Decentralized Defense: Leveraging Blockchain against Poisoning Attacks in Federated Learning Systems

Rashmi Thennakoon*, Arosha Wanigasundara†, Sanjaya Weerasinghe ‡, Chatura Seneviratne§,
Yushan Siriwardhana¶, Madhusanka Liyanage‖
*†‡§Dept of Electrical and Information Engineering, University of Ruhuna, Sri Lanka.
¶Centre for Wireless Communications, University of Oulu, Finland.
‖School of Computer Science, University College Dublin, Ireland.
Email: *eg183480@engug.ruh.ac.lk, †eg183490@engug.ruh.ac.lk, ‡eg183494@engug.ruh.ac.lk,
§chatura@eie.ruh.ac.lk, ¶ yushan.siriwardhana@oulu.fi ‖ madhusanka@ucd.ie

***Abstract*—Federated learning (FL) has become the next generation of machine learning (ML) by avoiding local data sharing with a central server. While this becomes a major advantage to client-side privacy, it has a trade-off of becoming vulnerable to poisoning attacks and malicious behavior of the central server. As the decentralization of systems enhances security concerns, integrating decentralized defense for the existing FL systems has been extensively studied to eliminate the security issues of FL systems. This paper proposes a decentralized defense approach to FL systems with blockchain technology to overcome the poisoning attack without affecting the existing FL system's performance. We introduce a reliable blockchain-based FL (BCFL) architecture in two different models, namely, Centralized Aggregated BCFL (CA-BCFL) and Fully Decentralized BCFL (FD-BCFL). Both models utilize secure off-chain computations for malicious mitigation as an alternative to high-cost on-chain computations. Our comprehensive analysis shows that the proposed BCFL architectures can defend in a similar manner against poisoning attacks that compromise the aggregator. As a better measure, the paper has included an evaluation of the gas consumption of our two system models.**

***Index Terms*—Federated Learning, Poisoning Attacks, Single Point of Failure, Blockchain, Off-chain computation**

I. Introduction

Federated learning (FL) has emerged as a promising solution to address privacy concerns, communication overhead, and the higher cost associated with conventional centralized machine learning (ML) methods. In 2016 [1], Google introduced FL as an innovative ML approach that addresses the limitations of conventional ML. FL allows collaborative training of a shared global model across decentralized local FL clients without the need to share the original local data with a central server.

However, FL systems still endure various limitations in terms of security [1]. Primarily, FL systems are vulnerable to poisoning attacks due to their distributed nature, which presents significant security threats in the local model training process. This includes data poisoning attacks, wherein a subset of malicious clients intentionally submit false or manipulated data to influence the global model, and model poisoning attacks, where malicious attackers compromise the local training models [2]. Moreover, FL systems are also susceptible to the single point of failure (SPoF) caused by the unreliable central server, malicious aggregation of the global model at the central server, and model inversion attacks [3], wherein adversaries attempt to extract sensitive information from the aggregated model. These attacks can ultimately compromise the global model by reducing the global accuracy while affecting the system's overall performance.

Hence, FL systems require robust security measures to defend against poisoning attacks. In recent years, several robust aggregation schemes have been proposed as a means of defense mechanisms against poisoning attacks to improve the robustness of FL systems [4]. Robust aggregation rules for model aggregation with multiple distributed clients ensure security by eliminating adversarial data and unnecessary model manipulations. However, recent studies show that the existing robust aggregation algorithms are not fully robust. In particular, they do not address the issue of SPoF or the malicious compromising of the global model [5] which demands new defense strategies.

The existing FL systems have several limitations that ultimately lead to a decentralized approach integrated with BC technology. First, FL relies on the assumption that the centralized aggregator is a trusted entity. However, this assumption is often not true in real-life applications hence, biased aggregators can compromise the learning performance [6]. Secondly, FL relies on a centralized aggregator, making the system vulnerable if the central server fails. The resilience of the overall FL network depends entirely on the robustness of the central aggregator [1]. Moreover, the existing FL design is vulnerable to malicious clients who may upload poisoned models to attack the FL network [4]. Therefore, most of the recent studies are carried out regarding the decentralized FL framework to address the above security concerns. Many solutions involve replacing the central server with a public or private blockchain to provide secure and immutable model storage.

