Toward Trustworthy AI: Blockchain-Based Architecture Design for Accountability and Fairness of Federated Learning Systems

Sin Kit Lo, Yue Liu, Qinghua Lu, *Senior Member, IEEE*, Chen Wang, Xiwei Xu, Hye-Young Paik, *Member, IEEE*, and Liming Zhu

***Abstract*—Federated learning is an emerging privacy-preserving AI technique where clients (i.e., organizations or devices) train models locally and formulate a global model based on the local model updates without transferring local data externally. However, federated learning systems struggle to achieve trustworthiness and embody responsible AI principles. In particular, federated learning systems face accountability and fairness challenges due to multistakeholder involvement and heterogeneity in client data distribution. To enhance the accountability and fairness of federated learning systems, we present a blockchain-based trustworthy federated learning architecture. We first design a smart contract-based data-model provenance registry to enable accountability. Additionally, we propose a weighted fair data sampler algorithm to enhance fairness in training data. We evaluate the proposed approach using a COVID-19 X-ray detection use case. The evaluation results show that the approach is feasible to enable accountability and improve fairness. The proposed algorithm can achieve better performance than the default federated learning setting in terms of the model's generalization and accuracy.**

***Index Terms*—Accountability, AI, blockchain, fairness, federated learning, machine learning, responsible AI, smart contract.**

I. Introduction

The exponential growth of data, owing to the wide usage of smart devices has fueled the extensive application of the AI technology [1]. The ground-breaking advances of deep learning have been demonstrated in multiple domains, such as healthcare, autonomous driving vehicles, Web recommendation, and more. However, the extensive acquisition of data by the machine learning models owned by big companies has raised data privacy concerns. For instance, the general data protection regulation (GDPR)[1] by EU stipulates a range of data protection measures, resulting in the "data hungriness issues." Since data privacy is now the main ethical principle of machine learning systems [2], a solution is needed to extract a sufficient amount of training data while maintaining the data privacy. Furthermore, trustworthy AI has become an emerging topic lately due to the new ethical, legal, social, and technological challenges brought on by the technology [3].

Google proposed federated learning [4] in 2016 as an approach to solve the limited training data and data sharing restriction challenges. Federated learning trains a model collaboratively in a distributed manner. The data collected by each client for training can be utilized directly for local training, without transferring them to a centralized location. This addresses not only the data privacy issue but also the high communication costs as it does not need to transfer the raw data from the client devices to a central server.

However, federated learning struggles to achieve trustworthiness, i.e., responsible AI principles and requirements. In this work, we focus on the accountability and fairness challenges of trustworthy federated learning. First, as a large distributed system that involves different stakeholders, federated learning is vulnerable to accountability issues [5], [6]. Second, fairness issues also often occur in AI systems because of model bias and unfairness against specific groups [7], [8], and this challenge appears in federated learning systems caused by heterogeneous data distribution, specifically known as Non-IID[2] data.

To improve the accountability and fairness of federated learning systems, we proposed a blockchain-based trustworthy federated learning architecture. Blockchain and smart contracts are utilized for federated learning to maintain data integrity with its immutability [10]–[12]. The transparency property of blockchain ensures auditability and accountability, and this has been widely studied and evaluated [13]–[15]. Thus, we propose to leverage blockchain and smart contract technology to improve the accountability of federated learning systems. Designing such an integration is feasible as the designs of both federated learning and blockchain systems are decentralized in nature. We chose the COVID-19 detection scenario using X-rays as a use case to demonstrate and validate our approach. The contributions of this article are as follows.

Manuscript received 28 October 2021; revised 22 December 2021; accepted 10 January 2022. Date of publication 19 January 2022; date of current version 6 February 2023. *(Corresponding author: Sin Kit Lo.)*

Sin Kit Lo, Yue Liu, Qinghua Lu, Xiwei Xu, and Liming Zhu are with the Data61, CSIRO, Sydney, NSW 2015, Australia, and also with the School of Computer Science and Engineering, University of New South Wales, Sydney, NSW 2052, Australia (e-mail: kit.lo@data61.csiro.au).

Chen Wang is with the Data61, CSIRO, Sydney, NSW 2015, Australia.

