

Blockchain-Enabled Privacy-Preserving Second-Order Federated Edge Learning in Personalized Healthcare

Anum Nawaz , Muhammad Irfan , Xianjia Yu , Zhuo Zou , Tomi Westerlund  (Senior IEEE Member)

Abstract—Federated learning (FL) has attracted increasing attention to mitigate security and privacy challenges in traditional cloud-centric machine learning models specifically in healthcare ecosystems. FL methodologies enable the training of global models through localized policies, allowing independent operations at the edge clients’ level. Conventional first-order FL approaches face several challenges in personalized model training due to heterogeneous non-independent and identically distributed (non-iid) data of each edge client. Recently, second-order FL approaches maintain the stability and consistency of non-iid datasets while improving personalized model training. This study proposes and develops a verifiable and auditable optimized second-order FL framework *BFEL* (*blockchain enhanced federated edge learning*) based on optimized *FedCurv* for personalized healthcare systems. *FedCurv* incorporates information about the importance of each parameter to each client’s task (through Fisher Information Matrix) which helps to preserve client-specific knowledge and reduce model drift during aggregation. Moreover, it minimizes communication rounds required to achieve a target precision convergence for each edge client while effectively managing personalized training on non-iid and heterogeneous data. The incorporation of Ethereum-based model aggregation ensures trust, verifiability, and audibility while public key encryption enhances privacy and security. Experimental results of federated CNNs and MLPs utilizing Mnist, Cifar-10, and PathMnist demonstrate the high efficiency and scalability of the proposed framework.

Index Terms—Federated Learning, *FedCurv*, Data Privacy, Blockchain, Personalized Healthcare, Federated Edge Learning, Second-Order Federated Learning

I. INTRODUCTION

Traditional machine learning (ML) methodologies necessitate training on data consolidated within a single data repository, which may be either centralized or distributed [1]. It requires raw data from multiple participants to be transmitted to a centralized server (where

the data are aggregated). However, aggregating data from multiple stakeholders in healthcare systems poses significant challenges, particularly concerning security, compromising data owners’ privacy, and possibly exposing sensitive health data [2].

Within the distributed ML paradigm, two primary frameworks exist: data centre-based distributed ML and cross-device FL. The former utilizes optimized computing nodes, data shuffling, and high-bandwidth communication networks, whereas the latter operates on a large number of resource-constrained devices with limited computational, storage, and communication capabilities [3].

Federated edge learning (FEL) has emerged as a distributed machine learning paradigm that mitigates privacy concerns by facilitating collaborative model training while ensuring that data remains decentralized. FEL bridges these two paradigms, possessing computational and storage capacities comparable to data centre-based ML while sharing communication constraints with cross-device FL due to physical distance between edge nodes, multi-hop transmissions, and diverse communication mediums. It offers a decentralized alternative, enabling multiple devices to collaboratively train a model without sharing raw data, where each participant trains a model locally, and shares only the model parameters. Some of the widely used gradient-based first-order federated learning approaches are FedAvg, FedSgd, FedAdam, and FedYogi. While these methodologies preserve the confidentiality of the sensitive data, the resulting shared model parameters remain vulnerable to confidentiality breaches during aggregation and dissemination. Despite its several advantages, FL faces two fundamental challenges: (i) managing heterogeneous systems within the federated network and (ii) handling real-world data that are often non-independently and identically distributed (non-iid) among clients [4]. Specifically, the delivery of personalized FL services stands crucial in healthcare advancement because patients manifest individual health profiles with specific requirements. The implementation of first-order FL algorithms encounters substantial hurdles when it comes to personalizing deep learning and machine learning models [5]. Such training datasets have limited representation of particular classes or behaviors because specific data points are spread sparsely throughout the dataset.

However, second-order FL methods provide better suitability for varied heterogeneous healthcare data present in different edge client devices. It allows better customization of local models thus enhancing their value for per-

Anum Nawaz is working with Shanghai Key Laboratory of Intelligent Information Processing Lab, Fudan University, China and Turku Intelligent Embedded and Robotic Systems (TIERS) Lab, Faculty of Technology, University of Turku, Finland (e-mail: 18110720163@fudan.edu.cn), Muhammad Irfan and Xianjia Yu is working with TIERS Lab, University of Turku, Finland.

Zhuo Zou (Senior Member, IEEE) is a Professor with the School of Information Science and Technology, Fudan University, Shanghai, China. (e-mail: zhuo@fudan.edu.cn)

Tomi Westerlund (Senior Member, IEEE) is a Professor with Robotics and Autonomous Systems at the University of Turku. He leads the TIERS Lab (tiers.utu.fi), Faculty of Technology, University of Turku, Finland. (e-mail: toveve@utu.fi)

sonalized healthcare systems. Such adaptations provide potential benefits of privacy and security in heterogeneous environments while preserving personalization (i.e. higher model adaption variability) such as Natural Gradient Descent [6] and Quasi-Newton method [7]. Nevertheless, the distributed implementation of second-order methodologies in a traditional manner remains challenging due to their reliance on inverse matrix-vector product computations, which introduce complexity in determining the descent direction. Addressing these challenges, it is crucial to develop efficient and scalable second-order optimization techniques tailored to heterogeneous FL environments. Second-order optimization methods offer a key advantage by incorporating curvature information of the loss function, thus improving personalized training processes.

Recently, some adaptations incorporate second-order curvature information using the Fisher Information Matrix (FIM) for better convergence [8]. Moreover, integration of distributed ledger technologies in FL approaches significantly improves systems auditability and transparency in data-sensitive health ecosystems to build reliable and unbiased intelligent systems [9].

This study aims to optimize a distributed optimization algorithm FedCurv (Federated Curvature) that minimizes communication rounds required to achieve a target precision for convergence while effectively managing personalized training on non-iid and heterogeneous data across edge client devices. The main contributions of this article:

- i). A novel privacy-preserving blockchain enhanced federated edge learning (*BFEL*) framework, which maintains auditability, verifiability, availability and ensures privacy protection in edge-FL environments.
- ii). We proposed optimized implementation of *FedCurv* for heterogeneous medical data to improve personalized health monitoring and prediction.
- iii). Blockchain-based secure aggregation mechanisms to ensure trust, tamper-proof model updates, decentralized auditability, and accountability.
- iv). Security, scalability, and correctness analysis of our proposed scheme, *BFEL* demonstrate high performance and model utility while maintaining privacy.