Our Contribution:We propose a novel, robust Blockchain-based FL (BCFL) system comprised of two system models using secure off-chain computation for malicious mitigation. The off-chain computation replaces on-chain aggregation which consumes higher gas costs and computational power[7]. We evaluate the proposed BCFL system models against poisoning attacks and show that they can achieve the same expected accuracy level of the current robust FL systems while improving reliability and security. Finally, we

evaluate the cost-effectiveness of the proposed system models in terms of the total gas cost for the on-chain and off-chain executions.

The remainder of the paper is organized as follows: Section II presents the related work on robust FL and BCFL systems. Section III describes the novel system architecture. Section IV provides the systematic evaluation of the proposed two models under the novel BCFL architecture. Finally, in Section V, the conclusions are drawn with the future research directions.

II. Related Work

FL algorithms, which aim to mitigate the impact of malicious model updates on the central server while preserving the privacy of the FL system [4], [8]. The aggregation rules namely; Krum[9], Trimmed mean [5], and Median work as robust aggregation algorithms against Byzantine adversaries in FL. Nevertheless, the existing Byzantine-robust FL methods do not satisfy the expected robustness goals against poisoning attacks. The FooldsGold algorithm is a state-of-the-art defense against targeted poisoning attacks, which is proposed in [10]. FoolsGold addresses the shortcomings of the previous algorithms performs well with many adversaries, and does not rely on a specific number of adversaries[11].

Blockchain has gained significant attention for addressing issues like SPoF and poisoning attacks in FL systems due to its decentralized nature and reliability. Therefore, various BCFL systems have been proposed [6]-[12] by integrating BC technology. The BlockFL architecture that proposed in [12] facilitates decentralization and secure model storage. The BAFFLE system is proposed in [13], which utilizes smart contracts (SC) to aggregate local models on-chain. Similar to BlockFL[12], the BAFFLE system eliminates the risk of SPoF while achieving lower gas costs compared with the existing BCFL systems. Moreover, in [6], the BCFL system known as BLADEFL is designed to prevent the SPoF and identify malicious and lazy FL clients. However, these solutions involve on-chain aggregation, which relies on SCs to aggregate local models, resulting in higher computational power. Moreover, most of these systems are unable to differentiate between malicious gradients. A BCFL framework with Committee consensus (BFLC) is proposed in [14] to overcome the above challenges. This framework effectively reduces the amount of consensus computing and safeguards against malicious attacks. However, the selection criteria for the committee remain an unresolved issue. Based on the literature, BC has the potential to serve as an effective decentralized storage solution, but still, the existing BCFL systems have some unsolved issues [1] such as higher computation power and security threats. In TABLE I, a summary of feature comparison of the above-mentioned existing BCFL systems with the proposed system is included.

TABLE I: Feature comparison of related work of BCFL systems with Centralized Aggregation(CA) and Fully Decentralized (FD) BCFL models

Feature	[6]	[12]	[13]	[14]	[15]	CA-BCFL	FD-BCFL
Eliminates SPoF	✔	✔	✔	✔	✔	✔	✔
On-chain aggregation	✗	✗	✗	✔	✔	✗	✗
Detection as a service	✗	✗	✗	✔	✗	✔	✔
Efficiency and Scalability	✗	✗	✔	✔	✔	✔	✔

