BlockFL: A Blockchain-enabled Federated Learning System for Securing IoVs

Jawad Rahman, Charles Roeder, Oscar De La Cruz, Tyler Baudier, and Wenjia Li*
Department of Computer Science
New York Institute of Technology
New York, NY 10023, USA
{jrahma04, croeder, odelacru, tbaudier, wli20}@nyit.edu

***Abstract*— The rapid evolution of Internet of Vehicles (IoVs) technologies has ushered in an era of connected transportation, which has enabled us to collect and analyze data at an unprecedented scale. However, the vast amount of data generated by IoVs poses significant privacy and security challenges, as they may be susceptible to leakage and manipulation by malicious attackers. In this work, we explore the integration of emerging blockchain technology with federated learning to create a secure and decentralized framework for IoV data management and analysis. The proposed system leverages the immutable and transparent nature of blockchain to ensure data integrity and trust among IoV nodes, while federated learning facilitates collaborative machine learning without compromising individual data privacy. Through a combination of cryptographic techniques and consensus mechanisms, the blockchain-enabled federated learning system aims to thwart adversarial attacks, ensure secure data aggregation, and enhance the overall resilience of IoV networks. This study presents a comprehensive architecture, evaluates its performance through simulations, and demonstrates its potential in mitigating security risks in IoV environments. The findings highlight the feasibility and effectiveness of this approach, paving the way for more robust and secure IoV systems.**

***Keywords*— Security, Blockchain, Federated Learning, IoV**

I. Introduction

In recent years, Internet of Things (IoT) has exploded in popularity, not just in the consumer market but other niches like vehicular networks, such as the Internet of Vehicles (IoV), and industrial applications, such as the Industrial Internet of Things (IIoT) [1]. The Internet of Vehicles (IoV) is transforming the transportation sector by facilitating seamless communication and data exchange among vehicles, infrastructure, and other networked devices. Key applications include real-time traffic monitoring, accident response systems, and dynamic routing. However, the extensive connectivity and data sharing in IoV networks also introduced significant security and trust challenges, which may include potential data breaches, unauthorized access, and cyberattacks that can compromise both personal and vehicular safety.

Traditional centralized data processing models are increasingly inadequate for addressing these security and trust issues in IoVs due to their vulnerability to single points of failure, scalability issues, and the heightened risk of large-scale data breaches. In contrast, Federated learning (FL) emerges as a promising decentralized approach to machine learning that keeps data localized on individual devices, thereby enhancing privacy and reducing the risk of data exposure [2]. Despite its advantages, federated learning faces several challenges, including the secure aggregation of model updates, maintaining the trustworthiness of participating nodes, and defending against adversarial attacks. For example, malicious nodes might attempt to corrupt the global model through data poisoning attacks by sending inverted model parameters [3]. They could also seek to gain disproportionate control over the global model through model replacement attacks [3]. Therefore, it is essential to implement additional security measures.

Since Bitcoin's inception in 2009, blockchain technology has attracted a lot of attention mainly because of its decentralized, transparent, and immutable nature [4]. Blockchain technology provides a robust and effective solution to enhance federated learning. By leveraging blockchain, a secure and tamper-proof system can be established to manage and verify the contributions of each node in the federated learning process. This integration ensures data integrity, fosters trust among participants, and significantly bolsters the overall security of the IoV network.

In this research, we aim to design, implement, and evaluate a blockchain-enabled federated learning system named BlockFL, which is specifically tailored to address the security and privacy needs of IoVs. By combining the strengths of blockchain and federated learning, this project seeks to develop a resilient framework that can securely handle the vast amounts of data generated by IoVs, protect against cyber threats, and maintain user privacy.