Hye-Young Paik is with the School of Computer Science and Engineering, University of New South Wales, Sydney, NSW 2052, Australia.

Digital Object Identifier 10.1109/JIOT.2022.3144450

[1] https://gdpr-info.eu/

[2] Non-Identical and Independent Distribution: Skewed and personalized data distribution that differs across different clients and restricts the model generalization [9].

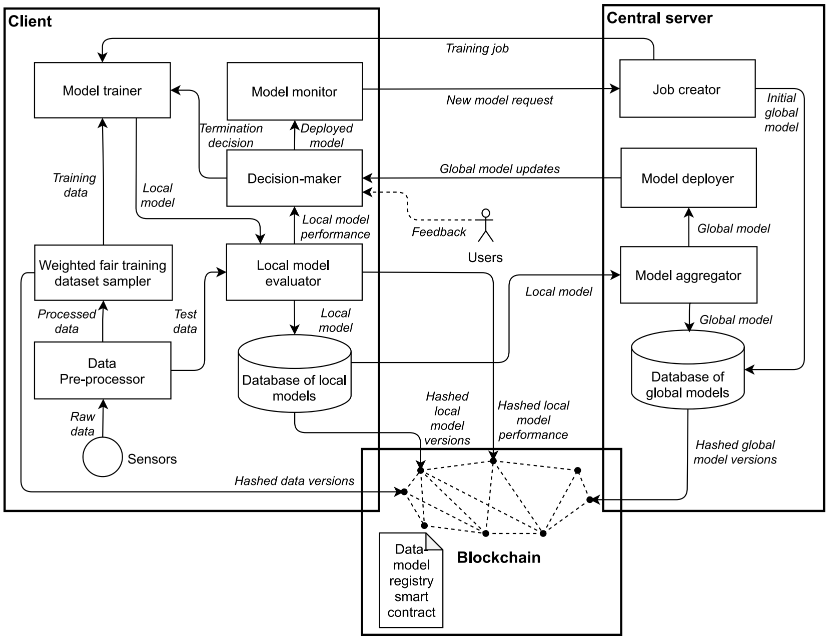

Fig. 1. Blockchain-based trustworthy federated learning architecture.

1) We present a blockchain-based trustworthy federated learning architecture to enable accountability in federated learning systems.
2) We design a smart contract-driven data-model provenance registry to track and record the local data used for local model training, and maps both the data and local model versions to the corresponding global model versions for auditing.
3) We propose a weighted fair training data set sampler algorithm to improve the fairness of data and models that are affected by the heterogeneity in data class distributions.

The remainder of this article is organized as follows. Section II describes the accountability and fairness issues that occur in COVID-19 X-ray detection using federated learning. Section III presents the blockchain-based trustworthy federated learning architecture. Section IV elaborates the weighted fair training data set sampler algorithm to address the fairness issue in federated learning. Section V evaluates the proposed approaches. Section VI discusses the related work. Finally, Section VII concludes this article.

II. Fairness and Accountability Issues in Federated Learning for COVID-19 Detection

A. Accountability

Federated learning across different parties is exposed to accountability issues, specifically between client devices and the central server. Conventionally, federated learning systems train models using local data that are undisclosed, and the data and local models trained are not tracked or mapped to the formed global models particularly. For instance, the details of the X-rays used to train each local model cannot be disclosed, and hence, the model user cannot check if the hospitals are providing genuine X-ray data. Furthermore, the local models from each hospital are also only evaluated locally and therefore the model user cannot determine which local models are poisoning the global model performance. Since model users cannot determine which party should be held accountable if the model is not performing properly, the federated learning system is not accountable. Therefore, we intend to leverage immutable and transparent blockchain to improve the accountability of the federated learning systems.

B. Fairness

Fairness issue in AI systems exists when the training data or the training procedure are biased [7], [16], [17]. In this work, we specifically target the unfairness caused by the biased training data. A model is fair when it is trained with balanced data, resulting in the model being generalized to the entire class distribution of the data. Suppose each hospital has X-ray scans of different lungs diseases, the X-rays numbers for each disease varies across different hospitals and the class distributions of these X-rays cannot be disclosed. Furthermore, the normal X-rays are the most abundant while positive COVID-19 cases are the least abundant. Hence, the data is biased toward normal. Moreover, the data from different hospitals cannot be collected and processed centrally. Therefore, we intend to enhance the fairness of the federated models by improving the fairness of the training data.