The remainder of this article is organized as follows. Section II presents the background and motivation, Section III details the proposed BFEL framework, and Section IV describes the experimental setup. Section V discusses the results, and Section VI concludes the paper with future research directions.

II. RELATED WORK

In recent years, advancements in AI, particularly in the domain of DL within ML, have significantly contributed to the enhancement of smart and personalized healthcare ecosystems [10]. This progress has been largely driven by the expansion of training datasets. However, as the volume of training data increases, so too does the associated risk of privacy breaches [11]. Studies indicate that privacy attacks targeting DL models can result in the

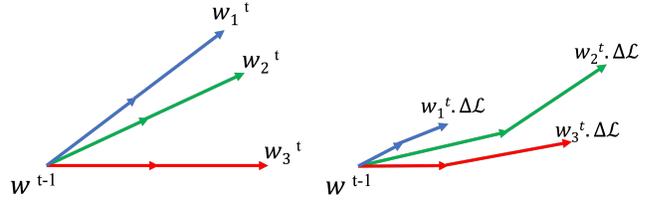


Fig. 1. Left: Weight divergence in *FedAvg* due to data heterogeneity. Right: *FedCurv* induces a regularization parameter to minimize divergence to update weights according to less critical parameters

inadvertent leakage of sensitive training data. Such privacy concerns present a substantial barrier to the continued advancement and deployment of DL technologies [12]. Centralized aggregation of private data for ML models poses significant challenges, particularly in terms of security, privacy, and confidentiality [13]. Consequently, retaining control over data without external dissemination to prioritize the security and privacy of data owners is an essential requirement for healthcare applications. This requirement has led to the development of FL methodologies. Auditability and verifiability are essential components for establishing trustworthiness in FL. Several studies proposed blockchain-empowered FL approaches to enhance transparency, accountability, verifiability and the independent validation of FL processes [9]. Authors in [14] proposed a blockchain based trusted execution platform to secure each client’s local model training data and provide multi-signature-based global model verification to enhance model auditability. In [15] authors develop smart contract empowered local data training policy control and verification of the integrity of trained models.

In the literature, first-order federated edge learning techniques, which rely solely on gradient information, are frequently utilized due to their robustness in distributed settings and minimal local computational requirements [16]. Conversely, second-order methods, utilize both gradient and curvature information, thereby facilitating improved descent direction selection and significantly accelerating convergence. This acceleration reduces the number of communication rounds required to achieve convergence, making second-order methods particularly advantageous in heterogeneous FL environments.

Continual learning and FL approaches employ diverse strategies to address challenges like catastrophic forgetting, task interference, and communication efficiency [17]. Elastic Weight Consolidation (EWC) [18] mitigates catastrophic forgetting by restricting parameter updates critical to previous tasks using FIM, ensuring solutions compatible with both old and new tasks. In contrast, Incremental Moment Matching (IMM) [19] models the posterior distribution of parameters for multiple tasks as a mixture of Gaussians to harmonize task-specific knowledge. Stable SGD [20] enhances performance by dynamically adjusting hyperparameters and incrementally reducing the learning rate upon encountering new tasks. For FL, *FedCurv* [21] adapts a modified EWC framework to minimize dispar-

loss function $\mathcal{L}_k(\theta)$, which combines the standard local loss with a curvature penalty term. The penalty term involves the squared difference between local and global parameters weighted by the diagonal FIM. The gradient of the regularized loss is then computed as the sum of the gradient of the local loss and a regularization term scaled by a factor λ . The model parameters are updated using stochastic gradient descent (SGD) for E local epochs based on the gradient of this regularized loss. $F_k[i]$ corresponds to diagonal entry of F_k for parameter i .

$$\nabla \log p(y|x; \theta) \quad (2)$$

above equation shows the gradient of the log-likelihood with respect to parameters.

Compute local training with curvature regularization to minimize the regularized loss:

$$\mathcal{L}_k(\theta) = \underbrace{\text{Local Loss}}_{\mathcal{L}_{\text{LocalLoss}}} + \underbrace{2(\theta - \theta_{\text{global}})^T \cdot \text{diag}(F_k) \cdot (\theta - \theta_{\text{global}})}_{\text{Curvature Penalty}} \quad (3)$$

where λ corresponds to regularization strength. Gradient calculation is computed as:

$$\nabla \mathcal{L}_k = \nabla \mathcal{L}_{\text{LocalLoss}} + \lambda \cdot \text{diag}(F_k)(\theta - \theta_{\text{global}}) \quad (4)$$

Subsequently, update θ via SGD for E epochs:

$$\theta_{\text{local}} = \theta_{\text{global}} - \eta_{\text{local}} \cdot \nabla \mathcal{L}_k \quad (5)$$

After local training, compute the gradient for server by computing the gradient of the **original loss** (without regularization) at θ_{local} :

$$g_k = \nabla \mathcal{L}_{\text{LocalLoss}}(\theta_{\text{local}}) \quad (6)$$

In Post-training, clients transmit θ_{local} curvature and g_k gradients to the aggregation server through blockchain. This approach inherently safeguards data privacy by ensuring sensitive information remains within its original jurisdiction, serving as the primary defence against privacy breaches in sensitive healthcare applications. These updates are systematically broadcast to a private blockchain network ensuring auditability and transparency.

B. Blockchain Layer

The blockchain layer serves as a foundational component of the proposed system, providing an immutable and transparent ledger to enhance security, accountability, and compliance in healthcare ecosystems. By design, the blockchain records all transactions including consent management, data access requests, and model updates in a tamper-evident manner, ensuring traceability across the ecosystem.