The main contributions of this work are listed as follows.

i. We proposed and implemented a novel system combining federated learning and blockchain to enhance the security of the IoVs.
ii. We increase the level of decentralization in federated learning while also maintaining sufficient coordination and validation through the application of a blockchain design.
iii. To compare how well models generated by federated learning perform with an increasing number of clients, as well as to models developed via a traditional, centralized system, we conduct a series of experiments, the results of which indicate federated learning models are on par with non-federated counterparts.

* Corresponding author: Dr. Wenjia Li (E-mail: wli20@nyit.edu)

The remainder of this article is organized as follows. Section II covers some existing works that are related to this research. Section III details the design of our proposed BlockFL system. In Section IV, we present and analyze the results from a variety of experiments of our system and compare it to a baseline model. Finally, Section V discusses some of the limitations and possible future directions of our study and concludes this paper.

II. Related Works

In recent years, there have been numerous research efforts which aim at enhancing the security of various IoT systems.

Zhang et al. [5] attempt to enhance the trust and security of the Internet of Things being uploaded to the cloud by implementing a blockchain with an alliance chain, which is used to support the central authority by allowing other nodes to assist the central authority in making partial private keys. This was done to reduce the bottleneck around only having one central authority and the security issues with one node holding all the private keys of all the users in the blockchain. They use anonymity when interacting with the blockchain to protect user privacy and found that their scheme outperformed the standard means for cloud-based sharing of IoT data such as attribute-based encryption.

In [6], a trust management system based on blockchain is used to help manage the high level of complexity, volume, and mobility issues when using the Internet of Vehicles network. The Internet of Vehicles network is a network of vehicles with sensors that can share information between vehicles. The authors use Bayesian Inference to generate a "judgment", or a summary of all messages a query vehicle has received. By using this, messages deemed to have low credibility will have less influence on the judgment generated, resulting in a more accurate judgment. Lastly, they use a combined consensus mechanism so that vehicles with the largest change in trust have priority when updating the blockchain.

Li and Song studied an attack-resistant trust management scheme (ART) for Vehicular ad hoc networks (VANETs) [7], which is able to detect and cope with malicious attacks and also evaluate the trustworthiness of both data and mobile nodes in VANETs. More specifically, the trustworthiness of a node is evaluated in two separate dimensions, namely functional trust and recommendation trust, which indicate how likely a node can fulfill its functionality and how trustworthy the recommendations from a node for other nodes will be.

In [8], an AI-enabled trust management system (AIT) was proposed for vehicular networks using the blockchain technology. In the AIT system, each vehicle senses, generates, and exchanges traffic related messages with other vehicles, and these messages will then get validated by the neighboring vehicles. As vehicles process messages from nearby vehicles, they will establish and update the trust relationship of those vehicles, which is enabled by utilizing the deep learning algorithm. Once a vehicle identifies untrustworthy vehicles, it reports them to the nearby roadside unit (RSU), and the RSU will validate the authenticity of the report as well as the identity of the vehicle by using the blockchain technology.

In [9], a protocol called SAFELearning is designed to mitigate data poisoning attacks during the aggregation phase of federated learning. This technique involves randomly partitioning participating nodes into one-time subgroups, ensuring that nodes within a subgroup are unaware of the other members in that group. The parameters from each subgroup are then recursively aggregated into the global model. Partial Parameter Disclosure (PPD) is employed to detect anomalies within the aggregated parameters of a subgroup, thereby identifying instances of poisoning attacks.

Vucovich et al. also address data poisoning attacks using Bayesian statistics in their method called FedBayes [10]. This approach calculates the mean and standard deviation for parameters in each layer of the global model, creating a normal distribution from these values. The probability of each client's parameters is then calculated using this distribution to assess their influence on the global model. Clients whose parameters show significant deviations from the previous global model are flagged as potentially malicious, and their influence is reduced during the aggregation process.

In [11], compressed sensing (CS) is explored as a method for more efficient and secure aggregation, particularly for environment sensing capability (ESC) nodes. A CS-based spreading algorithm is employed by all nodes to transmit information, resulting in a superimposed signal that obscures the data from individual nodes, thus protecting their privacy against eavesdropping attacks. However, while this approach enhances privacy, it does not adequately address poisoning attacks or the secure storage of model parameters.