III. System Architecture

A. Novelty of the proposed BCFL system

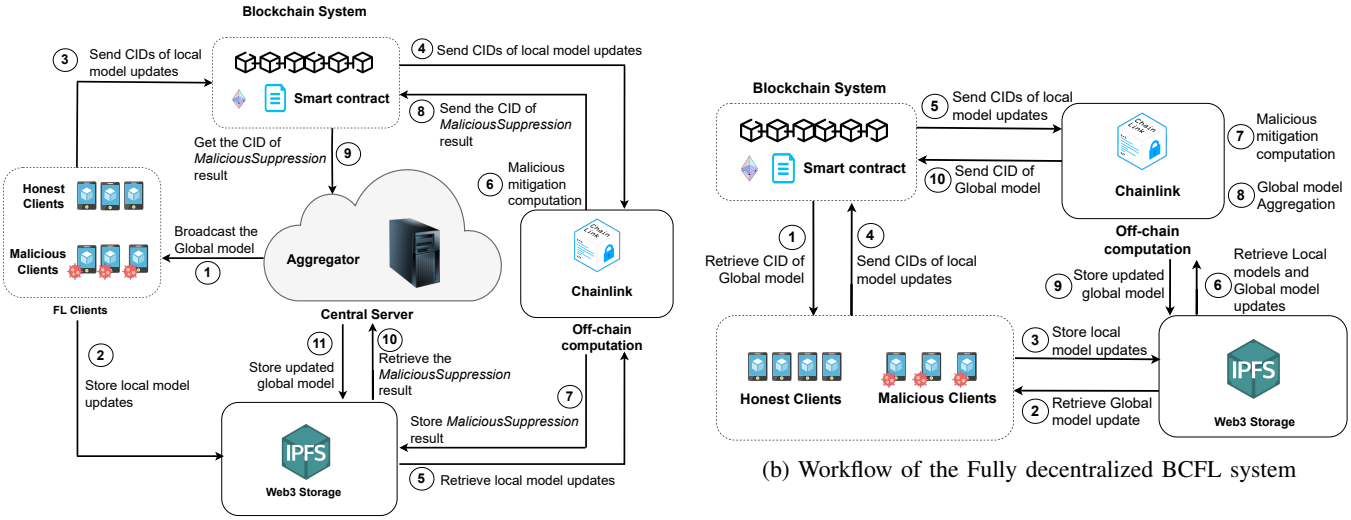
Most of the existing BCFL solutions [6], [16] aim to fully decentralize the aggregation process by replacing the central server with on-chain smart contracts. However, utilizing SCs for the on-chain aggregation of the local model updates and defense computations introduces substantial computational and communication burdens on the nodes within the BC network. In contrast, the first model with a central server proposed in our novel architecture is less reliant on the BC, unlike the on-chain aggregation schemes mentioned in previous works [13], [6]. It maintains a central server for global model aggregation with secure off-chain for complex computations, while BC is only used for secure decentralized storage of local model updates. This hybrid approach combines the cost-efficiency of centralized aggregation with the security of decentralized BC storage.

The second model of the proposed architecture is fully decentralized and similar to the existing on-chain aggregation schemes. However, the aggregation process and malicious mitigation computations are done on a secure off-chain computation platform, drastically reducing the on-chain computational cost and increasing scalability. Since the only existing feasible solution for the decentralized defense is private BCs, we have introduced a hybrid approach along with the off-chain computations.

Furthermore, most of the prior solutions [13], [15] have not effectively distinguished between malicious clients. They have focused on using BC, primarily to ensure accountability and avoid SPoF with the help of decentralized storage of local model updates. The proposed BCFL system also introduces a cost-effective off-chain defense computation scheme where we can implement any robust aggregation algorithm or malicious mitigation computation against poisoning attacks.

