hurdle. Moreover, safeguarding against threats such as data and model poisoning, Sybil attacks, and insider threats requires robust mechanisms to ensure trustworthiness and integrity.

In response to these challenges, this research proposes a novel framework that integrates dynamic cohort formation, hierarchical blockchain architecture, and Gaussian Differential Privacy (GDP) mechanisms. Dynamic cohort formation involves grouping clients based on factors such as data quality, computational power, and network stability [16]. This ensures that cohorts are optimized for efficient and effective training, reducing bottlenecks and enhancing data quality. The hierarchical blockchain framework employs a multi-tier architecture, with cohort-level blockchains feeding into a global blockchain. This design provides secure and immutable validation of model updates while enhancing scalability. GDP mechanisms add a layer of privacy by introducing Gaussian noise to the data or model deltas, preventing inference attacks and preserving confidentiality.

Additionally, the proposed approach leverages blockchain technology to enhance transparency and security. Blockchain, an open, distributed, and immutable ledger, has gained prominence in FL systems for recording model updates and managing client contributions securely [17]. By combining blockchain with CBFL, this research addresses the challenges of update aggregation and scalability. The framework also strengthens resilience against potential malicious updates, ensuring robust performance even in adversarial environments.

The contributions of this study are described in the following.

- Dynamic Cohort Formation: This method involves dynamically grouping clients based on data quality, computational power, and network stability [16]. By clustering clients with similar data characteristics, the training process can benefit from improved data quality, as cohorts can share insights and learn from each other's data distributions. Additionally, optimizing computational power ensures that clients with similar processing capabilities are grouped together, reducing the risk of bottlenecks during training. Finally, considering network stability allows for more reliable communication within cohorts, minimizing disruptions and enhancing the overall training efficiency. These factors collectively contribute to a more effective and efficient FL process.

- Hierarchical Blockchain Framework: Proposes a hierarchical blockchain architecture to securely and efficiently manage and validate model updates, ensuring transparency and immutability.
- GDP Integration: Integrates GDP mechanisms to ensure that client data remains confidential and protected against inference attacks during the learning process.
- Enhanced Security and Scalability: Demonstrates how the proposed framework improves the security, scalability, and overall performance of FL systems, addressing key challenges such as data poisoning, model poisoning, and Sybil attacks.
- Comprehensive Security Analysis: Provides a thorough security proof analysis, demonstrating the robustness of the proposed framework against various potential attacks in CBFL.

This paper is structured as follows: Section 1 contains the introduction. Section 2 summarizes existing work in the field, highlighting the gaps that this research aims to fill. Section 3 details the basic preliminaries in CBFL architecture and discusses the vulnerabilities in CBFL architecture to specific security attacks. Section 4 outlines the proposed methodology introduced in this study. Section 5 provides a detailed analysis of the framework's security. Section 6 presents and analyzes the findings, and finally, Section 7 summarizes the study and suggests directions for future work.

2. Literature Review

This literature review explores the evolving landscape of CBFL, GDP, and Blockchain. By critically analyzing existing research, we highlight key trends and challenges that drive advancements in CBFL, particularly in enhancing data privacy and security. This examination aims to illuminate the comparative overview of state-of-the-art CBFL methodologies. The Security and Privacy Analysis of State-of-the-Art is given in Table 1.

Several approaches to CBFL have emerged, each enhancing privacy, security, and performance in FL systems, particularly in resource-constrained environments. One notable method is the FL-as-a-Service (FLaaS) approach described in [11], which integrates Type-based cohort formation with Kernel Principal Component Analysis (KPCA) to counter label-flipping attacks. This method employs Multi-path Service Routing (MSR) to optimize service requests across edge nodes, resulting in improved memory usage, accuracy, and efficiency compared to centralized FLaaS solutions.

In the context of industrial applications, the approach outlined in [12] utilizes K-means clustering and SeqFL to group clients with similar data distributions. This enables collaborative model training without the need for data sharing, effectively addressing data heterogeneity and privacy concerns prevalent in industrial settings. Another significant contribution is the Differentially Private Federated Continual Learning (DP-FCL) system presented in [18]. This system introduces privacy-preserving algorithms designed to mitigate "catastrophic forgetting", allowing FL models to adapt continuously while maintaining user privacy. DP-FCL has been shown to outperform traditional methods, particularly in scenarios with variable privacy budgets.

The paper [2] presents Federated Analytics, which utilizes Secure Multiparty Computation (SMPC) for privacy-preserving cohort analysis in smart cities. This method keeps individual location data private while allowing aggregated data sharing, making it scalable for applications requiring strict privacy. In [19], Lightweight Industrial Cohorted Federated Learning (LICFL) addresses industrial FL challenges by grouping clients with similar model parameters, enhancing performance through adaptive aggregation methods. Real-time datasets validate its efficiency, offering a practical solution for diverse industrial data.

Table 1. Comparison of State-of-the-Art Work in CBFL. (The ✓and × in the table state that all the studies cited in the table only focus on either security or the privacy of the system, But in our work PA-CBFL we focused on both the security and privacy).

Ref	Method	Attacks	Security	Privacy
[11]	CB-KPCA, MSR	Label Flipping	✓	×
[12]	K-means Clustering and SeqFL	Data Privacy	×	✓
[18]	DP-FCL	Data Leakage and Re-identification	×	✓
[2]	SMPC	Re-identification	×	✓
[19]	LICFL	Not Mentioned	×	✓
[20]	Edge-Cloud Architecture	Not Mentioned	×	✓
[21]	MR-FFL	Model Poisoning	✓	✓
[22]	Explainable FL	Not Mentioned	×	×
[23]	SMOTE	Not Mentioned	×	✓
[24]	Grid Search and Bayesian Optimization	Not Mentioned	×	×
[25]	Unsupervised Model Fingerprint and Autoencoder NN	Network-Based	✓	×
[26]	COCE and K-means Clustering	Data and Model Poisoning	✓	×
[27]	FedClust and IFL	Data Poisoning	×	✓
Our Work	PA-CBFL	Data Poisoning Attack, Model Poisoning Attack, Inference Attack, Eavesdropping, Colluding Cohort, insider Attack, Byzantine Failure, SPoF, Membership Inference Attack	✓	✓

The CommunityAI framework in [20] organizes FL participants based on data similarities, particularly in healthcare and education. This approach promotes scalability and privacy, addressing data heterogeneity in community-driven FL applications. MR-FFL, introduced in [21], focuses on heterogeneous UAV networks, emphasizing fairness and privacy through differential privacy mechanisms. It adapts FL for decentralized, resource-limited environments, showcasing its potential in mobile applications.

The industrial lifecycle dashboard proposed in [22] enhances transparency and decision-making in FL by monitoring model performance from registration to deployment, marking a shift towards operational oversight in industrial settings. In [23], the authors assess FL's effectiveness in predictive maintenance and quality inspection, comparing aggregation algorithms across diverse datasets. The findings provide insights into FL's industrial applicability and limitations. The study in [24] explores hyperparameter tuning in FL, showing that global optimization generally outperforms local tuning in non-IID data environments, crucial for efficient model performance. The solution in [25] for Network Anomaly Detection (NAD) in IoT environments employs an unsupervised model fingerprint and Autoencoder Neural Network to efficiently detect anomalies, addressing security needs in distributed IoT networks.

In the realm of security within the Metaverse, the Cohort-based Credit Evaluation (CoCE) model discussed in [26] categorizes clients based on data distribution to assess credit scores and detect potential attackers. This approach effectively addresses issues related to data distribution and sophisticated poisoning attacks, enhancing security in high-risk applications. The framework presented in [27] focuses on the lifecycle management of FL artifacts in industrial environments. By integrating federated clustering for real-time monitoring, this framework enhances the applicability of FL in privacy-sensitive industrial applications, ensuring high performance throughout the entire FL workflow.