For instance, patient consent parameters, such as permissible data usage and authorized entities, are codified within smart contracts. These self-executing agreements autonomously verify compliance with predefined policies,

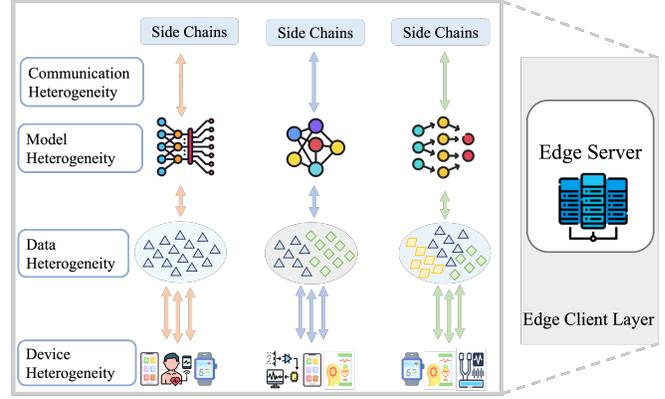


Fig. 3. Edge Client Layer

thereby eliminating manual oversight and reducing the risk of unauthorized access. Each step of the aggregation workflow such as server update requests, participant submissions, and the generation of the global model is cryptographically hashed and immutably logged on the blockchain, enabling independent verification of procedural integrity.

Blockchain broadcasts stored the finalized global model parameters in encrypted form using public-key infrastructure pki , ensuring only authorized entities with corresponding decryption keys can access sensitive insights. Subsequent training rounds are initiated by broadcasting these encrypted parameters via the blockchain to aggregation servers, thereby maintaining synchronization while preserving security. Access control is further reinforced through blockchain mediated authentication; for example, researchers requesting model access trigger smart contracts that validate their permissions against patient defined policies prior to granting approval. All such interactions, including consent modifications and data retrieval attempts, are permanently recorded on the ledger, creating auditable trails for regulatory review. This architecture not only ensures adherence to standards such as HIPAA and GDPR but also empowers patients with visibility into data usage through transparent audit logs. By integrating decentralized consensus mechanisms, cryptographic encryption, and automated policy enforcement, the system establishes a robust framework for privacy preserving collaboration, balancing operational transparency with uncompromising data security.

C. Cloud Server Layer

Cloud servers are deployed virtually over the network and working as aggregation servers, collecting locally computed θ_{local} and gradients g_k from each edge client k . Compute aggregated curvature F_{global} to get important scores (FIMs) from all clients and average gradients g_{global} .

$$F_{\text{global}} = \frac{1}{K} \sum_{k=1}^K F_k \quad (7)$$

$$g_{\text{global}} = \frac{1}{K} \sum_{k=1}^K g_k \quad (8)$$

Secondly, compute inverse FIM for diagonal F_{global}

$$F_{\text{global}}^{-1}[i] = \frac{1}{F_{\text{global}}[i] + \epsilon} \quad (9)$$

Inverse FIM scores invert the average importance scores, which ensures priority of important parameters (with large curvature values) gets smaller updates. It helps the model learn from new data without forgetting important knowledge from previous clients.

Following this, server clouds update the global model by applying curvature-scaled update:

$$\theta_{\text{global}}^{\text{new}} = \theta_{\text{global}} - \eta_{\text{global}} \cdot \left(F_{\text{global}}^{-1} \odot g_{\text{global}} \right) \quad (10)$$

Finally, calculated $\theta_{\text{global}}^{\text{new}}$ global updates are broadcast to the clients through blockchain broadcasts using smart contracts.

D. Communication Model

The system employs a layered communication model and protocol architecture designed to ensure privacy-preserving, secure, and auditable operations across asynchronous phases. Data collection is initiated by sensors and edge client devices, which acquire raw physiological metrics such as blood glucose levels and electrocardiogram signals. HTTPS/gRPC encrypts federated learning client-server communications, while MQTT optimizes lightweight data transfer from IoT/wearables to the edge client servers.

Our proposed access scheme is used along with the widely used Elliptic Curve Integrated Encryption Scheme (ECIES), Child Key Derivation function (CKD), and the Elliptic Curve Digital Signature Algorithm (ECDSA) are cryptographic techniques to ensure secure data storage and communication. ECIES works by independently deriving a bulk encryption key and a MAC key from a common secret. The data is first encrypted under a symmetric cipher, and then the ciphertext is authenticated under an authentication scheme. Finally, the common secret is encrypted under the public part of a public/private key pair. The CKD function is used for managing data in batches; child key derivation functions are used in hierarchical deterministic wallets (HD wallets). It helps in generating a tree of keys from a single master key, which can be very useful for managing multiple keys securely and systematically. ECDSA is a digital signature algorithm that is used for secure key sharing for both data and communication. The ECDSA ensures that the data and communication are coming from the stated sender (authenticity), have not been altered in transit (integrity), and repudiation by the sender can be disputed (non-repudiation). Implementing ECIES, CKD, and ECDSA in this proposed system provides a robust framework that ensures secure data storage and communication. The ECIES

offers a strong encryption scheme for data protection, the CKD provides an efficient way to manage data in batches, and the ECDSA guarantees secure key sharing and data authenticity.

Following this, local model training at edge client level is completed using *FedCurv*, model local weights F_k, g_k are transmitted through private ethereum network, ensuring the confidentiality of individual client k data weights, preventing re-identification. Subsequently, processed data is transmitted to the model aggregation servers via a blockchain network, where aggregated global models are cryptographically hashed and immutably logged on the blockchain ledger. Transaction validation occurs through a distributed consensus protocol, ensuring integrity. Central to access governance is a patient-centric framework: patients configure data sharing permissions via a portal, triggering Ethereum based smart contracts, while researchers and clinicians submit queries via blockchain transactions, which undergo automated authorization checks. Approved requests retrieve models or insights from secure decentralized storage systems such as IPFS, with all access events (identity, timestamp, purpose) permanently recorded on-chain for auditability. Role-Based Access Control (RBAC) policies are programmatically enforced through smart contracts.