B. System Models

We propose two novel BCFL system models integrated with malicious mitigation off-chain computation. The first model, which is Centralized aggregation BCFL(CA-BCFL), maintains a central server for global model aggregation. Therefore, it is a partially decentralized system architecture, as shown in Fig. 1(a). The second system model shown in Fig. 1(b) is the Fully Decentralized BCFL (FD-BCFL) system, which aggregates on a decentralized off-chain platform. Two models share the same overall system architecture, with the key difference being the aggregation approach. We have



(a) Workflow of the Centralized aggregation BCFL system

(b) Workflow of the Fully decentralized BCFL system

Fig. 1: Overall system architecture and the workflow of the proposed BCFL system models

implemented two system models, that are partially and fully decentralized, to support the existing BCFL system types which are fully decentralized with SCs and BCFL systems which use central servers. In both cases, the malicious mitigation function is implemented in a secure, decentralized off-chain platform. Once the aggregation process is completed, the updated global model of each iteration is saved in BC. It increases reliability by providing an immutable record of each global model version and allowing recovery and restoration of the current global without retraining the models from scratch.

C. Proposed Novel Decentralized BCFL System Architecture

At a high level, the proposed system architecture for both models consists of the following key entities: the FL system, Off-chain secure storage, Blockchain network and Smart contracts, and Off-chain computations. Fig. 1 illustrates the workflow of CA-BCFL and FD-BCDL system models.

1) FL system: The system model consists of a decentralized set of clients with their own local models and a central server. The CA-BCFL is comprised of a central aggregator, which aggregates the global model with local model updates. Initially, the central server broadcasts the global model to the FL clients. The global model is iteratively updated through communication between the local FL clients and the central server. Since the FD-BCFL system model does not contain a central server, the initial global model should be initialized on the off-chain platform and then broadcast to the FL clients for the model training process.

2) Off-chain secure storage: BC served as a fully decentralized and secure storage system for the conventional FL system. The primary objective of integrating BC with FL is to protect the privacy of data owners and prevent malicious clients [1]. In the proposed architecture, we utilize the BC as a secure and decentralized storage for recording local model updates. However, considering the larger dimension of the local model weight vector, it is incompatible with saving the local model weight vectors directly using SCs. Therefore, we integrate a third-party off-chain file storage system like the Interplanetary file system (IPFS). IPFS is a distributed file storage facilitating private and permanent data storage. In previous studies, [17], IPFS is employed to store actual local models, while their corresponding hash values are stored in BC to ensure immutability. In this approach, CIDs are generated for corresponding data stored in IPFS, and these CIDs can be used to identify and access the stored local model updates for the required computations. This means that off-chain data can be validated by comparing the hash values of the on-chain records, preserving the integrity and efficiency of the FL process.

3) Smart Contracts: In each iteration, FL clients upload the CIDs of the corresponding local model updates to the SC while securely storing the actual model data off-chain. Once the aggregation is completed, the central server saves the CID of the global model in the SC to enhance the reliability of the system. The above functionality is performed by the consumer contract in Figure 2. In addition, the Chainlink node deploys its own Oracle SC.

4) Off-chain computation: In order to mitigate poisoning attacks, we perform an off-chain computation to filter out poisoned local model updates. SCs consume higher gas costs and have a limited capacity to connect to external APIs and perform complex computations on-chain. We utilize the Chainlink decentralized oracle network (DON) to overcome these limitations for complex off-chain computation. Chainlink allows SCs to access external APIs to fetch the necessary data, perform a decentralized off-chain computation, validate through DON nodes, and submit the *MaliciousSuppression* result, which only carries the benign local model updates on-chain for secure storage. The high-level architecture of the Chainlink off-chain computation process is shown in Fig. 2. The node operator is responsible for setting up and running the local Chainlink node.

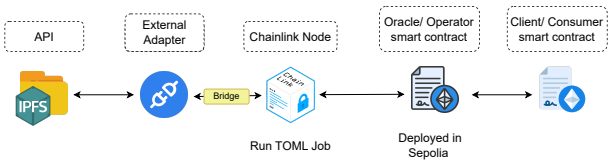


Fig. 2: Architecture of the Off-chain computation process using Chainlink

In addition, it deploys the Oracle contract and does the basic configurations of the external adapters. The node operator adds a job with specifications in a TOML file. After the job is added to the node, the node can be used to fulfill requests as a requirement. The consumer contract requests tasks from the node through the Oracle contract. The customized external adapter includes the off-chain computation and fetches the required data from the external APIs.