In [5], an FL framework built on Hyperledger Fabric showcases a decentralized approach to securing model updates among multiple untrusted participants. By implementing threshold homomorphic encryption (HE) and a verification mechanism to counteract malicious behaviors, this study mitigates privacy and integrity risks while incorporating incentives to boost user engagement, marking a crucial advancement in secure multi-party learning. For healthcare, ref. [28] introduces a blockchain-enhanced FL model aimed at managing vast clinical data from IoT devices in a secure, decentralized way. This scheme reduces congestion and energy demands, aligning with privacy standards and supporting the transition to Healthcare 4.0 by enabling efficient, privacy-preserving data processing.

Another comprehensive study [8] reviews the integration of blockchain with FL, providing an in-depth analysis of security measures, privacy-preserving methods like DP, and implementation challenges across diverse fields. The paper establishes a foundation for future BCFL research, emphasizing the need for robust smart contract security and decentralized control. The framework in [17], called PrivateFL, further strengthens FL security by integrating it with Hyperledger Fabric and using secure aggregation algorithms like VPSA. This system addresses the challenges of inference attacks and malicious participants, establishing a standard for privacy-preserving FL through decentralized trust and secure prediction mechanisms. In credit scoring, ref. [29] examines the convergence of blockchain, explainable AI (XAI), and FL to improve the transparency and trustworthiness of automated credit assessments. The study proposes a novel framework that enables decentralized model learning, data integrity, and model explainability, aiming to replace outdated credit scoring methods reliant on limited historical data. Ref. [30] addresses the persistent challenge of non-IID data in FL by integrating hierarchical blockchain architecture. Smart contracts automate the model distribution and update process, improving accuracy and privacy in FL training with heterogeneous data. This approach demonstrates the potential of blockchain to support robust, privacy-preserving collaborative learning in decentralized environments.

Introducing BLADE-FL [31], this study overcomes centralization issues in traditional FL by implementing a blockchain-assisted decentralized architecture. With client-driven mining and learning, BLADE-FL strengthens security and resilience, incorporating local DP and incentive mechanisms to improve reliability against failures and attacks. FGFL [32] leverages blockchain for a fair incentive mechanism in FL, combining blockchain transparency with trust models like Subjective Logic Model (SLM). This solution addresses FL challenges of low-quality updates and malicious contributions, integrating reputation-based server selection and digital signatures for secure model updates.

The article [33] targets data security in the Industrial IoT (IIoT) using a blockchain-enhanced FL scheme that incorporates privacy-preserving methods such as DP and HE. The proposed aggregation framework secures IIoT data sharing and model training, achieving high accuracy and robust data protection for industrial applications. Focused on privacy and collusion in crowdsourcing, CoPiFL [34] integrates blockchain with HE to secure FL in decentralized platforms. It provides transparency through a tamper-proof ledger, smart contracts for task and reward distribution, and privacy-preserving computations, setting a standard for collaborative, secure FL in crowdsourcing environments.

ESB-FL [35] merges FL with blockchain in healthcare to improve data privacy and scalability. Using a designated decryptor function encryption and a fair payment mechanism, this approach secures medical model training, balancing privacy and accuracy while incentivizing participant engagement and preserving patient data confidentiality. LAFED [36] introduces a lightweight authentication method for blockchain-enabled FL systems. Employing zero-knowledge proofs and a user ID system, this framework reduces resource demands and optimizes participant authentication, addressing bottlenecks in FL scalability and security, thereby paving the way for efficient decentralized learning across domains.

The proposed solution builds on these state-of-the-art contributions by introducing a comprehensive framework that addresses a wider range of attacks, including data and model poisoning, inference attacks, Byzantine failures, insider threats, colluding cohorts, and single points of failure in the CBFL system. By integrating privacy-aware cohort formation with GDP and blockchain-enhanced integrity, our work offers a robust, multi-faceted response to the security and privacy challenges identified in prior works. This solution not only refines cohort formation techniques but also innovates attack resistance, as a cutting-edge advancement in CBFL for diverse and security-sensitive environments.

3. Preliminaries

This section presents the key notations and descriptions essential for understanding the concepts and algorithms related to our methodology.

3.1. Notations Used

It is important to establish the foundational notations and terminologies referenced throughout this article. These symbols form the backbone of the models, algorithms, and analytical frameworks employed, ensuring a coherent understanding of the technical aspects. The notations used in this paper are given in Table 2.

Table 2. Notations and Descriptions.

Notation	Description
$\theta_i^{(t)}$	Model parameters of client i at iteration t
$\Delta_i^{(t)}$	Local model update of client i at iteration t
$\mathcal{C}_j$	Cohort j formed based on data quality, computational power, and network stability
D_i	Data quality of client i
P_i	Computational power of client i
N_i	Network stability of client i
$\Delta_{\mathcal{C}_j}^{(t)}$	Aggregated update of cohort j at iteration t
$B_j^{(t)}$	Blockchain State of cohort j at iteration t
$B_g^{(t)}$	Global blockchain State at iteration t
$\Delta_g^{(t)}$	Global model update at iteration t
$\theta_g^{(t+1)}$	Updated global model parameters at iteration $t+1$
ϵ	Privacy budget
δ	Predefined delta for Gaussian noise magnitude
T	Number of communication rounds
K	Number of cohorts
$\mathcal{N}(0, \delta)$	Gaussian noise with mean 0 and variance δ
$B_g^{(0)}$	Initial global blockchain State
$B_j^{(0)}$	Initial blockchain State of cohort j
$f(P_i, D_i, N_i)$	Function to assign client i to cohort j

3.2. Attack Analysis on CBFL

The presented CBFL is a solid groundwork for distributed ML. Nonetheless, its structure makes it vulnerable to numerous security risks. This section describes the viable attacks on various CBFL components; a clear indication of the significance of security in its design. The attacker model is described in Figure 2.