III. Overview of the BlockFL System

One of the main novelties of our system lies within the seamless connection between federated learning and blockchain. Our system's overall workflow is shown in Fig. 1.

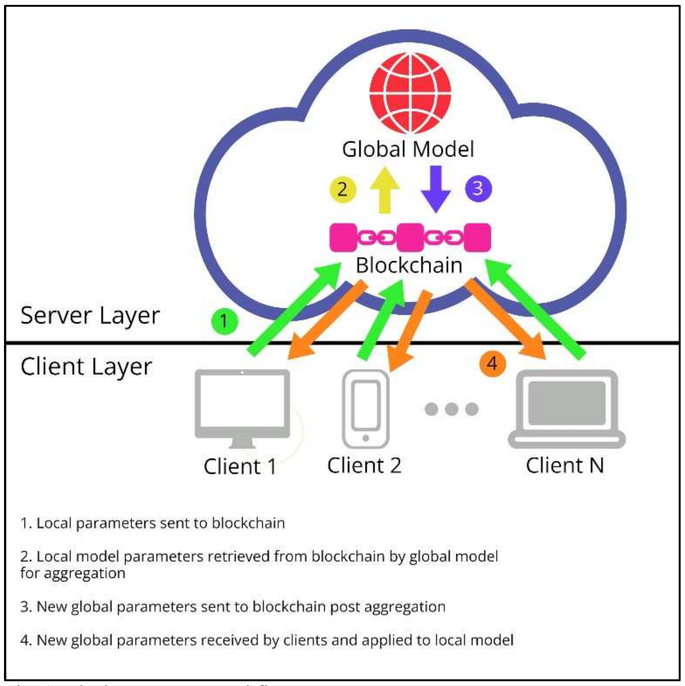
Global Model
2
3
Blockchain
Server Layer
1
Client Layer
4
Client 1
Client 2
Client N
1. Local parameters sent to blockchain
2. Local model parameters retrieved from blockchain by global model for aggregation
3. New global parameters sent to blockchain post aggregation
4. New global parameters received by clients and applied to local model

Fig. 1. BlockFL System workflow

There are two general layers – the local/client layer and the server layer. These two layers correspond to the general federated learning framework, where there are multiple local models that each exist on devices at the source of the data, and a central federated server that collects these local models and aggregates them into a global model. In this work, the local models are located at the client layer and the global model is located at the server layer. The server layer itself is made up of two different modules, which are responsible for managing the global model and blockchain, respectively. Unlike traditional federated learning, where the federated server directly communicates with client devices to collect local models, our system leverages blockchain to store models from both clients and the global server. This approach enhances the security and trustworthiness of the entire process.

Federated learning is typically a highly coordinated process, and our system is designed with several coherent steps to produce accurate and functional global models. Before training begins, each client and the global model are assigned unique identifiers. In the client layer, local models are trained on local data, and each client sends its model parameters to the blockchain, including its identifier in the transaction. These identifiers enable verification of the model parameters, allowing the originator of specific parameters to be traced back to a particular node and ensuring that the node is authorized to contribute. Once these parameters are recorded in a block, the global model retrieves them from the blockchain. The local parameters are then aggregated into a single set, forming the new global model. This updated global model, along with its identifier, is sent to the blockchain. This process ensures that parameters from both local clients and the global model can be verified. After the global model parameters are mined as a block, clients retrieve and apply them to their local models, replacing the local parameters with the global ones. As the global model is an aggregate of multiple local models, it is generally more accurate and better suited to a wider range of data, making it the preferred model for all clients.