IV. EXPERIMENTAL SETUP

To evaluate the performance of our proposed *BFEL* framework through experimental testbeds, optimized *FedCurv* is employed as federated learning algorithm to address data heterogeneity across clients. *FedCurv* is compared with baseline federated algorithm *FedAvg*. The federated learning framework involves 10 local training rounds per participant and 20 global aggregation rounds. We utilise three benchmark datasets: MNIST, CIFAR10, and MedMNIST. MNIST and CIFAR10 were selected as standard benchmarks in neural network research, while MedMNIST specifically its PathMNIST subset, derived from 2D image classification. On the client side, each edge device performs local training while incorporating a curvature-aware regularization term. This regularization penalizes deviations from the global model based on the estimated importance of each parameter, thereby preserving critical knowledge and improving model stability. On the server side 2, the curvature matrices and gradients collected from the clients are aggregated. The server 1 then applies a curvature-scaled update to the global model, ensuring that parameter updates are inversely proportional to their estimated importance. This approach enables more efficient and robust federated optimization in non-iid settings. The private ethereum blockchain network is utilised as a service layer demonstrating its capability to uphold auditability, verifiability, and availability while ensuring privacy preservation in edge-FL settings.

A. Parameter Settings

MNIST, a standard benchmark for image classification, includes 60,000 images (28×28 pixels), split into 50,000

training and 10,000 test samples, providing a baseline for foundational algorithm validation. MedMNIST subclass PathMNIST (28×28 pixels), a more challenging alternative, comprises 35,000 training images and 8000 testing images in non-iid, enabling evaluation of model generalizability in scenarios with higher intra-class variability. CIFAR-10, a widely used dataset for object recognition, offers 60,000 color images (32×32 pixels). Together, these datasets simulate real-world edge-FL challenges, such as decentralized computation and heterogeneous data privacy requirements, while validating the framework’s ability to balance transparency, security, and efficiency in privacy-sensitive environments.

B. Network Configuration:

This study utilizes two widely recognized neural architectures: a multi-layer perceptron (MLP) and a convolutional neural network (CNN). These models served as the foundational deep learning frameworks for training classification systems within a federated learning setup, simulating server client training scenarios. The experiments aimed to assess how effectively each algorithm handles non-iid data and preserves knowledge across distributed medical and non-medical imaging tasks. For MNIST and PathMNIST (image datasets), the CNN consists of two convolutional layers, each followed by a max-pooling layer, and concludes with two fully connected layers. To accommodate the RGB input channels of the CIFAR-10 dataset, the architecture is modified while retaining the core structure. Both configurations employ the SGD optimizer with a learning rate of 0.001.

C. Hardware and Software Configuration

We utilise raspberry Pi3 model B+ minicomputers as edge servers (manager nodes of side chains) and lightweight nodes are implemented using STM32F427 development boards (low-power ARM Cortex M3,M4 and M7 processors), which are used for high-speed implementation of asymmetric cryptographic algorithms. The elliptic curve digital signature algorithm (ECDSA) is used to generate public and private keys, and device authentication mechanisms. The STMicroelectronics X-CUBE-CRYPTOLIB library is utilized to implement several standard cryptographic algorithms with the ARM Cortex-M series processors. System components are developed using Go (Golang), solidity to write smart contracts and deployed using remix IDE, and MetaMask handles concurrent transactions and interactions with third-party cloud services. A PoS consensus ensures block validation, while the gossip protocol enables fast, resilient message propagation and node synchronization. Figure 8 depicts the system’s initialized nodes.

V. PERFORMANCE EVALUATIONS

To validate the effectiveness of *BFEL*, we conducted experiments across varying task sequence configurations

Algorithm 1 FedCurv Server Side Computation

Require: Initial model θ_{global}^0 , total rounds T , number of clients K , server learning rate η_{global} , numerical stability ϵ

Ensure: Trained global model θ_{global}^T

- 1: Initialize global model
 - 2: **for** round $t = 1$ **to** T **do**
 - 3: Broadcast θ_{global} to all participating clients
 - 4: Collect client updates
 - 5: Aggregate Fisher Information Matrices
 - 6: Aggregate gradients
 - 7: Compute inverse Fisher information
 - 8: Update global model
 - 9: **return** θ_{global}
-

Algorithm 2 FedCurv Client Side Computation

Require: Global model parameters θ_{global} , local dataset \mathcal{D}_k , regularization strength λ , local epochs E , learning rate η_{local}

Ensure: Updated Fisher Information Matrix F_k , gradient g_k

- 2: Client Update
 - Compute Fisher Information Matrix (FIM)
 - 4: Initialize local model
 - for** epoch = 1 **to** E **do**
 - 6: **for** each batch $(x_b, y_b) \in \mathcal{D}_k$ **do**
 - Compute regularized loss
 - 8: Compute gradient
 - Update local model
 - 10: Compute Server Gradient
 - return** $\{F_k, g_k\}$
-

using *FedCurv* and compared its performance against established baselines using *FedAvg*. The results are analyzed through multiple perspectives, offering distinct insights. Code is available publically at our github account¹.

A. Federated Simulation Results

From the epochs per round standpoint, accuracy consistently improves as the number of local training epochs increases across all settings and algorithms. It depicts the close alignment of local optima with global optima, making extended local training within each round advantageous when maintaining a fixed number of communication rounds. Accuracy of *FedAvg* increases sharply after each epoch as compared to *FedCurv* at the base level depicted in figure 7 and 6 as well as edge client level, which shows the minimum divergence of results after each round in *FedCurv*, makes it more consistent and according to the previous weights of edge client device.

Overall comparison of performance, *FedCurv* designed primarily to address non-iid data challenges in federated learning, surprisingly outperforms *FedAvg* even in uniform data settings. Notably, *FedCurv* often achieves superior

¹<https://github.com/AnumNawazKahloon/FedCurv>

accuracy after 100 rounds, indicating a potential need for extended training phases to reach convergence compared to *FedAvg*. Regarding communication efficiency, reducing the frequency of communication rounds while keeping the total number of training epochs constant yields better model performance, implying that less frequent parameter exchanges may enhance learning stability or optimization. The experimental setup utilized an *SGD* optimizer with adaptive learning rate decay (reduced by a factor of 3 every 5 epochs if validation loss stagnated), a mini-batch size of 20, 20 rounds per task, and 1 epoch per round. For the MLP configuration, adjustments included a reduced mini-batch size of 10 and an initial learning rate of $1e-4$, while client sampling fractions of 0.25 and 0.05 were applied at each round. Hyperparameters λ_1 and λ_2 were fixed at $[1e-1, 4e-1]$ and 100, respectively, across all experiments. These findings collectively highlight the interplay between local training intensity, algorithmic robustness, and communication strategies in federated learning frameworks.