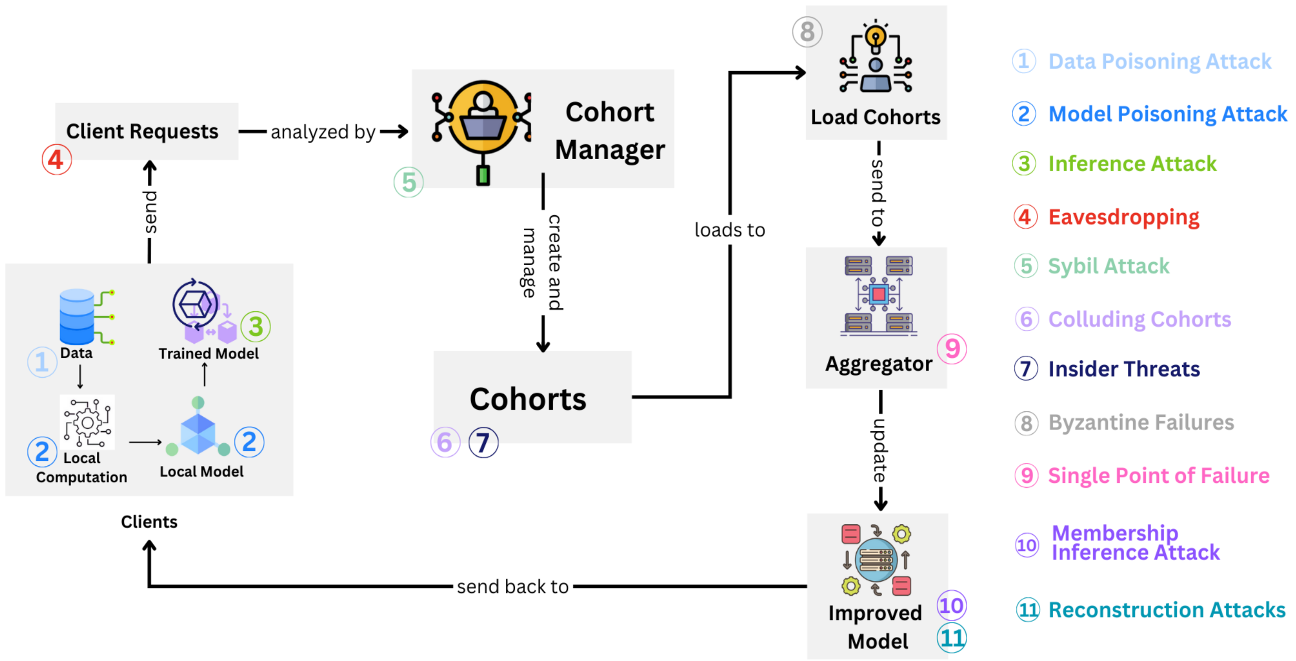

Figure 2. Attacker Model for CBFL.

3.2.1. Data

Data Poisoning Attack: Opponents provide artificial data that acts maliciously upon the dataset of the client and corrupts the global model. The decentralized approach in data collection in CBFL intensifies this risk because poisoned data from a particular client may massively affect the system.

3.2.2. Local Model

Model Poisoning Attack: Cybercriminals modify the parameters of the local model before it is transmitted to the server. Such a massive and noisy attack can lead to potential vulnerabilities or biases being integrated into the global model accumulated at each iteration.

3.2.3. Trained Model

Inference Attack: It has been suggested that opponents study updates to a model to deduce information about the training data—all of which threaten privacy.

3.2.4. Client Requests

Eavesdropping: Interception of messages by unauthorized users is also a major issue because details in communication between the client and the server will be made available.

3.2.5. Cohort Manager

Sybil Attack: Opponents develop bogus profiles to control the formation of cohorts of the CBFL system since they are in the majority. This can lead to biased or compromised global models, which is the major issue to be discussed in this paper.

3.2.6. Cohorts

- Colluding Cohorts: Several groups coordinate to skew the worldwide model or capitalize on its susceptibilities. It can, however, prove difficult to prevent or counter this planned synergy.
- Insider Threats: Malicious insiders within cohorts can change the model updates and or release sensitive information.

3.2.7. Load Cohorts

Byzantine Failures: Some of the malicious cohorts contaminate the data during aggregation, which might undermine the global model. This means that the aggregation algorithm needs to be able to withstand such faults if model integrity is to be maintained.

3.2.8. Aggregator

Single Point of Failure: Cohort updates are incorporated into the global model by use of the Aggregator, which is a very important component. If so, it can damage the entire CBFL process very much. As it is in human anticipations, redundancy is indispensable in system management, and therefore, failover mechanisms are inevitable.

3.2.9. Improved Model

- Membership Inference Attack: They try to decipher whether certain points of information were trained, which is always dangerous from the privacy point of view. It is only possible to overcome such inferences with the aid of specific privacy-preserving approaches like differential privacy.
- Reconstruction Attack: The model updates inspire the creation of new train sets, making adversary's privacy threats huge. Proposes advanced levels of security protocols to thwart such attacks, less invasive methods of data anonymization are essential to such techniques.

Since CBFL is substantially distributed in its method and implementation, it can be vulnerable to assault on different parts of its structure. To ensure that the system is uncompromising, integration of privacy-preserving algorithms and strong security measures have to be put in place.

4. Proposed Methodology

In this section, we propose our approach to improving FL by implementing GDP in dynamic cohort creation and a hierarchical blockchain system. Our approach aims to enhance the efficiency, scalability, and security of FL, with a focus on the integration of cohort-based training, a DP approach, and the principles of blockchain technology. The outline of the proposed approach is given in Algorithm 1 and the architecture diagram is given in Figure 3.

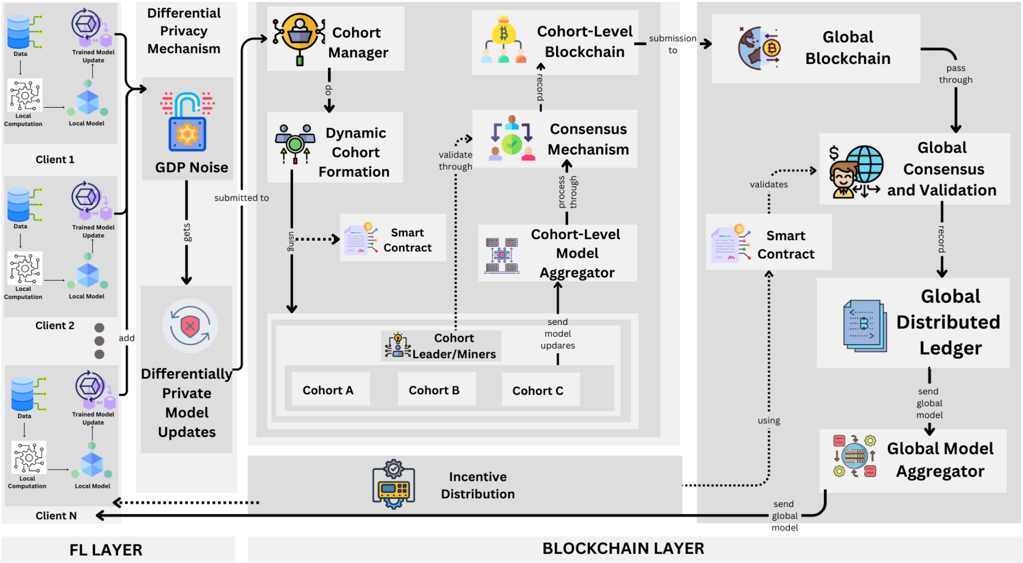

Figure 3. Architectural Diagram of Dynamic Cohort Formation with Hierarchical Blockchain for Enhanced FL.

Algorithm 1 Dynamic Cohort Formation using GDP and Hierarchical Blockchain for Enhanced FL.

```
Input: Client properties $P_i, D_i, N_i$; initial global model parameters $\theta_g^{(0)}$; privacy budget $\epsilon$; number of cohorts $K$; number of communication rounds $T$
Output: Updated global model parameters $\theta_g^{(T)}$
Algorithm:
for round $t = 0$ to $T - 1$ do
    for each client $i$ do
        Train local model using local data to obtain $\theta_i^{(t+1)}$
        Compute local model update $\Delta_i^{(t)} = \theta_i^{(t+1)} - \theta_i^{(t)}$
    end for
    for each client $i$ do
        Compute noise $\mathcal{N}(0, \delta)$ where $\delta$ is a predefined constant
        Add Gaussian noise to local update $\Delta_i^{(t)} = \Delta_i^{(t)} + \mathcal{N}(0, \delta)$
    end for
    for each client $i$ do
        Calculate client properties $P_i, D_i, N_i$
        Assign client $i$ to a cohort $\mathcal{C}_j$ using $\mathcal{C}_j = f(P_i, D_i, N_i)$
        Update cohort memberships dynamically using smart contracts
    end for
    for each cohort $\mathcal{C}_j$ do
        Cohort leader aggregates noisy updates $\Delta_{\mathcal{C}_j}^{(t)} = \frac{1}{|\mathcal{C}_j|} \sum_{c_i \in \mathcal{C}_j} \Delta_i^{(t)}$
        Update cohort-level blockchain $B_j^{(t+1)} = B_j^{(t)} \cup \{\Delta_{\mathcal{C}_j}^{(t)}\}$
    end for
    Aggregate updates from all cohort blockchains $\Delta_g^{(t)} = \frac{1}{K} \sum_{j=1}^{K} \Delta_{\mathcal{C}_j}^{(t)}$
    Update global blockchain $B_g^{(t+1)} = B_g^{(t)} \cup \{\Delta_g^{(t)}\}$
    Update global model parameters $\theta_g^{(t+1)} = \theta_g^{(t)} + \Delta_g^{(t)}$
    Distribute updated global model parameters $\theta_g^{(t+1)}$ to all clients
    Reward clients based on their contributions recorded in the blockchain
    Manage incentives using smart contracts
end for
Return: Final global model parameters $\theta_g^{(T)}$
```