A. Blockchain Design

In our system, the type of blockchain that we use is a private blockchain. A private blockchain still uses the idea of distributed ledgers, however, the participants that are allowed on the network can be controlled by some party, typically the administrators. This allows for the greater coordination that federated learning requires. For example, local devices can be verified before being allowed to participate in training, something common in most federated learning schemes. Yet, the decentralized nature of blockchain is still somewhat preserved since nodes within the network each retain a copy of the ledger and cryptographic means are employed to avoid the tampering of this ledger, with multiple nodes still participating in validation and consensus.

Our blockchain uses the Proof-of-Work (PoW) consensus protocol to reach an agreement and resolve conflicts about the ledger's state. In this mechanism, miners compete to figure out the hash value of the next block, which involves determining a nonce value that when hashed with the data of the block, meets a certain PoW threshold. This threshold is usually defined as a certain number of zeros preceding the eventual final hash. The process of finding the hash of the next block is illustrated in Fig. 2 [12]. Upon finding a correct solution, other nodes in the network verify the solution by applying the hash function with the nonce provided, and if verified, the miner who determined the nonce obtains the right to append the block to the chain along with a reward [13]. Determining a correct nonce and hash value is computationally expensive, so it makes it unlikely that any one node can control the whole blockchain and maliciously alter past transactions.

In our blockchain, if a chain starts to diverge, meaning multiple different versions of the same chain start to emerge, the longest chain that has been validated is chosen as the correct chain, and all nodes replace their chain with this. Validation occurs in a similar way to how new blocks are validated before being added to the chain, with hashes being applied to check the validity of block data and nonces.

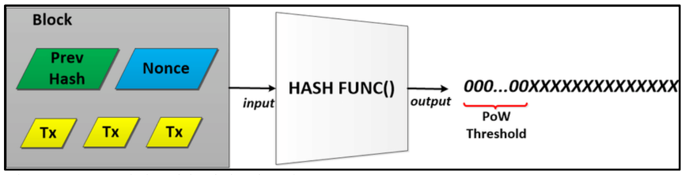

Fig. 2. Determining block hash using PoW

Our use of blockchain is not only designed to ensure the security and integrity of model parameters, it is also done to provide an archival history of local models and global models, having the added benefit of auditability. Previous versions of a global model can be accessed should there be a need, such as if the latest version's performance is lacking and a previous version must be used instead. The parameters of local models are also recorded, so there is potential to verify whether a client is truly participating in local learning or sending fabricated model parameters to blockchain, adding another layer of verification to the entire process.

B. The Details of the Deep Learning Algorithm

The deep learning algorithm that we use in this work is the Convolution Neural Network (CNN). This algorithm is used to perform classification tasks, usually image recognition. Because they handle multidimensional data, CNNs are composed of multiple layers which are, more generally, the convolution, pooling, and fully connected layers. The input data first goes through the convolution layers where filters are applied to it to create an activation map. This map is fed to the pooling layer, which reduces the spatial dimensions of this activation map size, again, by applying filters, this time to condense regions of the activation map based on summary statistics. Finally, in the fully connected layer, the flattened data from the pooling layer is processed using a fully connected neural network, with multiple layers of connected neurons mapping the relationship between input and output [14]. A diagrammatic representation of this algorithm is shown in Fig. 3 [15].

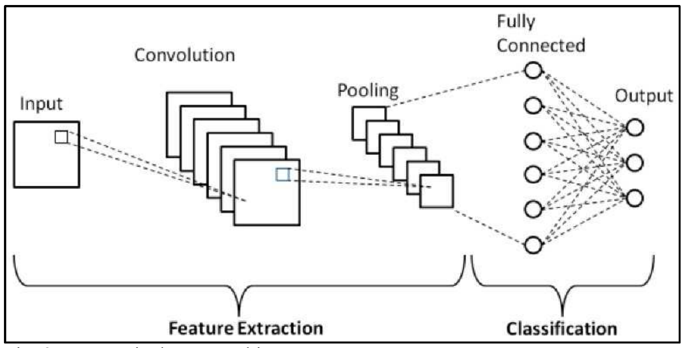

Fig. 3. Typical CNN architecture

The dataset we use in our experiments is the Modified National Institute of Standards and Technology (MNIST) dataset [16], which is composed of handwritten digits from 0 to 9. A total of 49,800 samples were used for training, 10,200 for validation, and 10,000 for testing. Being images, we used a 2-Dimensional CNN model to process the data.