1) *Client side training*: In the edge client side, federated learning experiments, *FedCurv* demonstrated consistent performance as shown in 4 over *FedAvg* 5, particularly in non-iid settings with CNN models. On the PathMNIST dataset using CNN, *FedCurv* achieved significantly better accuracy of around 86% and exhibited stable convergence, while *FedAvg* struggled with fluctuations and lower accuracy of 75%. Similarly, for CNN on MNIST, *FedCurv* outperformed *FedAvg* with a final test accuracy around 95%, compared to *FedAvg*'s noisier convergence and lower accuracy of 85%. The difference was significant on CIFAR-10 with CNN, where *FedCurv* showed a gradual increase to 50% accuracy, outperforming *FedAvg*, which displayed highly unstable learning and plateaued around 40%. After each epoch round, *FedCurv* maintains low divergence in non-iid settings.

2) *Server side training*: However, during server side base experiments across the MNIST, PathMNIST, and CIFAR-10 datasets. *FedAvg* consistently outperformed *FedCurv* in terms of final testing accuracy and convergence behavior. For CNN based experiments, *FedAvg* achieved slightly higher accuracies across all datasets. On the PathMNIST dataset, *FedAvg* reached around 90% accuracy, while *FedCurv* trailed slightly at approximately 88%. A similar trend was observed for MNIST, where *FedAvg* achieved about 99% accuracy compared to *FedCurv*'s 98%. On the more complex CIFAR-10 dataset, *FedAvg* demonstrated better generalization, achieving approximately 72% accuracy, while *FedCurv* lagged behind at around 68%.

B. Layer2 Ethereum Implementation

The second layer ethereum network is implemented using a polygon sidechain structure which leverages proof-of-stake (PoS) consensus. It consists of a main blockchain layer running on cloud servers and a sub-blockchain network comprised of sidechains. The main blockchain layer implements a consortium blockchain, while the sub-blockchain layer consists of multiple private sidechains,

each deployed on individual edge servers to facilitate localized model training in *BFEL*. Within each sub-blockchain, the corresponding edge server functions as a miner node, responsible for performing local model training and updating local model weights based on the global model parameters after each training epoch. Subsequently, each edge server transmits its locally trained results to the main blockchain layer, where global model parameters are aggregated.

The edge servers are also tasked with collecting and packaging model training outputs (treated as transactions) into candidate blocks, which are then published to their respective private blockchains. The main blockchain layer is operated by cloud servers configured as a consortium blockchain. Each registered user organization such as hospitals or governmental agencies within the consortium is permitted to establish its own private sidechain. Furthermore, to reduce potential latency caused by complex task assignment mechanisms, transaction packaging responsibilities in both private and consortium blockchains are delegated to designated manager nodes.

C. Security Analysis

We have conducted a performance evaluation of three different types of STM32F427 M series processors within the framework of asymmetric cryptography. We utilized the X-CUBE-CRYPTOLIB library to implement the ECDSA. To ascertain the statistical error of the results obtained over the number of executions, we calculated the mean, standard deviation, and standard error using the appropriate equations.

The execution time was determined, which encompasses the total time required for key generation, encryption, and decryption using ECDSA. To identify the optimal execution time of ECDSA, we examined the records of different numbers of executions for each processor. The execution time for each processor for ECDSA is visually represented in figure 9 subsection (a). For a comprehensive analysis, the execution time was calculated in terms of mean, standard deviation, and standard error for each processor. The estimated execution time of ECDSA for processor M3 is $26.352\text{ s} \pm 0.002\text{ s}$, and the execution time for M4 processor is $1.451\text{ s} \pm 0.007\text{ s}$ and $1.167\text{ s} \pm 0.002\text{ s}$ for M7. Based on the results, the average execution times of M3 processors are 17.253 seconds, while the execution times for processors M4 and M7 are 1.462 seconds and 1.156 seconds, respectively. The data suggests that the M4 and M7 processors exhibit superior performance in executing ECDSA. These time measurements facilitate easy planning and adjustments to determine the delay tolerance in the network.

Power consumption is an important parameter of micro-controllers, we determine power utilization during the execution of cryptographic algorithms. The power consumed was determined by measuring the voltage across a shunt resistance R . The current consumption of the processors

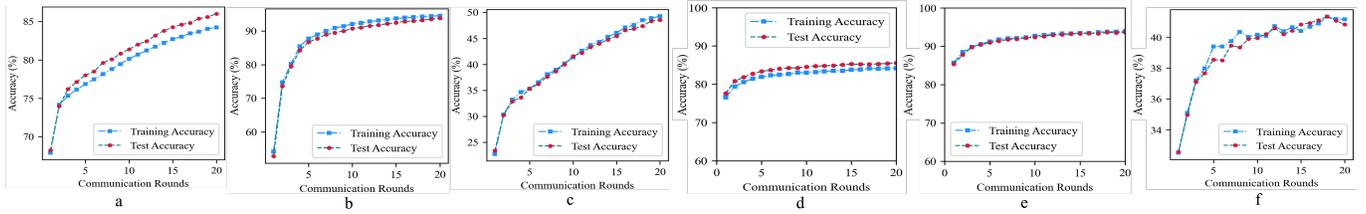


Fig. 4. *FedCurv* for Federated Non-iid (a) CNN PathMnist (b) CNN Mnist (c) CNN Cifar (d) MLP PathMnist (e) MLP Mnist (f) MLP Cifar

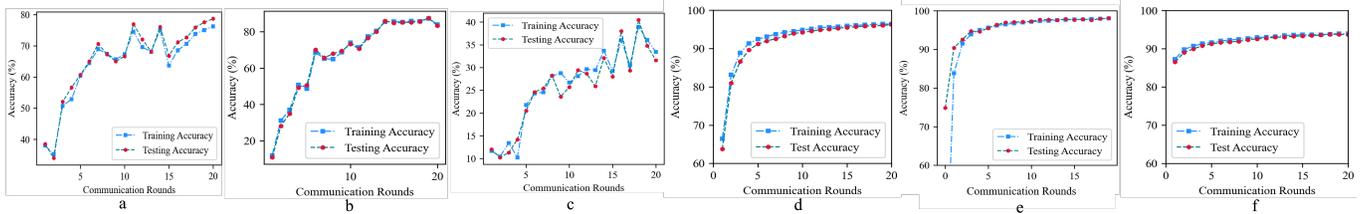


Fig. 5. *FedAvg* for Federated (a) Non-iid Fed CNN PathMnist (b) Non-iid Fed CNN Mnist (c) Non-iid Fed CNN Cifar, (d) Iid Fed CNN PathMnist (e) Iid Fed CNN Mnist (f) Iid Fed CNN Cifar