The Chainlink node regularly monitors the BC network for contract requests matching the job specifications, and when the requests match the job, they are picked up for processing. The node initiates the associated external adapter through the bridge. Chainlink nodes utilize DON to provide additional validation through consensus. The Oracle contract makes the same request to multiple Chainlink nodes in DON. Each node independently retrieves data and validates the results. This specifically eliminates the security threats of the off-chain computation process and allows for the implementation of a trusted BCFL system.

In this case, the consumer contract, which contains the client's requirement to fetch the actual local model updates of the corresponding CIDs, requests the Chainlink node through the Oracle smart contract, which is the operator contract of the node. The external adapter fetches the required data from IPFS, performs the required malicious mitigation computation, and returns the *MaliciousSuppression* result. The Chainlink node processes and validates the result of the external adapter. The oracle contract forwards the result to the consumer contract, and the consumer contract receives the result and saves it on-chain.

IV. Evaluation of the Proposed BCFL System

A. Experimental Setup

1) FL setting and Attack model: The FL system was set up with a TensorFlow backend and the Keras deep learning library. The number of clients involved is set to N = 10, and all the clients are selected in each iteration for the training. For the benchmarking, we have adopted a widely used dataset, MNIST, for the evaluation of the scheme. The training data is uniformly distributed among all clients. In order to train the dataset, a simple MLP model with three layers is used. The model parameters are shown in Table II. The hyperparameters of the optimizer FedSGD of the model are tuned to the following values: Learning rate = 0.1, Momentum = 0.9, Epochs = 5.

The attack model consists of a C number of malicious clients, where the total number of FL clients in the system is N. In the implementation of the attack model, the value C is

TABLE II: Summary of the Simple MLP model

Layer	Output Shape	Activation Function	Parameters
Input	784	-	0
Dense	50	ReLU	39250
Output	10	Softmax	510

controlled, and malicious clients perform targeted poisoning attacks, label flipping a fraction of the training dataset. The remaining honest clients do not perform label flipping on their dataset, thus providing accurate local model updates. In the simulation, the malicious clients only flip the labels from source class 1 to target class 7 as a (1→7) attack.

2) Off-chain computation setting: We use the Chainlink platform for the implementation of the secure off-chain computation. We run a local Chainlink node as the basic requirement to execute the off-chain computation with the use of an external adapter. The customized external adapter contains the off-chain computation logic for malicious mitigation and is hosted on a Google Cloud VM instance. A bridge is created by the node operator to add the external adapter to the node. A TOML Chainlink job is run to execute the external adapter. The bridge is included in the job as a fetch task. For the simulation, the Ethereum Sepolia testnet is utilized as the blockchain environment, and both the Chainlink node operator contract and the client consumer contract are deployed.

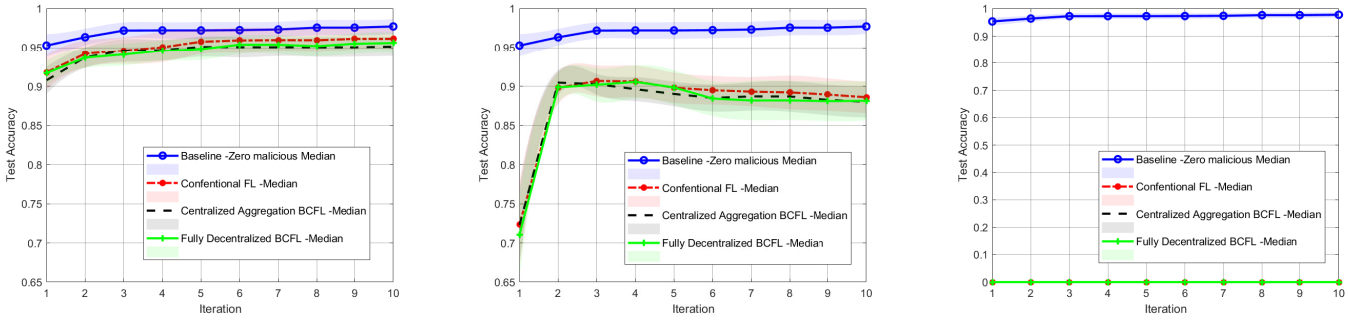
We use three evaluation metrics to determine the effectiveness of the proposed robust BCFL system model and compare the performance with baseline scenarios.