4.1. Client Local Model Training

Each client i independently trains a local model using its private dataset, resulting in model parameters $\theta_i^{(t)}$. Each client calculates the local model update $\Delta_i^{(t)}$ as follows:

$$\Delta_i^{(t)} = \theta_i^{(t+1)} - \theta_i^{(t)} \tag{1}$$

4.2. Application of GDP

Considering data privacy, GDP is adopted as part of the model update. This entails aggregating Gaussian noise to the local updates before passing them through the Cohort Manager. The noise is calculated based on a predefined privacy budget ϵ, which controls the trade-off between privacy and model accuracy.

The noise $\mathcal{N}(0, \delta)$ added to each update is drawn from a Gaussian distribution:

$$\Delta_i^{(t)} = \Delta_i^{(t)} + \mathcal{N}(0, \delta) \tag{2}$$

In this approach, the noise is directly determined by the privacy budget ϵ and a predefined delta (δ). The delta parameter is used as a fixed measure of noise magnitude relative to the desired privacy guarantees, without the need for further calculations involving sensitivity or standard deviation.

This method allows for straightforward integration of privacy-preserving noise into the model updates, ensuring the protection of individual data points while maintaining a balance with model accuracy.

Submission of Differentially Private Updates

Clients submit their differentially private updates to the Cohort Manager to ensure updates are privacy-preserved before further processing.

The sequence diagram for the proposed methodology is shown in Figure 4.

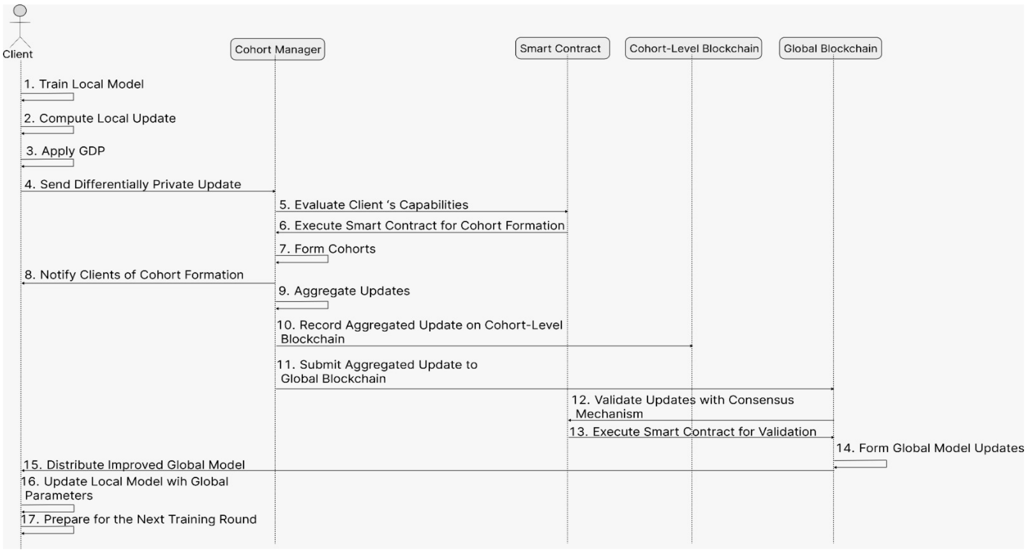
Client
Cohort Manager
Smart Contract
Cohort-Level Blockchain
Global Blockchain
1. Train Local Model
2. Compute Local Update
3. Apply GDP
4. Send Differentially Private Update
5. Evaluate Client 's Capabilities
6. Execute Smart Contract for Cohort Formation
7. Form Cohorts
8. Notify Clients of Cohort Formation
9. Aggregate Updates
10. Record Aggregated Update on Cohort-Level Blockchain
11. Submit Aggregated Update to Global Blockchain
12. Validate Updates with Consensus Mechanism
13. Execute Smart Contract for Validation
14. Form Global Model Updates
15. Distribute Improved Global Model
16. Update Local Model wih Global Parameters
17. Prepare for the Next Training Round

Figure 4. Sequence Diagram of proposed methodology.

4.3. Dynamic Cohort Formation

Dynamic cohort formation is an important aspect of our approach that seeks to enhance the efficiency and security of FL. It entails clustering customers into groups with similar capacities in computation, quality of data, and network stability among others.

Cohort Formation by Cohort Manager

The Cohort Manager dynamically forms cohorts $\mathcal{C}_j$ by evaluating clients based on computational capacity P_i, data quality D_i, and network stability N_i:

$$\mathcal{C}_j = f(P_i, D_i, N_i) \tag{3}$$

Smart contracts are utilized to automate and enforce cohort formation, ensuring that cohorts meet predefined criteria. This makes the process dynamic to adequately accomplish efficient utilization of resources as well as bundling of clients, who have relatively similar capabilities in terms of training, thus reducing communication overhead. Smart contracts guarantee that constant updates are the mirror of changes in the client's performance and availability.

4.4. Hierarchical Blockchain Architecture and Model Aggregation

To address security and scalability issues, we propose a hierarchical blockchain model consisting of two levels: All of these require a cohort-specific blockchain and a global reference blockchain.

4.4.1. Cohort-Level Blockchain Aggregation

Each cohort $\mathcal{C}_j$ maintains a unique blockchain $B_j^{(t)}$ for recording and validating member updates. The cohort leader aggregates the updates from clients within the cohort:

$$\Delta_{\mathcal{C}_j}^{(t)} = \frac{1}{|\mathcal{C}_j|} \sum_{c_i \in \mathcal{C}_j} \Delta_i^{(t)} \tag{4}$$

This aggregated update is recorded in the cohort-level blockchain:

$$B_j^{(t+1)} = B_j^{(t)} \cup \{\Delta_{\mathcal{C}_j}^{(t)}\} \tag{5}$$

Distributing the validation process across multiple cohort-level blockchains reduces bottlenecks associated with a single global blockchain.

4.4.2. Global Blockchain Aggregation

The global blockchain serves as an overarching ledger that integrates updates from all cohort-level blockchains. Aggregated updates from each cohort are submitted to the global blockchain, where they are validated using a global consensus mechanism:

$$\Delta_g^{(t)} = \frac{1}{K} \sum_{j=1}^{K} \Delta_{\mathcal{C}_j}^{(t)} \tag{6}$$

where K is the number of cohorts. The global model update is recorded in the global blockchain:

$$B_g^{(t+1)} = B_g^{(t)} \cup \{\Delta_g^{(t)}\} \tag{7}$$

4.4.3. Distribution of Improved Global Model

The improved global model parameters $\theta_g^{(t+1)}$ are distributed back to all clients for the next round of training:

$$\theta_g^{(t+1)} = \theta_g^{(t)} + \Delta_g^{(t)} \tag{8}$$

4.5. Incentive Mechanism

Clients are rewarded for their contributions based on the updates recorded in the blockchain. Smart contracts manage the distribution of rewards, ensuring fair compensation for client contributions and participation.