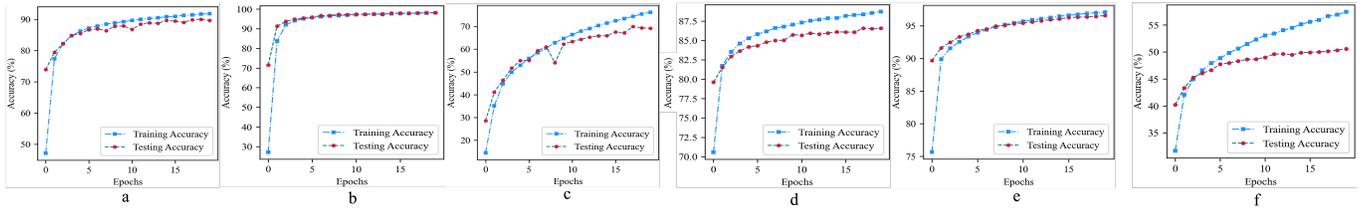


Fig. 6. *FedCurv* for Base (a) CNN PathMnist (b) CNN Mnist (c) CNN Cifar (d) MLP PathMnist (e) MLP Mnist (f) MLP Cifar

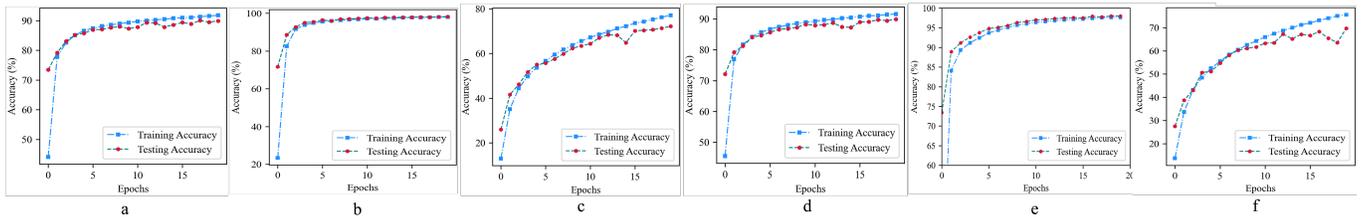


Fig. 7. *FedAvg* for Base (a) CNN PathMnist (b) CNN Mnist (c) CNN Cifar (d) MLP PathMnist (e) MLP Mnist (f) MLP Cifar

```
geth --identity "node1" --networkid 52419 --syncmode "full" --datadir "~/Nodes/node1/"
--nodiscover --ws --ws.port=8011 --ws.origins="*" --ws.addr "127.0.0.1"
--http --http.corsdomain "*" --http.port 8041 --http.api
"db,eth,net,personal,admin,miner,web3,pledge" --port 30301 --authrpc.port 8451 --unlock
--password ~/Nodes/password.sec --unlock 0x10c5bd4f90a6e3cb63f93a761a8192f4f12a6b74
--allow-insecure-unlock --mine --miner.etherbase=0x10c5bd4f90a6e3cb63f93a761a8192f4f12a6b74
--cache 2048 --log.debug --ipcpath "~/Nodes/node1/geth.ipc" console
```

Manager Node

```
geth --identity "node3" --networkid 52419 --syncmode "full" --datadir "~/Nodes/node3"
--nodiscover --ws --ws.port=8013 --ws.origins="*" --ws.addr "127.0.0.1"
--http --http.corsdomain "*" --http.port "8043" --http.api
```

Light weight Node

Fig. 8. Manager nodes and lightweight nodes after layer2 ethereum initialization

was calculated using ohm's law. To determine the average power consumption, ECDSA was executed for 15 runs and a comparison of power consumption is presented in figure 9(b). The average power consumption by M3 and M4 was $\pm 200\text{mW}$, whereas M7 used an average of $\pm 290\text{mW}$. Results indicate the superior performance of M4 cortex microcontrollers are best fit while consuming fewer resources.

Table II illustrates how different security measures have been taken into account. A deterministic wallet is used for CKD functions to determine a child's key from a parent's key. Using this technique, each batch of data to be encrypted in the device is given a unique secret key [24]. The 512-bit hash is calculated according to the parent's public key (public and private keys are 256 bits) and the desired child index. It is impossible to deduce the original

TABLE I
AVERAGE *FEDAVG* VERSUS *FEDCURV* EDGE CLIENT-SIDE ACCURACY AND AVERAGE SERVER-SIDE BASE LEARNING ACCURACY RESULTS

Dataset	Model	client Acc <i>FedAvg</i>	client Acc <i>FedCurv</i>	Base Acc <i>FedAvg</i>	Base Acc <i>FedCurv</i>
PathMNIST	CNN	~75%	~86%	~90%	~88%
MNIST	CNN	~85%	~95%	~99%	~97%
CIFAR-10	CNN	~40%	~50%	~72%	~68%
PathMNIST	MLP	~95%	~85%	~91%	~88%
MNIST	MLP	~94%	~94%	~97%	~95%
CIFAR-10	MLP	~45%	~42%	~70%	~50%

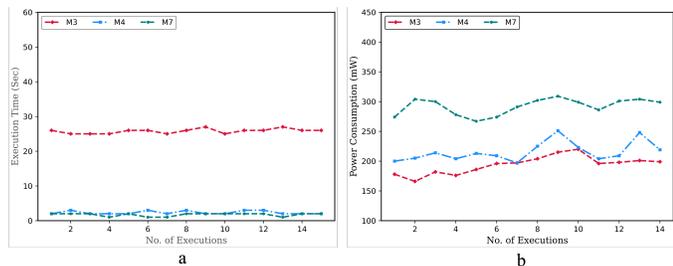


Fig. 9. (a) Execution time and (b) Average power consumption, during ECDSA implementation using STM32F427 M series processors

parent key from the n th-child key because of the one-way hashing used in the process. This process appears to generate random numbers due to the additions *modulo n*.