1) Test accuracy of the source class at the convergence.
2) Global accuracy at the convergence
3) Total Gas cost for on-chain and off-chain phases.

B. Experimental Results

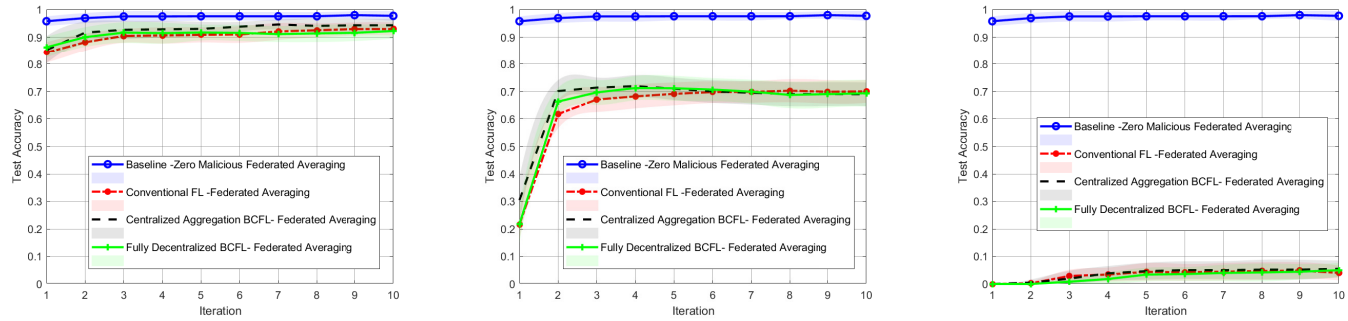
1) Test accuracy: First, we evaluate the resilience of the novel BCFL scheme against poisoning attacks using the test accuracy of the source class over ten iterations under a label-flipping attack. For the simulation results obtained in Fig.3, the BCFL models are integrated with the off-chain malicious mitigation algorithm, Median, to compute the *MaliciousSuppression* result for the aggregation. Then, we compare the test accuracy of the source class for two proposed models, CA-BCFL and FD-BCFL, with different proportions of malicious clients against the baseline scenario, which represents the optimal accuracy of a conventional FL system under zero malicious clients with the Median algorithm. All the results included in Fig. 3 are the average of 25 experiments.

In Fig. 3, the test accuracy curves of the CA-BCFL and FD-BCFL models approximately coincided with the accuracy curve of the Median algorithm implemented on the conventional FL system. Hence, both BCFL models achieve comparable performance to the Median robust aggregation method under varying proportions of malicious clients, where C = 2, 4, and 6. Therefore, the results outlined above show that integrating our novel decentralized defense system does



(a) Attack = 20% (b) Attack = 40% (c) Attack = 60%

Fig. 3: Test Accuracy of the source class under Label-flipping attack with Median



(a) Attack = 20% (b) Attack = 40% (c) Attack = 60%

Fig. 4: Test Accuracy of the source class under Label-flipping attack with Federated averaging

not adversely affect the normal performance of the current FL system. In addition to the accuracy preservation, the novel BCFL system also provides extra security enhancements of a decentralized defense approach. Both BCFL models save the *MaliciousSuppression*result and the current global model of each iteration on the BC so that they can be traced back in potential system failure. This enables traceability and auditability, achieving the goal of reliability by eliminating the issues caused by SPoF and malicious compromising of the central server of the conventional FL system.