This hierarchical approach ensures that the system remains scalable and can efficiently handle a large number of clients without compromising security. This step-by-step process ensures that the FL framework is secure, transparent, and efficient by leveraging blockchain technology. By incorporating dynamic cohort formation with the hierarchical blockchain using GDP structure, the method presented in this paper eliminates the existing issues of the conventional FL system and offers a feasible infrastructure for future advancements in decentralized ML.

5. Security Analysis

In this section, a comprehensive discussion of how the proposed framework tackles several security attacks regularly observed in FL systems is done. Evaluating the security performance, the defined framework minimizes these security threats by implementing dynamic cohort formation, hierarchical blockchain, and GDP which safeguard data integrity, privacy, and the model's robustness.

5.1. Mitigation of Data and Model Poisoning Attacks

In FL systems, poisoning attacks can severely impact model performance. The proposed framework incorporates several countermeasures:

1. Dynamic Cohort Formation: By grouping clients with similar data distributions, the proposed framework enhances the robustness of the model against potential threats. This approach ensures that the influence of a single malicious client is minimized because the aggregated updates from cohorts are based on the collective contributions of multiple clients. In this way, if one client submits a malicious update, its impact is diluted by the contributions of other clients within the same cohort who possess high data integrity. Furthermore, the dynamic nature of cohort formation allows for continuous assessment of client behavior, enabling the system to identify and isolate clients that exhibit anomalous patterns. This mechanism not only protects the integrity of the global model but also fosters a collaborative environment where trustworthy clients can effectively counteract the potential negative effects of malicious actors
2. Hierarchical Blockchain: Each client's updates are immutably recorded on the blockchain, where cohort leaders and members verify the updates prior to aggregation. This process identifies and isolates poisoned or manipulated data early in the training process, preventing its integration into the global model.
3. GDP Integration: Gaussian Differential Privacy adds noise to model updates, diminishing the impact of any poisoned contributions. By maintaining statistical noise within updates, GDP limits the influence of individual malicious inputs, preserving the model's integrity.

5.2. Defenses Against Inference, Sybil, and Eavesdropping Attacks

The framework addresses privacy-related attacks that target client data or try to manipulate system participation.

1. GDP for Privacy Protection: GDP effectively obfuscates sensitive information by adding noise to model updates. This prevents adversaries from extracting specific data points, safeguarding client privacy against inference attacks.
2. Hierarchical Blockchain for Identity Verification: The blockchain tracks each client's contributions and verifies identity during cohort formation, mitigating Sybil attacks by ensuring only legitimate participants contribute to the model. Unauthorized access is also prevented, securing the model from unauthorized data leakage.
3. Secure Communication Channels: In addition to GDP, the blockchain's cryptographic protocols protect data during transmission. Even if an attacker intercepts updates, the obfuscated data ensures privacy, rendering the intercepted information meaningless.

5.3. Resilience Against Byzantine Failures, Insider Threats, and Collusion

The system's decentralized structure enhances robustness against malicious participants or cooperative adversarial behaviors.

1. Consensus Mechanism: Within the hierarchical blockchain, the consensus mechanism ensures that multiple cohort members validate each update. This process identifies and rejects any erroneous or malicious updates, enhancing model resilience against Byzantine failures.
2. Cohort-Based Verification: Dynamic cohort formation allows cross-verification of updates among cohorts, enabling the framework to detect and isolate Byzantine actors. This layer of redundancy ensures model integrity and minimizes disruptions caused by compromised clients.
3. Transparency and Auditability: The blockchain ledger provides a transparent, auditable record of all client actions, enabling quick identification of insider threats. Additionally, cohort-based contributions prevent single entities from exerting excessive influence, reducing the risk of collusion among clients.

5.4. Protection from Membership Inference, Reconstruction Attacks, and Single Point of Failure

To prevent unauthorized access and potential data breaches, the framework incorporates the following protections:

1. Membership and Data Privacy via GDP: GDP's noise mechanism ensures that updates are not significantly influenced by any individual data point, preventing adversaries from inferring the presence of specific data in the training set. This obfuscation also protects against reconstruction attacks, making it nearly impossible for attackers to reverse-engineer the original data.
2. Decentralized Aggregation Process: By utilizing hierarchical blockchain for decentralized aggregation, the framework eliminates single points of failure, distributing responsibility across nodes. Cohorts are formed dynamically, ensuring that no single client or node can compromise the system's stability.
3. Distributed Consensus for Resilience: Dynamic cohort formation and decentralized processing collectively ensure system resilience. The responsibility for maintaining model updates is distributed across nodes, allowing the framework to continue functioning effectively, even if a cohort or node fails.

Decentralization and dynamic cohort formation together ensure that the system remains operational and robust, even in the face of failures.

The proposed framework protects FL from a diverse selection of security attacks and is well-equipped to prevent such incursion. The proposed approach enhances the security, privacy, and reliability of the FL system through privacy-preserving techniques, decentralized processing, and secure communication channels. All the attacks are then countered with measures that are built right into the FL framework, making the FL a comprehensive security solution for collaboration learning platforms.

6. Results and Discussions

This section presents the results of our proposed method, using the CIFAR-10 and MNIST dataset [37,38]. We provide an in-depth analysis of the results obtained from our experiments, which involved 10 FL rounds, 5 local training rounds, up to 15 clients, and up to 3 cohorts. The experiment was conducted on the Kaggle platform utilizing a GPU with 100 computing units, 13 GB of GPU RAM, and a 64-bit operating system.

6.1. Results

The performance of the proposed model, evaluated on CIFAR-10 and MNIST datasets, is presented in Figure 5. Metrics such as accuracy, loss, memory usage, training time, CPU usage, sent bandwidth, and received bandwidth are analyzed across 10 training rounds.

6.1.1. Accuracy and Loss Analysis

The accuracy trends (upper-left graph) reveal a trade-off between privacy budgets (ϵ) and model performance. On CIFAR-10, accuracy peaks at 95.24% with $\epsilon = 0.0$ and five clients but declines to 93.07% at $\epsilon = 1.0$ with 10 clients. MNIST results are even more robust, with accuracy starting at 99.86% ($\epsilon = 0.0$) for five clients and marginally dropping to 99.42% ($\epsilon = 1.0$) for 15 clients.

The loss trends (lower-right graph) demonstrate effective convergence across datasets, with CIFAR-10 reaching a minimum loss of 0.1421 ($\epsilon = 0.0$) and MNIST achieving 0.0043 ($\epsilon = 0.0$), both with five clients. These results validate the model's capability to maintain high accuracy and low loss while adhering to DP constraints.