TABLE II
SECURITY MEASURES

Parameter	Implementation
Authorization	Public key cryptography to encrypt CSK
Confidentiality	Public key cryptography and proof of authority
Integrity	Broadcast hash of each data_batch
Availability	Achieved by limiting number of requests
Anonymity	Discard raw data and store only processed information

D. Scalability Analysis

Edge clients based DLTs offer several benefits but also face inherent limitations in terms of scalability, which limit their application range. Nevertheless, *BFEL* handles this challenge using the side chains concept, gossip protocol, and PoS consensus. FoBSim simulation tool² is utilized to check the scalability of the proposed model. Manager nodes ranging from 5 to 500 were used to check the performance matrix of the proposed model and measure the total time required to complete the request procedure at edge clients versus at the cloud layer.

The edge client measures and divides the time needed to complete a transaction into three sections: Time to retrieve data (TRD), time for checking the transaction (VTR), and time for confirming the transaction (TCT).

²<https://github.com/sed-szeged/FobSim>

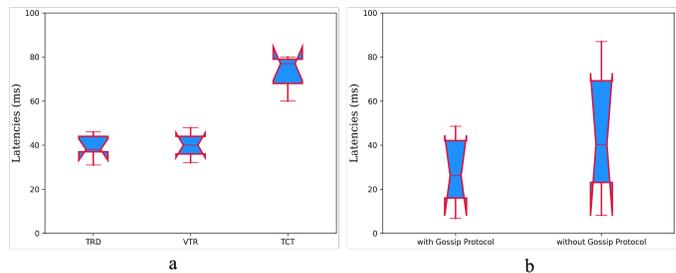


Fig. 10. (a) Latencies to retrieve meta data (TRD), transaction validation time (VTR) needed for single transaction request, and time required to confirm one transaction (TCT), (b) Impact of concurrent transactions on end-to-end transaction latency using gossip protocol versus without gossip protocol

Figure 10(a) presents the results of the measurements. Based on the available resources, it is impressive that TRD requires only 34.6 milliseconds on average, VTR 36 milliseconds on average, and TCT 73.6 milliseconds on average. Additionally, it is essential to note that TCT also relies on the network, which in this experiment was adversely affected by our shared Wi-Fi's slow response time, causing the time to be extended overall.

Gossip protocols are integrated to improve the scalability of the proposed network model and *public key infrastructure* (PKI) is utilised for cryptographically signing model updates and transactions. This architecture integrates end-to-end privacy preservation, tamper evident logging, and granular access controls, ensuring regulatory compliance and transparency across all operational phases.

The impact of the gossip protocol on the total elapsed time at edge clients was evaluated to demonstrate its scalability enhancement. Figure 10(b) illustrates the improved scalability of the proposed access scheme when utilizing the gossip protocol, as compared to transactions conducted without it.

Figure 11(a) shows that the cloud layer utilises around double the time as compared to edge clients to complete transaction requests during concurrent transactions starting from 5 to 100 transactions at a time.

$end\text{-}to\text{-}end\ delay = request\ initialization\ by\ interested\ buyer + time\ to\ retrieve\ metadata + response\ time\ by\ manager\ nodes + time\ to\ confirm\ one\ transaction$

Figure 11(a) illustrates that increasing concurrent trans-

action requests leads to increased end-to-end delays. This study’s results demonstrate the proposed model’s effectiveness for autonomously implementing data sharing in information-critical systems. Using this trustless structure, data trade becomes more reliable and transparent. We have concluded that single-board computers can act as data and transaction managers, with no need for third-party cloud services, as the necessary computation makes space for other edge services and data processing processes to run simultaneously. However, a parallel number of transactions will cause a significant delay.

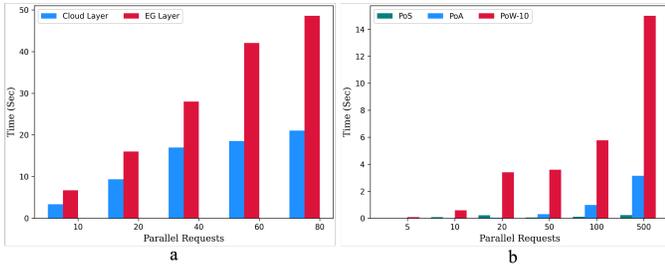


Fig. 11. (a) Impact of concurrent transactions on end-to-end transaction latency on *BFEL* versus cloud services, (b) Average Block Confirmation Time using PoS, PoW-10, and PoA

Block confirmation time was also measured using different numbers of manager nodes, and the average time is illustrated in the figure for proof of stake versus two famous consensus algorithms PoW and PoA 11(b). *BFEL* does not present a scalability issue due to the fact that it works in a private P2P network that can be segmented into side chains. It does not require edge clients to process many requests per minute. The manager nodes must check whether new requests for data have been received after every time T . If it has a queue of requests, the manager nodes will respond to each request one at a time.

VI. DISCUSSION AND ANALYSIS

In this study, we propose and develop a personalized healthcare system *BFEL* based on second order FEL. Second-order FL methods bring a substantial benefit of personalized training through their capability to use loss function curvature data to improve personalized training procedures during local data training. *BFEL* leverages blockchain as a service layer to ensure the privacy, transparency and confidentiality of data owners while maintaining auditability, verifiability, availability, and robust privacy protection in FEL environments. Upon model training completion, the global model and local parameters are securely stored on the blockchain and subsequently distributed to the federated learning edge clients and cloud server and vice versa in accordance with the established policy. Smart contracts are utilized to broadcast policies while providing automated confirmation and distribution operations.