We have experimented with the same procedure by integrating the Federated averaging as the off-chain computation and obtained similar results for the test accuracy, as shown in Fig. 4. Based on the results, it can be infer that, our decentralized defense approach is compatible with any off-chain computation that is integrated with different malicious mitigation defense algorithms. It successfully achieves the same expected accuracy level of the integrated defense algorithm while keeping the accuracy intact in a more trusted, accountable, reliable, and cost-effective environment compared with conventional FL and on-chain aggregation schemes.

To further evaluate the robustness of the two BCFL system models we increase the number of malicious clients as C = 2, 4, and 6 in the FL system of 10 clients. The test accuracy of the source class and the global accuracy of the FL system of the three schemes in the presence of different proportions of malicious clients at the convergence under label-flipping attack are included in Tables III and IV, respectively.

TABLE III: Test Accuracy of source class with Conv. FL, CA-BCFL and FD-BCFL system under different malicious percentages

Scheme	**Malicious Percentage**			
	20%	**40%**	**50%**	**60%**
Conventional FL- Median	0.93	0.71	0.28	0.00
CA-BCFL - Median	0.96	0.73	0.26	0.00
FD-BCFL - Median	0.94	0.73	0.29	0.00

TABLE IV: Global accuracy with Conv. FL, CA-BCFL and FD-BCFL system under different malicious percentages

Scheme	**Malicious Percentage**			
	20%	**40%**	**50%**	**60%**
Conventional FL- Median	0.90	0.87	0.83	0.79
CA-BCFL - Median	0.91	0.89	0.83	0.80
FD-BCFL - Median	0.92	0.89	0.83	0.80

The results in TABLE III show that the variation of the test accuracy values of the two system models of the proposed BCFL architecture for each malicious percentage remains the same as the conventional FL system with Median robust

TABLE V: The average transaction fee for the CA-BCFL and FD-BCFL systems

System Models	Transaction Fee (ETH)			
	Gas for initiating requests		Callback gas	
	Lower	Upper	Lower	Upper
CA-BCFL	$1.60e^{-9}$	0.00156	$0.70e^{-9}$	0.00068
FD-BCFL	$1.73e^{-9}$	0.00164	$0.76e^{-9}$	0.00070

aggregation scheme as Median fails to defend poisoning attacks after 50% of malicious percentage. Also, TABLE IV global accuracy values show that both BCFL models have comparable system performance for the Median robust defense, and decentralization of the proposed system model does not affect the existing FL performance.

2) Total gas cost: Finally, we analyze the cost-effectiveness of the CA-BCFL and FD-BCFL models comparatively using the total gas cost for the on-chain and off-chain processes. In the on-chain phase, uploading data to BC and fulfilling requests consumes gas. We use the Sepolia test network to experiment and test the gas cost. In addition, LINKs are used to pay node operators for retrieving data and providing computation services in Chainlink. LINK costs are generally far lower than Ethereum gas fees for the equivalent computations. As shown in TABLE V, we calculate the transaction fee for each mode, considering the total gas cost, which comprises:

1) Gas used for initiating requests by consumer contract to the Chainlink node to trigger off-chain computations.
2) Callback gas for the Oracle contract

The gas price (Gwei) fluctuates depending on network conditions. To analyze the transaction fees, we calculated the average gwei values of 50 transactions over a period of time for each model. Considering the fluctuation of the gwei values based on network congestion, we include both the lower and upper ranges. Considering both upper and lower ranges, the total transaction fee is calculated as:

$$\text{Transaction fee} = \text{Gas cost} \times \text{Gas price (Gwei)}$$

The consumer SC has a fixed gas cost of 135577, while for the oracle SC, the fixed gas cost is 59837 for the transactions. The node operator is paid in LINKs for fulfilling requests, and the average value is around 0.1 LINK. Therefore, the gas cost for computations done off-chain that require only minimal on-chain coordination is much lower compared to the gas cost for the equivalent computation done on-chain using SCs.