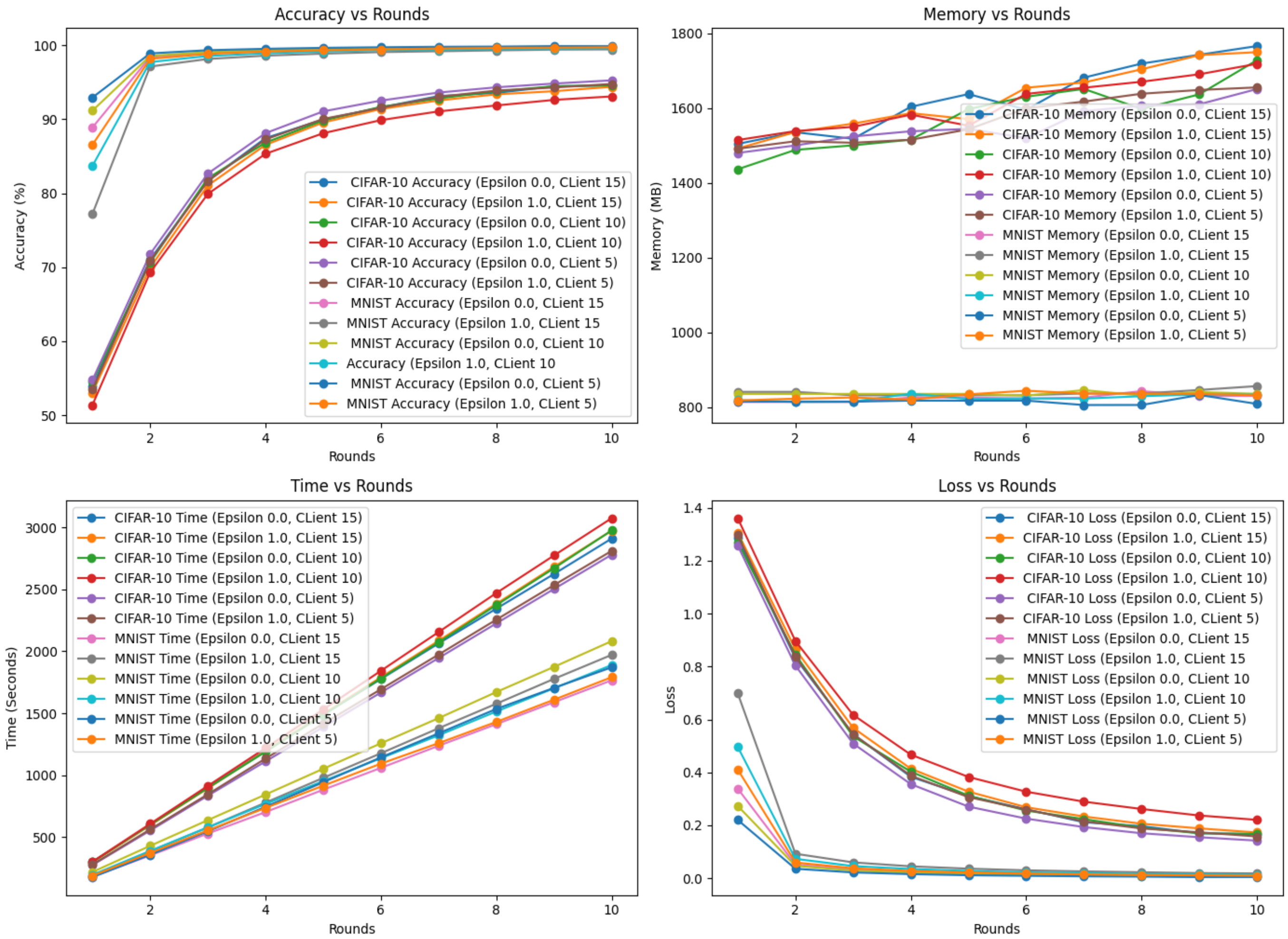

Figure 5. Results of proposed approach.

6.1.2. Memory Usage and Training Time

The upper-right graph in Figure 5 depicts memory usage during training. Memory remains stable across different ϵ values, with CIFAR-10 peaking at 1766.16 MB ($\epsilon = 0.0$, 15 clients) and MNIST peaking at 856.30 MB ($\epsilon = 1.0$, 15 clients). This reflects efficient memory management even with privacy-preserving mechanisms.

The training time (lower-left graph) highlights the computational overhead introduced by privacy-preserving mechanisms. On CIFAR-10, training time ranges from 2780.31 s ($\epsilon = 0.0$, five clients) to 2976.02 s ($\epsilon = 1.0$, 10 clients). MNIST shows similar trends, with training time increasing from 1766.01 s ($\epsilon = 0.0$, 15 clients) to 2082.24 s ($\epsilon = 1.0$, 10 clients). These results underline the balance between robust privacy mechanisms and computational feasibility.

6.1.3. CPU Usage, Sent Bandwidth, and Received Bandwidth

Table 3 provides a detailed analysis of CPU usage, sent bandwidth, and received bandwidth, offering insights into resource utilization across datasets and privacy budgets.

- CPU Usage: CPU utilization remains efficient for CIFAR-10, ranging between 4.0% and 4.5%. For MNIST, usage varies from 4.7% to 9.2%, with higher values observed for smaller cohorts due to increased computational load per client. These trends suggest that resource demands are manageable, even on devices with constrained processing power.

Table 3. Results for our proposed approach on different privacy budgets.

Dataset	Client	Epsilon	Accuracy	Loss	Memory	Time	CPU	Sent Bandwidth	Received Bandwidth
CIFAR-10	15	0.0	94.55%	0.1669	1766.160	2910.55	4.5%	4.5051	166.21
CIFAR-10	15	1.0	94.37%	0.1728	1749.914	2974.163	4.2%	2.0995	164.67
CIFAR-10	10	0.0	94.57%	0.1674	1728.425	2976.018	4.0%	2.5577	166.03
CIFAR-10	10	1.0	93.07%	0.2205	1718.75	3074.994	4.5%	2.2510	164.77
CIFAR-10	5	0.0	95.24%	0.1421	1651.203	2780.307	4.3%	1.8222	164.60
CIFAR-10	5	1.0	94.76%	0.1578	1655.585	2811.001	4.5%	2.1382	164.84
MNIST	15	0.0	99.74%	0.0083	829.7578	1766.010	9.2%	0.7727	12.07
MNIST	15	1.0	99.42%	0.0181	856.3046	1971.252	4.8%	0.9183	12.43
MNIST	10	0.0	99.78%	0.0066	835.7187	2082.244	4.7%	1.5511	11.94
MNIST	10	1.0	99.59%	0.0129	833.6875	1889.353	8.2%	0.7956	12.16
MNIST	5	0.0	99.86%	0.0043	808.6718	1870.821	5.2%	1.6583	11.97
MNIST	5	1.0	99.71%	0.0095	831.6523	1791.158	5.2%	0.6125	11.66

- Sent Bandwidth: Sent bandwidth is consistently low across datasets, varying from 1.8222 MB (five clients, $\epsilon = 0.0$) to 4.5051 MB (15 clients, $\epsilon = 0.0$) for CIFAR-10, and 0.7727 MB to 1.6583 MB for MNIST. This demonstrates that the communication overhead introduced by privacy mechanisms is minimal and scales efficiently with cohort size.
- Received Bandwidth: Received bandwidth shows more variation. On CIFAR-10, it peaks at 166.21 MB ($\epsilon = 0.0$, 15 clients), while MNIST results are significantly lower, with a maximum of 12.43 MB ($\epsilon = 1.0$, 15 clients). These findings indicate that the architecture remains scalable for higher client configurations while received bandwidth increases with client count.

Critical Analysis and Practical Implications: The additional metrics reveal critical insights into the computational and communication overheads introduced by dynamic cohort formation and GDP. While the trade-off between privacy and accuracy is evident, our proposed approach maintains efficient resource utilization and communication scalability. Deployment Feasibility: The low sent bandwidth and stable CPU usage make our proposed approach suitable for deployment in resource-constrained environments, such as IoT devices or edge computing scenarios. However, higher received bandwidth in large cohorts may pose challenges in low-connectivity networks, requiring further optimization. Scalability: The results confirm that our proposed framework can handle larger client configurations without significant performance degradation. The findings validate the balance achieved between privacy, efficiency, and scalability, demonstrating the robustness of our proposed architecture.