Optimized *FedCurv* is used to minimize the number of communication rounds needed for personalized non-iid and heterogeneous distributed training across edge client

devices to enhance personalized health monitoring and prediction. Performance evaluations demonstrate better performance of *FedCurv* and resistance capabilities than *FedAvg* using CNNs and MLPs in non-iid conditions. Key evaluation metrics including throughput, accuracy, privacy, and scalability analysis has been analyzed. Analysis of the experimental results demonstrates that the proposed method preserves the accuracy of the machine learning process while ensuring auditability and verifiability throughout the training and aggregation procedures. Privacy protection is achieved through second-order FEL using *FedCurv* and public key encryption ECDSA and ECIES, while latency and throughput were evaluated by measuring communication transactions on a permissioned blockchain and compared against a benchmark model *FedAvg*. The results demonstrate that the proposed model outperforms the benchmark by achieving enhanced privacy, accuracy and scalability. Results underscore better stability and accuracy of the proposed framework *BFEL* in heterogeneous settings and non-iid environments where patient data varies widely.

REFERENCES

- [1] A. Simkó, A. Garpebring, J. Jonsson, T. Nyholm, and T. Löfstedt, “Reproducibility of the methods in medical imaging with deep learning,” in *Medical Imaging with Deep Learning*. PMLR, 2024, pp. 95–106.
- [2] B. Liu, M. Ding, S. Shaham, W. Rahayu, F. Farokhi, and Z. Lin, “When machine learning meets privacy: A survey and outlook,” *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–36, 2021.
- [3] S. AbdulRahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M. Guizani, “A survey on federated learning: The journey from centralized to distributed on-site learning and beyond,” *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5476–5497, 2020.
- [4] H. Zhu, J. Xu, S. Liu, and Y. Jin, “Federated learning on non-iid data: A survey,” *Neurocomputing*, vol. 465, pp. 371–390, 2021.
- [5] D. Javeed, M. S. Saeed, P. Kumar, A. Jolfaei, S. Islam, and A. N. Islam, “Federated learning-based personalized recommendation systems: An overview on security and privacy challenges,” *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 2618–2627, 2023.
- [6] J. Qi and M.-H. Hsieh, “Federated quantum natural gradient descent for quantum federated learning,” in *Federated Learning*. Elsevier, 2024, pp. 329–341.
- [7] S. M. Hamidi and L. Ye, “Distributed quasi-newton method for fair and fast federated learning,” *arXiv preprint arXiv:2501.10877*, 2025.
- [8] D. Jhunjunwala, S. Wang, and G. Joshi, “Towards a theoretical and practical understanding of one-shot federated learning with fisher information,” in *Federated Learning and Analytics in Practice: Algorithms, Systems, Applications, and Opportunities*, 2023.
- [9] J. Zhu, J. Cao, D. Saxena, S. Jiang, and H. Ferradi, “Blockchain-empowered federated learning: Challenges, solutions, and future directions,” *ACM Computing Surveys*, vol. 55, no. 11, pp. 1–31, 2023.
- [10] H. Bolhasani, M. Mohseni, and A. M. Rahmani, “Deep learning applications for iot in health care: A systematic review,” *Informatics in Medicine Unlocked*, vol. 23, p. 100550, 2021.
- [11] A. López Martínez, M. Gil Pérez, and A. Ruiz-Martínez, “A comprehensive review of the state-of-the-art on security and privacy issues in healthcare,” *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, 2023.
- [12] Y. Otoum, N. Gottimukkala, N. Kumar, and A. Nayak, “Machine learning in metaverse security: Current solutions and future challenges,” *ACM Computing Surveys*, vol. 56, no. 8, pp. 1–36, 2024.

- [13] Z. Chen, J. Liu, Y. Shen, M. Simsek, B. Kantarci, H. T. Mouftah, and P. Djukic, "Machine learning-enabled iot security: Open issues and challenges under advanced persistent threats," *ACM Computing Surveys*, vol. 55, no. 5, pp. 1–37, 2022.
- [14] A. P. Kalapaaking, I. Khalil, and M. Atiquzzaman, "Blockchain-enabled and multisignature-powered verifiable model for securing federated learning systems," *IEEE Internet of Things Journal*, vol. 10, no. 24, pp. 21 410–21 420, 2023.
- [15] —, "Smart policy control for securing federated learning management system," *IEEE Transactions on Network and Service Management*, vol. 20, no. 2, pp. 1600–1611, 2023.
- [16] D. C. Nguyen, Q.-V. Pham, P. N. Pathirana, M. Ding, A. Seneviratne, Z. Lin, O. Dobre, and W.-J. Hwang, "Federated learning for smart healthcare: A survey," *ACM Computing Surveys (Csur)*, vol. 55, no. 3, pp. 1–37, 2022.
- [17] Y. Ma, Z. Xie, J. Wang, K. Chen, and L. Shou, "Continual federated learning based on knowledge distillation." in *IJCAI*, 2022, pp. 2182–2188.
- [18] S. Wen and L. Itti, "Overcoming catastrophic forgetting problem by weight consolidation and long-term memory," *arXiv preprint arXiv:1805.07441*, 2018.
- [19] S.-W. Lee, J.-H. Kim, J. Jun, J.-W. Ha, and B.-T. Zhang, "Overcoming catastrophic forgetting by incremental moment matching," *Advances in neural information processing systems*, vol. 30, 2017.
- [20] R. Jin, Y. Huang, X. He, H. Dai, and T. Wu, "Stochastic-sign sgd for federated learning with theoretical guarantees," *arXiv preprint arXiv:2002.10940*, 2020.
- [21] N. Shoham, T. Avidor, A. Keren, N. Israel, D. Benditkis, L. Mor-Yosef, and I. Zeitak, "Overcoming forgetting in federated learning on non-iid data," *arXiv preprint arXiv:1910.07796*, 2019.
- [22] J. Ayeelyan, S. Utomo, A. Rouniyar, H.-C. Hsu, and P.-A. Hsiung, "Federated learning design and functional models: Survey," *Artificial Intelligence Review*, vol. 58, no. 1, pp. 1–38, 2025.
- [23] C. Ren, H. Yu, H. Peng, X. Tang, B. Zhao, L. Yi, A. Z. Tan, Y. Gao, A. Li, X. Li *et al.*, "Advances and open challenges in federated foundation models," *IEEE Communications Surveys & Tutorials*, 2025.
- [24] A. M. Antonopoulos, *Mastering bitcoin: Programming the open blockchain.* " O'Reilly Media, Inc.", 2017.