C. Attack Model and Mitigation

We have taken the following cyber attacks into consideration when designing and developing our proposed BlockFL system.

1. Data poisoning attacks, where malicious nodes inject false data to corrupt the global model [3], are mitigated through the immutable audit trail provided by the blockchain, enabling collective verification and rejection of anomalies.
2. Model replacement attacks, where a node attempts to control the global model [3], are countered by the decentralized ledger and Proof-of-Work consensus mechanism, making it impractical for any single node to manipulate the model.
3. Eavesdropping on the communication channels [17], which is generally common in wireless networks, are addressed by keeping data on local devices and only transmitting model parameters, with the blockchain providing secure, tamper-proof storage and controlled access, thus enhancing security and privacy protection compared to the traditional centralized systems.

IV. Experimental Study

A. Experimental Setup

The Python PyTorch library was used to implement CNNs in our experiment. Client nodes and the server's Global Model were both run on the Google Colab, and they were run on a dedicated VM in Google Compute Engine (GCE). The MNIST training data was divided by the number of total clients, and each client had some portion of the data loaded on to it. Each client was also assigned a unique ID.

We constructed a Python class to implement our blockchain mechanisms, specifically using the hashlib library to implement SHA-256 hashes for block creation, validation, and conflict resolution using PoW. To allow the other components in our system to interface with the blockchain over network, we used Flask to develop an API to handle HTTP requests to the blockchain. Requests were formatted using JSON, which meant to ensure compatibility with the format, tensors containing model parameters needed to be converted into byte strings before being sent as transactions to the blockchain in the global model and local client's code.

Because the global model and blockchain exist on the same layer – the server layer – the global model is allowed to know more details about the blockchain than local clients, and gets to requests when a block should be mined upon certain conditions being met, similar to the way smart contracts work in Ethereum [18].

The code for our blockchain and API was hosted on Replit, which not only assigns a dedicated VM for it, but also a URL where HTTP requests can be made to the API. This allowed us to keep our API running live, and as a separate module/service from the global model, something that was necessary as model training and blockchain hosting cannot be done simultaneously in the same runtime environment.

B. Experimental Results

We ran six experiments with a different number of clients each time: a 2-, 4-, 8-, 16-, 32-, and 64-client test. For each experiment, 20 rounds of training were performed. From each experiment, we collected performance metrics of the aggregated global model, specifically its accuracy, loss, precision, recall, and F1 score during the validation. We also collected data about the communication overhead generated in each experiment, and the global model's final performance on the test data at the end of the 20 rounds. For comparison purposes, we ran 20 rounds of training on a centralized CNN model using no federated learning as our baseline for comparison, helping us see the differences in training and performance between the regular machine learning and federated learning.

Fig. 4 and 5 show the performance of the aggregated global model for each test compared with the baseline model. Table 1 shows each final model's performance on the test data. Fig. 6 shows the communication overhead generated during each of our tests.