V. Conclusions

The paper proposes a novel, robust BCFL architecture with flexible off-chain computation for malicious mitigation. We introduce two BCFL system models under a similar architecture: central aggregation with off-chain malicious mitigation and fully decentralized robust aggregation. Both models maintain a comparable performance to the conventional FL system in terms of client malicious mitigation, where the fully decentralized model eliminates the SPoF due to its continuous validations of the off-chain computations. As the experimental results show both BCFL models will be a reliable and immutable alternative to on-chain computations with the benefit of low-cost off-chain computation. In future works, the focus is to reduce the latency of the system associated with BC networks.

Acknowledgment

This work has been partly supported by European Union under CONFIDENTIAL-6G (Grant No: 101096435), and Science Foundation Ireland under CONNECT phase 2 (Grant no. 13/RC/2077_P2) projects.

References

[1] A. Qammar, A. Karim, H. Ning, and J. Ding, "Securing federated learning with blockchain: a systematic literature review," *Artificial Intelligence Review*, vol. 56, no. 5, pp. 3951–3985, 2023.

[2] V. Tolpegin, S. Truex, M. Gursoy, and L. Liu, "Data poisoning attacks against federated learning systems," 07 2020.

[3] Z. Wang and Q. Hu, "Blockchain-based federated learning: A comprehensive survey," *arXiv preprint arXiv:2110.02182*, 2021.

[4] S. Li, E. C.-H. Ngai, and T. Voigt, "An experimental study of byzantine-robust aggregation schemes in federated learning," *IEEE Transactions on Big Data*, 2023.

[5] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5650–5659.

[6] C. Ma, J. Li, L. Shi, M. Ding, T. Wang, Z. Han, and H. V. Poor, "When federated learning meets blockchain: A new distributed learning paradigm," *IEEE Computational Intelligence Magazine*, vol. 17, no. 3, pp. 26–33, 2022.

[7] Y. Miao, Z. Liu, H. Li, K.-K. R. Choo, and R. H. Deng, "Privacy-preserving byzantine-robust federated learning via blockchain systems," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 2848–2861, 2022.

[8] V. Shejwalkar and A. Houmansadr, "Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning," in *NDSS*, 2021.

[9] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Advances in neural information processing systems*, vol. 30, 2017.

[10] C. Fung, C. J. Yoon, and I. Beschastnikh, "The Limitations of Federated Learning in Sybil Settings," in *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*, 2020.

[11] Y. Siriwardhana, P. Porambage, M. Liyanage, and M. Ylianttila, "Robust and resilient federated learning for securing future networks," in *2022 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*. IEEE, 2022, pp. 351–356.

[12] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Blockchained on-device federated learning," *IEEE Communications Letters*, vol. 24, no. 6, pp. 1279–1283, 2019.

[13] P. Ramanan and K. Nakayama, "Baffle: Blockchain based aggregator free federated learning," in *2020 IEEE international conference on blockchain (Blockchain)*. IEEE, 2020, pp. 72–81.

[14] Y. Li, C. Chen, N. Liu, H. Huang, Z. Zheng, and Q. Yan, "A blockchain-based decentralized federated learning framework with committee consensus," *IEEE Network*, vol. 35, no. 1, pp. 234–241, 2020.

[15] S. K. Lo, Y. Liu, Q. Lu, C. Wang, X. Xu, H.-Y. Paik, and L. Zhu, "Blockchain-based trustworthy federated learning architecture," *arXiv preprint arXiv:2108.06912*, 2021.

[16] Y. Miao, Z. Liu, H. Li, K.-K. R. Choo, and R. H. Deng, "Privacy-preserving byzantine-robust federated learning via blockchain systems," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 2848–2861, 2022.

[17] S. Kumar, S. Dutta, S. Chatturvedi, and M. Bhatia, "Strategies for enhancing training and privacy in blockchain enabled federated learning," in *2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM)*. IEEE, 2020, pp. 333–340.