III. Blockchain-Based Trustworthy Federated Learning Architecture for COVID-19 Detection

In this section, we present the blockchain-based trustworthy federated learning architecture. We designed the architecture based on a reference architecture for the federated learning system named FLRA [6]. Fig. 1 illustrates the architecture,

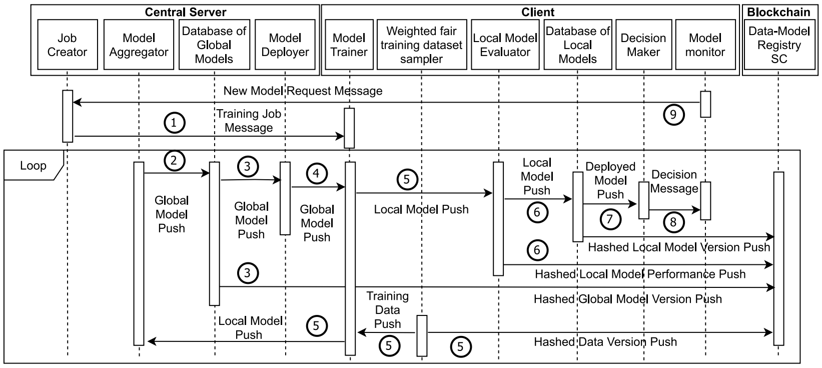

Fig. 2. Sequence diagram of blockchain-based trustworthy federated learning process.

which consists of four main components: 1) central server; 2) client; 3) blockchain; and 4) data-model registry smart contract. Fig. 2 is the sequence diagram that showcases the complete federated learning and blockchain processes.

A. Central Server

First, the *job creator* in the central server creates a model training job by initializing a global model and the training hyperparameters to be broadcast to the client's *model trainer*. Then, the *job creator* transfers the initial global model to the *database of global models*. The *model aggregator* on the central server then waits for the local model parameters from all the clients to perform model aggregation.

After the model aggregation, the updated global model is stored in the *database of global models*. The hashed value of the global model version is uploaded to the blockchain for provenance purposes and is then broadcast to the clients for the next training round by the *model deployer*. The *model deployer* will select the client devices to receive the global model updates based on the training performance or the availability of resources. In our default settings, all the clients will receive the updates to ensure fairness.

B. Client

The clients first collect the raw X-rays data and preprocess (image scaling, noise reduction, etc.) them by the *data preprocessor*. The processed data are then stored in the *database of local data*. After that, the training data is being sampled by the *weighted fair training data set sampler* which will be explained in Section IV. The sampled training data is used by the *model trainer* for local model training. The training data version of each training epoch is hashed and uploaded to the blockchain for data-model provenance. The model trainer set up the environment for local model training according to the training job received from the central server. After each local epoch, the local model is transferred to the *local model evaluator* for performance assessment. The hashed value of the local model versions and their performance are recorded and uploaded to the blockchain for data-model provenance.

The local model parameters are stored in the *database of local models* and are then sent to the *model aggregator* of the central server. After that, the *decision-maker* waits for the updated global model parameters from the *model deployer* of the central server. In each round, if the client is selected by the *model deployer*, it will receive the global model updates for the current round. The *decision-maker* will checks if the required federation epochs are achieved and decides when to terminate the training job. If terminate, the last global model version is deployed to the model users. Else, the entire process repeats until the designated federation epoch is reached.

When the training process terminates, the *decision-maker* in the client device stops all the local training to deploy the last global model. The *model monitor* then monitors the real-world data inference performance of the deployed global model. If the model performance degrades over a certain threshold level or user requests for a new model training job, the *model monitor* will trigger the *job creator* in the central server to initiate a new training job.

C. Blockchain

Each client and the central server will install at least one blockchain node to form a network. Each node holds a local replica of the complete transaction data in the form of a chain of blocks. The blockchain operations mainly cover the data-model provenance using smart contracts, in which all participants are identified via their blockchain addresses.