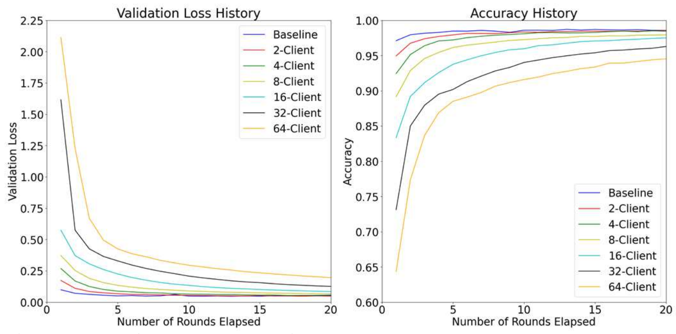

Fig. 4. Loss and accuracy comparison

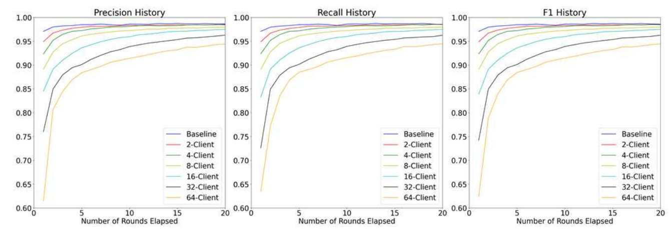

Fig. 5. Precision, recall, and F1 score comparison

TABLE I. FINAL MODEL PERFORMANCES ON TEST DATA

Model	Performance Metrics				
	Accuracy	***Loss***	***Precision***	***Recall***	***F1 Score***
2-Client	0.9874	0.0372	0.9871	0.9878	0.9875
4-Client	0.9873	0.0392	0.9873	0.9878	0.9875
8-Client	0.9832	0.0536	0.9828	0.9834	0.9831
16-Client	0.9771	0.0742	0.9769	0.977	0.977
32-Client	0.9656	0.1165	0.9661	0.9659	0.966
64-Client	0.9505	0.1853	0.951	0.9503	0.9506
Baseline	0.9875	0.0441	0.9875	0.9878	0.9877

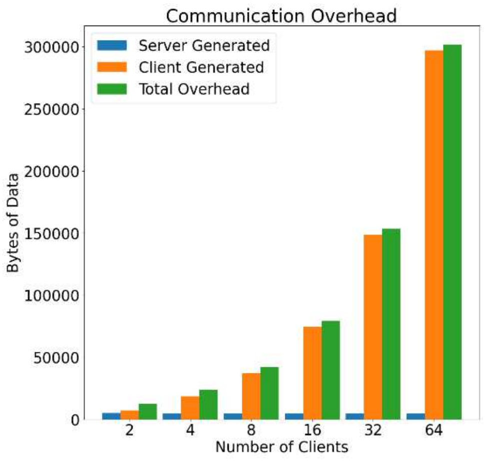

Fig. 6. Communication overhead for each federated learning model

These results show that even with an increasing number of clients, the federated learning model in our system still performs well. For all experiments, the global model was able to achieve a validation loss of 0.25 or less within 20 rounds of training, albeit slower than what it took the baseline to achieve such loss, with the 64-Client test's final model having the largest loss during testing. These numbers are all generally comparable to the performance of the baseline model, with the 2-Client and 4-Client models even slightly outperforming the baseline in terms of loss. It may take more rounds of training to reduce the loss as the number of clients increases, but this is still within a reasonable number of rounds. We also noticed, rather interestingly, that as the number of clients increases, the smoothness of all the curves also increases up to a certain threshold. Going from the baseline to the 8-client curves, the baseline model produced the "bumpiest" curve, however, beyond 8 clients, the curves start being bumpy again. This indicates that our system may be less susceptible to noise than a non-federated-learning-based system since by training and averaging multiple models, faults that a single model may have in its parameters will become less dominant. However, once data begins to be spread too thinly among the clients, noise may be reintroduced.

Finally, the communication overheads increased as the number of clients increased. Looking at the breakdown of communication overhead, while the overhead generated by the server remained almost constant for each test, the client generated overhead increased dramatically, which in turn increased the total overhead for that test. In the real world where data may not be in the same place before training, there would still be communication overhead in transmitting data to the central server, which may be increased by the need for secure protocols. Even if our entire dataset was transmitted using the best lossless compression technique, which would achieve a compression ratio of 3:1 [19], it would still be 18 MB altogether, so we are assuming this as the minimum communication for our baseline. Therefore, we believe that the communication overhead of our federated learning models would still be better than the overhead incurred by transmitting the raw data instead.