6.2. Discussion

The results of our experiments underscore the effectiveness of integrating dynamic cohort formation with a hierarchical blockchain architecture and GDP in FL. This section explores the strengths and weaknesses of our proposed approach, emphasizing how it addresses the challenges of efficiency, scalability, security, and privacy in traditional FL systems. Table 4 represents the comparison analysis of our proposed approach with previous CBFL methods. The ✓and × in Table 4 indicate whether the cited CBFL studies address the respective attacks listed in the table. ✓denotes that a specific attack is considered, while a × signifies that it is not. Unlike the previous studies, our proposed work addresses all the listed attacks comprehensively.

Table 4. Comparison Analysis of our proposed solution with previous CBFL Approaches.

Attacks	Our Work	[11]	[12]	[18]	[2]	[21]	[25]	[26]	[27]
Data Poisoning Attack	✓	✓	×	✓	×	×	×	✓	✓
Model Poisoning Attack	✓	×	×	×	×	✓	×	✓	×
Inference Attack	✓	×	✓	✓	✓	×	×	×	×
Sybil Attack	✓	×	×	×	×	×	×	×	×
Eavesdropping	✓	×	×	×	✓	×	✓	×	×
Byzantine Failures	✓	×	×	×	×	×	×	×	×
Membership Inference Attacks	✓	×	×	✓	×	×	×	×	×
Reconstruction Attacks	✓	×	×	×	✓	×	×	×	×
Insider Threats	✓	×	×	×	×	×	×	×	×
Colluding Cohorts	✓	×	×	×	×	×	×	×	×
Single Point of Failure	✓	×	×	×	×	×	×	×	×

6.2.1. Pros of the Proposed Approach

- Enhanced Efficiency: Dynamic cohort formation groups clients based on computational power, data quality, and network stability, accelerating convergence and reducing training time.
- Scalability: The hierarchical blockchain architecture efficiently manages a large number of clients by distributing computational load through cohort-specific blockchains, enabling seamless scaling.
- Blockchain technology ensures the integrity and security of model updates, providing tamper-proof records and safeguarding against malicious activities.
- Enhanced Privacy with GDP: GDP adds a robust layer of privacy by injecting noise into model updates, effectively protecting client data while balancing privacy and model accuracy.

6.2.2. Cons of the Proposed Approach

- Complex Implementation: Implementing smart contracts and managing hierarchical blockchains adds complexity to system setup, requiring careful design and deployment.
- Resource Requirements: Initial setup and maintenance of blockchain infrastructure require significant computational resources and expertise, which can be challenging for organizations with limited technical capabilities.
- Increased Computational Overhead: GDP introduces additional computational overhead, potentially leading to longer training times, especially with a lower privacy budget (epsilon).

This study paves the way for more robust and scalable FL systems, ensuring secure, efficient, and privacy-preserving collaborative model training across distributed clients. By addressing key limitations of traditional FL, our proposed approach contributes significantly to advancing the field of decentralized ML.

7. Conclusions

In this work, we proposed a novel framework to enhance FL by integrating dynamic cohort formation using smart contracts, hierarchical blockchain architecture, and GDP. Our experimental results demonstrated significant improvements in model accuracy, privacy preservation, resource efficiency, and computational scalability. The integration of GDP provided strong privacy guarantees by introducing noise to client data updates, ensuring that sensitive information remained protected throughout the training process.

Dynamic cohort formation proved instrumental in grouping clients with similar computational capabilities, data quality, and network stability. The hierarchical blockchain architecture addressed critical issues of scalability and security by facilitating trustworthy and verifiable aggregation of updates. It also minimized computational overhead while ensuring the integrity of the model training process. The architecture's ability to streamline communication and maintain robust security made it a vital component of our framework.

Building on these findings, future work can explore how this architecture could be adapted for other ML models and datasets with varying characteristics. Research into optimizing the smart contract algorithms for dynamic cohort formation could further improve efficiency. Additionally, exploring alternative consensus mechanisms to enhance blockchain performance remains an open avenue for improvement. On the privacy front, advancing DP techniques may enhance data protection without compromising model utility. By addressing these directions, the proposed framework can continue to evolve, making FL more robust, secure, and applicable across diverse domains.

Author Contributions: Conceptualization, M.H.S. and A.A.; Methodology, S.F.A., M.H.S., A.A. and S.R.; Software, S.F.A.; Validation, A.A.; Investigation, S.R.; Writing—original draft, S.F.A.; Writing—review & editing, Z.A. and M.H.S.; Visualization, Z.A.; Supervision, M.H.S., A.A. and S.R.; Funding acquisition, S.R. All authors have read and agreed to the published version of the manuscript.

Funding: The APC was funded by TU Wien Open Access Funding Program.

Data Availability Statement: No new datasets were created.

Acknowledgments: The authors acknowledge TU Wien Bibliothek for financial support through its Open Access Funding Program.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
ML	Machine Learning
FL	Federated Learning
IoT	Internet of Things
IID	Personally Identifiable Information
CBFL	Cohort-Based Federated Learning
KPCA	Kernel Principal Component Analysis
DP	Differential Privacy
HE	Homomorphic Encryption
SMP	Secure Multiparty Computation
GDP	Gaussian Differential Privacy
PA-CBFL	Privacy-Aware Cohort-Based Federated Learning

References

1. Joshi, M.; Pal, A.; Sankarasubbu, M. Federated learning for healthcare domain-pipeline, applications and challenges. *ACM Trans. Comput. Healthc.* **2022**, *3*, 1–36. [CrossRef]
2. Gjoreski, M.; Laporte, M.; Langheinrich, M. Toward privacy-aware federated analytics of cohorts for smart mobility. *Front. Comput. Sci.* **2022**, *4*, 891206. [CrossRef]
3. Voigt, P.; Von dem Bussche, A. The eu general data protection regulation (gdpr). In *A Practical Guide*, 1st ed.; Springer International Publishing: Cham, Switzerland, 2017; Volume 10, pp. 10–5555.
4. Abbas, Z.; Ahmad, S.F.; Syed, M.H.; Anjum, A.; Rehman, S. Exploring Deep Federated Learning for the Internet of Things: A GDPR-Compliant Architecture. *IEEE Access* **2024**, *12*, 10548–10574. [CrossRef]
5. Sun, J.; Wu, Y.; Wang, S.; Fu, Y.; Chang, X. Permissioned blockchain frame for secure federated learning. *IEEE Commun. Lett.* **2021**, *26*, 13–17. [CrossRef]
6. Krishnan, S.; Jose, A.A.; Srinivasan, R.; Kavitha, R.; Suresh, S. *Federated Learning*; CRC Press: Boca Raton, FL, USA, 2024.
7. Liu, B.; Lv, N.; Guo, Y.; Li, Y. Recent Advances on Federated Learning: A Systematic Survey. *arXiv* **2023**, arXiv:2301.01299
8. Salim, M.M.; Yang, L.T.; Park, J.H. Privacy-preserving and scalable federated blockchain scheme for healthcare 4.0. *Comput. Netw.* **2024**, *247*, 110472. [CrossRef]