In every federation epoch, all local and global model parameters are stored in off-chain *database of local models* and *database of global models*, respectively. Meanwhile, the hashed local data versions are produced and recorded in the on-chain *data-model registry* smart contract to achieve the provenance and co-versioning of data and models. We explicitly track the data that is used to train each local model using a smart contract, without recording the actual local data on the blockchain due to data privacy considerations. The data version consists of the timestamp and data size, and the information is hashed before being uploaded to the blockchain.

Database systems are used to store the actual local and global models on client devices and the central server. We have tried to use blockchain to store both the local and global models but the dimension is too large for the continuous federated learning processes. Despite only recording the hashed model version on-chain, the record of models on-chain and off-chain are still immutable and transparent to relevant stakeholders.

```
contract DataModelRegistry{
    struct Model{
        bool uploaded;
        string data_version;
        string model_parameter;
    }
    struct Client{
        uint num_model;
    }
    mapping (address => mapping (uint => Model) ) public
         provenance;
    mapping (address => Client) public client;
    function getNumModel(address _client) public view
         returns (uint){
        return client[_client].num_model;
    }
    function storeData(string _data_version, string
         _model_parameter) public{
        required(provenance[msg.sender][client[msg.sender
             ].num_model+1].uploaded == false);
        client[msg.sender].num_model++;
        provenance[msg.sender][client[msg.sender].
             num_model].data_version = _data_version;
        provenance[msg.sender][client[msg.sender].
             num_model].model_parameter = _model_parameter
             ;
        provenance[msg.sender][client[msg.sender].
             num_model].uploaded = true;
    }
    function retrieveDataVersion(address _client, uint
         _model) public view returns (string _dataVersion)
         {
        return provenance[_client][_model].data_version;
    }
    function retrieveUpdate(address _client, uint _model)
         public view returns (string _modelPara){
        return provenance[_client][_model].model_parameter
             ;
    }
}
```

Listing 1. Data-model registry smart contract.

The provenance of an off-chain model can be validated by comparing its hash value with the corresponding on-chain record. Hence, to perform model provenance while maintaining the feasibility and efficiency of the federated learning process, the architecture applied the *off-chain data storage* design pattern [18], adopting database systems to store the actual model while the blockchain only records the hashed versions of models.

D. Data-Model Registry Smart Contract

The clients upload the hashed localmodel parameters and data version to the *data-model registry smart contract* each round. We illustrate the simplified code of smart contract in Listing 1. After each global aggregation, the central server sends the hashed global model parameters to the smart contract, whic are recorded in the struct *Model*. Another struct is implemented to count the number of uploads for each client. Via the on-chain hash map, the two structs are connected to clients through their on-chain addresses, which are used to retrieve on-chain data version and model parameter information. Nevertheless, there are two issues needed to be addressed during the upload process: 1) the size of a model that may be too large while the block size is fixed in a blockchain and 2) the on-chain data are transparent to all participants in the intrinsic design of blockchain, which may affect the security and privacy of uploaded models without proper access control.

To address these two vital issues, we apply both hashing and asymmetric/symmetric encryption techniques. First, the original models are all stored off-chain, while the hash values are sent to the blockchain. Hashing can transform the large model updates into a fix-length value which is much smaller while maintaining the data authenticity. Asymmetric/symmetric encryption techniques are utilized to further assure the confidentiality of the local models. Before sending a local model to the blockchain, the client devices hash the model parameters and then encrypt the hash value using his/her key. Only the encrypted text of the local models is placed in the smart contract. Clients can then share the decryption key to the central server in any channel, which is out of the scope in this article. After receiving the decryption key, the central server can retrieve on-chain encrypted text and conduct decryption to obtain the original hash value.

With the use of blockchain to store the hashed value of data, local, and global model versions, data-model provenance is achievable and users can audit the federated learning model performance. The *data-model registry* automatically records users' on-chain addresses for the mapping of model parameters and data versions, while blockchain transactions also include uploaders' information. These operation logs cannot be modified or removed due to the intrinsic tamper-proof design of blockchain, which implies that they can provide evidence for the audit trail of federated learning and hence, ensure on-chain accountability and improve the trustworthiness of the system.