V. CONCLUSION AND FUTURE DIRECTIONS

In this work, we have compared our BlockFL system to a typical non-federated-learning-based system with respect to the model performance and the communication overhead. However, when it comes to security, we made the assumption that the inherent mechanisms of blockchain would ensure security of the data in our network, and that being a private blockchain, the clients and server would be verified to participate in the learning process beforehand. Yet, even with the use of federated learning and blockchain, security would still possibly remain an issue, and the proper measures must be taken to address it. Additionally, we have also made the assumption that mechanisms will be in place to enforce the proper coordination of server and clients which is required by the federated learning. While we manually coordinated some of the server and client activities in our work, further procedures are required to ensure the server and clients coordinate as intended in this process.

To implement more robust security measure, some system that maintains a database of registered clients and asks clients to validate themselves before sending a transaction, whether that is a password or some other private key, can be used to prevent unwanted parties from sending the model transactions through the network. The server could also check the list of pending transactions and remove any that originates from the unwanted parties. In either case, however, this has implications on communication and computational overhead, as well as the decentralized nature of the blockchain. Because the global model has many privileges like checking the blockchain's pending transactions and being able to decide when to mine a block, there must be extra measures taken to ensure only the global model can perform these actions, including validating that such requests originate from the global model in the first place. As for the future direction, we would like to integrate the aforementioned attacks into our experiments, and check how they affect the performance of the proposed approach. We would also like to experiment with a larger amount of client nodes to test the scalability of our system in schemes like IoVs. Thus, these issues remain open to future research.

In summary, we proposed the BlockFL system, which integrates blockchain into federated learning. In this system, the global model and client essentially communicate their model parameters using the blockchain. The blockchain is used to store model parameters from both the global and local models during the coordinated federated learning process, providing a secure means of storage on top of the data privacy offered by not moving data from its source device. Additionally, it provides a means of archiving model parameters for purposes like reverting model versions or tracking client participation. Such an integration overall enhances the privacy of participating parties in a decentralized, transparent manner. To demonstrate the capabilities of the system, we implemented a prototyped system and conducted experiments comparing the performance of its ML model at differing numbers of clients, as well as to a non-federated, baseline model. The results show that our system is able to maintain a solid performance even with an increasing number of participating client nodes, and performs comparably to a non-federated model, even overperforming the baseline in some instances.

Acknowledgment

We would like to thank Kaggle user Puru98 [20] for making the PyTorch implementation of federated learning public, which we adapted for this research. We would also like to thank software engineer Bimo Putro Tristianto for his tutorial on creating a blockchain with Python [21], which we used to implement a blockchain for this work.

References

[1] K. D. Foote, “A brief history of the internet of things - DATAVERSITY,” DATAVERSITY, Mar. 28, 2023. https://www.dataversity.net/brief-history-internet-things/

[2] L. U. Khan, W. Saad, Z. Han, E. Hossain and C. S. Hong, "Federated Learning for Internet of Things: Recent Advances, Taxonomy, and Open Challenges," in *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1759-1799, 2021, doi: 10.1109/COMST.2021.3090430.

[3] A. Panda, S. Mahloujifar, A. N. Bhagoji, S. Chakraborty, and P. Mittal, “SparseFed: Mitigating Model Poisoning Attacks in Federated Learning with Sparsification,” arXiv.org, Dec. 12, 2021. http://arxiv.org/abs/2112.06274

[4] V. Gupta, “A brief history of blockchain,” Harvard Business Review, Mar. 29, 2024. https://hbr.org/2017/02/a-brief-history-of-blockchain

[5] L. Zhang, X. Li, Q. Wu, and F. Rezaeibagha, “Blockchain-Aided anonymous traceable and revocable access control scheme with dynamic policy updating for the cloud IoT,” in *IEEE Internet of Things Journal*, vol. 11, no. 1, pp. 526–542, Jan. 2024, doi: 10.1109/jiot.2023.3287190.