9. Qu, Y.; Uddin, M.P.; Gan, C.; Xiang, Y.; Gao, L.; Yearwood, J. Blockchain-enabled federated learning: A survey. *ACM Comput. Surv.* **2022**, *55*, 1–35. [CrossRef]
10. Chiaro, D.; Prezioso, E.; Ianni, M.; Giampaolo, F. FL-Enhance: A federated learning framework for balancing non-IID data with augmented and shared compressed samples. *Inf. Fusion* **2023**, *98*, 101836. [CrossRef]
11. Sikandar, H.S.; Anjum, A.; Khan, A.; Jeon, G. Cohort-based kernel principal component analysis with Multi-path Service Routing in Federated Learning. *Future Gener. Comput. Syst.* **2023**, *149*, 518–530. [CrossRef]
12. Hiessl, T.; Rezapour Lakani, S.; Kemnitz, J.; Schall, D.; Schulte, S. Cohort-based federated learning services for industrial collaboration on the edge. *J. Parallel Distrib. Comput.* **2022**, *167*, 64–76. [CrossRef]
13. Bu, Z.; Dong, J.; Long, Q.; Su, W.J. Deep learning with gaussian differential privacy. *Harv. Data Sci. Rev.* **2020**, *2020*. [CrossRef]
14. Liu, J.; Lai, F.; Dai, Y.; Akella, A.; Madhyastha, H.V.; Chowdhury, M. Auxo: Efficient federated learning via scalable client clustering. In Proceedings of the 2023 ACM Symposium on Cloud Computing, Santa Cruz, CA, USA, 30 October–1 November 2023; pp. 125–141.
15. Sattler, F.; Wiedemann, S.; Müller, K.R.; Samek, W. Robust and communication-efficient federated learning from non-iid data. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 3400–3413. [CrossRef] [PubMed]
16. Charles, Z.; Garrett, Z.; Huo, Z.; Shmulyian, S.; Smith, V. On large-cohort training for federated learning. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 20461–20475.
17. Antal, M.; Mihailescu, V.; Cioara, T.; Anghel, I. Blockchain-based distributed federated learning in smart grid. *Mathematics* **2022**, *10*, 4499. [CrossRef]
18. Chathoth, A.K.; Necciai, C.P.; Jagannatha, A.; Lee, S. Differentially Private Federated Continual Learning with Heterogeneous Cohort Privacy. In Proceedings of the 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, 17–20 December 2022; pp. 5682–5691.
19. Amarlingam, M.; Wani, A.; NL, A. Lightweight Industrial Cohorted Federated Learning for Heterogeneous Assets. *arXiv* **2024**, arXiv:2407.17999.
20. Murturi, I.; Donta, P.K.; Dustdar, S. Community AI: Towards Community-based Federated Learning. In Proceedings of the 2023 IEEE 5th International Conference on Cognitive Machine Intelligence (CogMI), Atlanta, GA, USA, 1–3 November 2023; pp. 1–9.
21. Zhou, Z.; Zhuang, Y.; Li, H.; Huang, S.; Yang, S.; Guo, P.; Zhong, L.; Yuan, Z.; Xu, C. MR-FFL: A Stratified Community-Based Mutual Reliability Framework for Fairness-Aware Federated Learning in Heterogeneous UAV Networks. *IEEE Internet Things J.* **2024**, *11*, 20995–21009. [CrossRef]
22. Ungersböck, M.; Hiessl, T.; Schall, D.; Michahelles, F. Explainable federated learning: A lifecycle dashboard for industrial settings. *IEEE Pervasive Comput.* **2023**, *22*, 19–28. [CrossRef]
23. Pruckovskaja, V.; Weissenfeld, A.; Heistracher, C.; Graser, A.; Kafka, J.; Leputsch, P.; Schall, D.; Kemnitz, J. Federated learning for predictive maintenance and quality inspection in industrial applications. In Proceedings of the 2023 Prognostics and Health Management Conference (PHM), Paris, France, 31 May–2 June 2023; pp. 312–317.
24. Holly, S.; Hiessl, T.; Lakani, S.R.; Schall, D.; Heitzinger, C.; Kemnitz, J. Evaluation of hyperparameter-optimization approaches in an industrial federated learning system. In Proceedings of the Data Science–Analytics and Applications: Proceedings of the 4th International Data Science Conference–iDSC2021, Hyderabad, India, 26–27 December 2022; pp. 6–13.
25. Sáez-de Cámara, X.; Flores, J.L.; Arellano, C.; Urbieta, A.; Zurutuza, U. Clustered federated learning architecture for network anomaly detection in large scale heterogeneous IoT networks. *Comput. Secur.* **2023**, *131*, 103299. [CrossRef]
26. Li, H.; Zhou, Z.; Zhang, H.; Jiang, K. Cohort-based Federated Learning Credit Evaluation Method in the Metaverse. In Proceedings of the 2023 7th International Conference on High Performance Compilation, Computing and Communications, Jinan, China, 17–19 June 2023; pp. 1–7.
27. Hiessl, T.; Lakani, S.R.; Ungersboeck, M.; Kemnitz, J.; Schall, D.; Schulte, S. Lifecycle Management of Federated Learning Artifacts in Industrial Applications. In Proceedings of the 2023 IEEE 7th International Conference on Fog and Edge Computing (ICFEC), Bangalore, India, 1–4 May 2023; pp. 7–15.
28. Sameera, K.; Nicolazzo, S.; Arazzi, M.; Nocera, A.; KA, R.R.; Vinod, P.; Conti, M. Privacy-preserving in Blockchain-based Federated Learning systems. *Comput. Commun.* **2024**, *222*, 38–67.
29. Jovanovic, Z.; Hou, Z.; Biswas, K.; Muthukkumarasamy, V. Robust integration of blockchain and explainable federated learning for automated credit scoring. *Comput. Netw.* **2024**, *243*, 110303. [CrossRef]
30. Zhang, F.; Zhang, Y.; Ji, S.; Han, Z. Secure and decentralized federated learning framework with non-IID data based on blockchain. *Heliyon* **2024**, *10*, e27176. [CrossRef]
31. Ma, C.; Li, J.; Shi, L.; Ding, M.; Wang, T.; Han, Z.; Poor, H.V. When federated learning meets blockchain: A new distributed learning paradigm. *IEEE Comput. Intell. Mag.* **2022**, *17*, 26–33. [CrossRef]
32. Gao, L.; Li, L.; Chen, Y.; Xu, C.; Xu, M. FGFL: A blockchain-based fair incentive governor for Federated Learning. *J. Parallel Distrib. Comput.* **2022**, *163*, 283–299. [CrossRef]
33. Jia, B.; Zhang, X.; Liu, J.; Zhang, Y.; Huang, K.; Liang, Y. Blockchain-enabled federated learning data protection aggregation scheme with differential privacy and homomorphic encryption in IIoT. *IEEE Trans. Ind. Inform.* **2021**, *18*, 4049–4058. [CrossRef]
34. Xiong, R.; Ren, W.; Zhao, S.; He, J.; Ren, Y.; Choo, K.K.R.; Min, G. CoPiFL: A collusion-resistant and privacy-preserving federated learning crowdsourcing scheme using blockchain and homomorphic encryption. *Future Gener. Comput. Syst.* **2024**, *156*, 95–104. [CrossRef]

35. Chen, B.; Zeng, H.; Xiang, T.; Guo, S.; Zhang, T.; Liu, Y. Esb-fl: Efficient and secure blockchain-based federated learning with fair payment. *IEEE Trans. Big Data* **2022**, *10*, 761–774. [CrossRef]
36. Ji, S.; Zhang, J.; Zhang, Y.; Han, Z.; Ma, C. LAFED: A lightweight authentication mechanism for blockchain-enabled federated learning system. *Future Gener. Comput. Syst.* **2023**, *145*, 56–67. [CrossRef]
37. Alex, K. Learning Multiple Layers of Features from Tiny Images. 2009. Available online: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf (accessed on 11 February 2023).
38. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.