IV. WEIGHTED FAIR TRAINING DATA SET SAMPLER

To improve the fairness of the model for COVID-19 detection using non-IID X-ray data, we propose an algorithm to dynamically sample training data from classes that have a relatively lower number of samples, according to the inverse of the weight distribution of the test data set. It balances the number of samples per class used to train a local model by ensuring the data samples with higher weight (lower sample size) will be sampled more than those that have lower weights (higher sample size). For instance, if the number of samples for class "COVID-19" is relatively lower in comparison to other classes, the possibility of the samples from that class being used to train the local model is higher. To further ensure that the data set is fair, the weights are calculated using the class distribution of the test data set. This ensures that all the clients have the same weights.

For this approach, there are several assumptions to be made as follows.

1) *Horizontal Federated Learning Setting Is Adopted:* As horizontal settings train models with the same feature space of the data (e.g., X-ray scans of lungs diseases) across different sample spaces (e.g., patients), the data set across different clients (e.g., hospitals) might have high variance in the number of samples (e.g., patients) for each class (e.g., types of lung disease). Therefore, horizontal federated learning will benefit more from the algorithm compared to the vertical settings that train models on the different features (e.g., different diseases) of the same sample space (e.g., patients).

TABLE I
DETAILS OF COVID-19 ONLINE DATA SETS

Title	Total	Normal	COVID-19	Lung Opacity	Pneumonia	URL
COVID-19-Radiography-Dataset	21,165	10,192	3,616	6,012	1,345	https://www.kaggle.com/tawsifurrahman/covid19-radiography-database
Figure 1 COVID-19 Chest X-ray Dataset Initiative	55	18	35	-	2	https://github.com/agchung/Figure1-COVID-chestxray-dataset
Total	21,220	10,210	3,651	6,012	1,347	

2) The test data sets should have the same class distributions (ratio of samples per type of lung diseases).

Suppose we have a test data set D_{test} that are used by all the clients, with a set of classes $C\{c_1, c_2 \cdots c_n\}$, each with a sample size of $S_c\{s_{c_1}, s_{c_2} \cdots s_{c_n}\}$, where n denotes the total number of classes. We first calculate the weights per class of the test data set $W\{w_1, w_2 \cdots w_n\}$ by dividing the total number of test data set, $\Sigma\ s_c$, by the number of samples of each class, s_{c_k}, as shown in

$$w_k = \frac{\sum_{k=1}^{n} s_{c_k}}{s_{c_k}}. \tag{1}$$

After that, we iteratively assign the w_k to every sample of the local training data according to their respective class. Each w_k represents the tendency of the sample from the class c_k should be sampled, which means the higher the value of w_k for a sample (lower s_c), the higher the possibility for it to be sampled out of the total local training data sets D_{train} of each client.

Based on the w_k assigned, batches of training data d_{le} will be sampled out of D_{train} in every local epoch le by each client to train their local models m. Finally, m from all the clients are collected by a central server and aggregated to update the global model M. The fairness of all the local models is enhanced since they are trained with data that are randomly weighted sampled to balance the possible bias that exists in the local training data set. The detailed federated training process with the weighted fair training data set sampler algorithm is illustrated in Algorithm 1.

V. EVALUATION

A. Federated Learning Performance

We simulated a medical diagnostic image classification task to detect COVID-19 using a federated learning environment. GFL federated learning framework[3] is used to perform the experiments. We utilized a total of 21 220 real-world X-rays images obtained from two data sets available online, consisting of a total of 10 210 normal, 3651 COVID-19, 6012 lung opacity, and 1347 pneumonia X-rays. The detailed breakdown of each data set is presented in Table I.