[6] H. Zhang, J. Liu, H. Zhao, P. Wang and N. Kato, "Blockchain-Based Trust Management for Internet of Vehicles," in *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 3, pp. 1397-1409, 1 July-Sept. 2021, doi: 10.1109/TETC.2020.3033532.

[7] W. Li and H. Song, "ART: An Attack-Resistant Trust Management Scheme for Securing Vehicular Ad Hoc Networks," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 960-969, April 2016, doi: 10.1109/TITS.2015.2494017..

[8] C. Zhang, W. Li, Y. Luo and Y. Hu, "AIT: An AI-Enabled Trust Management System for Vehicular Networks Using Blockchain Technology," in *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3157-3169, March 1, 2021, doi: 10.1109/JIOT.2020.3044296..

[9] Z. Zhang, J. Li, S. Yu, and C. Makaya, “SAFELearning: Secure aggregation in federated learning with backdoor detectability,” IEEE Transactions on Information Forensics and Security, vol. 18, pp. 3289–3304, Jan. 2023, doi: 10.1109/tifs.2023.3280032.

[10] M. Vucovich, D. Quinn, K. Choi, C. Redino, A. Rahman and E. Bowen, "FedBayes: A Zero-Trust Federated Learning Aggregation to Defend Against Adversarial Attacks," 2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2024, pp. 0028-0035, doi: 10.1109/CCWC60891.2024.10427896.

[11] W. Li, G. Chen, X. Zhang, N. Wang, D. Ouyang, and C. Chen, “Efficient and secure aggregation Framework for Federated Learning-Based spectrum Sharing,” in *IEEE Internet of Things Journal*, p. 1, Jan. 2024, doi: 10.1109/jiot.2024.3357575.

[12] A. Panarello, N. Tapas, G. Merlino, F. Longo, and A. Puliafito, “Blockchain and IoT Integration: A Systematic survey,” *Sensors*, vol. 18, no. 8, p. 2575, Aug. 2018, doi: 10.3390/s18082575.

[13] Y. Yun, “Proof of work: what is it, and how does it figure into bitcoin halving?,” Forkast, Jan. 21, 2022. [Online]. Available: https://forkast.news/proof-of-work-what-is-it-bitcoin-halving/

[14] M. Mishra, “Convolutional neural networks, explained - towards data science,” Medium, Dec. 15, 2021. [Online]. Available: https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939

[15] V. H. Phung and E. J. Rhee, “A High-Accuracy Model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets,” *Applied Sciences*, vol. 9, no. 21, p. 4500, Oct. 2019, doi: 10.3390/app9214500.

[16] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges, “The MNIST Dataset”. [Online]. Available: http://yann.lecun.com/exdb/mnist/, last visited May 27, 2024.

[17] K. N. Kumar, C. K. Mohan, and L. R. Cenkeramaddi, “The impact of adversarial attacks on federated Learning: a survey,” in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–20, Jan. 2024, doi: 10.1109/tpami.2023.3322785.

[18] I. Berman et al., “Trustable environmental monitoring by means of sensors networks on swarming autonomous marine vessels and distributed ledger technology,” *Frontiers in Robotics and AI*, vol. 7, May 2020, doi: 10.3389/frobt.2020.00070.

[19] Handbook of Image and Video Processing. 2005. doi: 10.1016/b978-0-12-119792-6.x5062.

[20] Puru98, “ Federated Learning (PyTorch)”, https://www.kaggle.com/code/puru98/federated-learning-pytorch, last visited August 7, 2024.

[21] Bimo Putro Tristianto, “Build your own blockchain in Python: a practical guide”, https://bimoputro.medium.com/build-your-own-blockchain-in-python-a-practical-guide-f9620327ed03, last visited on August 7, 2024.