We set up a federated learning environment with one central server and three clients. The X-rays are randomly mixed, down-scaled, and evenly distributed across the three clients and one test data set. The detailed breakdown of the data configurations on each client is presented in Table II. Despite each client having the same total number of X-rays, all the data sets are biased and skewed toward normal and relatively less number of COVID-19 X-rays, which is similar to the real-world scenario. We conducted three groups of experiments each for (1) with weighted fair sampled training data sets, and (2) without weighted fair sampled training data sets. Based on (1), the set of weights W calculated from the test data set is [*Normal*: 2.0878, *COVID-19*: 5.894, *Lung Opacity*: 3.445, and *Pneumonia*: 16.373]. As observed, the normal X-rays have the lowest w as its number of samples, s_c, is the highest whereas the w of COVID-19 X-rays is relatively higher. Therefore, the

TABLE II
DATA SET CONFIGURATION FOR EACH CLIENT

Data classes	Client 1	Client 2	Client 3	Test dataset	Total per class
Normal	2,553	2,580	2,536	2,541	10,210
COVID-19	947	885	919	900	3,651
Lung Opacity	1,469	1,513	1,490	1,540	6,012
Pneumonia	336	327	360	324	1347
Total per client	5,305	5,305	5,305	5,305	-

Algorithm 1 Weighted Fair Federated Learning Training Data Set Sampler

```
On central server:
Initialises the model training job
Connects to all clients
Broadcast initial model training job to all clients
for federation epoch, $fe = 1, 2 \ldots n$ do
    On client:
    Receive model training job from central server
    Setup environment for local model training
    Calculate $W$ according to equation (1)
    Assign $W$ to the each training data samples
    for local epoch, $le = 1, 2 \ldots n$ do
        Sample $d_{le}$ according to $W$
        Train $m$ using $d_{le}$
        Test $m$ using $D_{test}$
        Record the loss $l$ and accuracy $acc$
    end for
    Upload $m$ to the central server
    On central server:
    collects $m$ from all clients
    Aggregate and update $M$
    Broadcast updated $M$ to all clients
end for
Save last $M$ as complete model
```

[3] https://github.com/GalaxyLearning/GFL

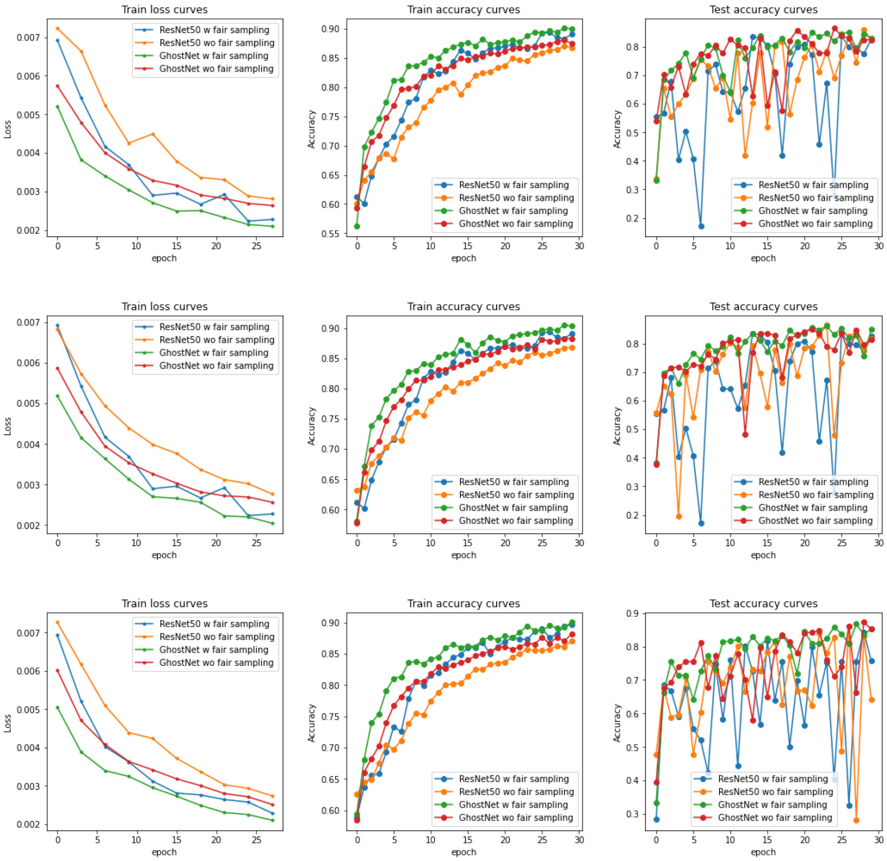

Fig. 3. Training losses, training accuracy, and test accuracy curves.

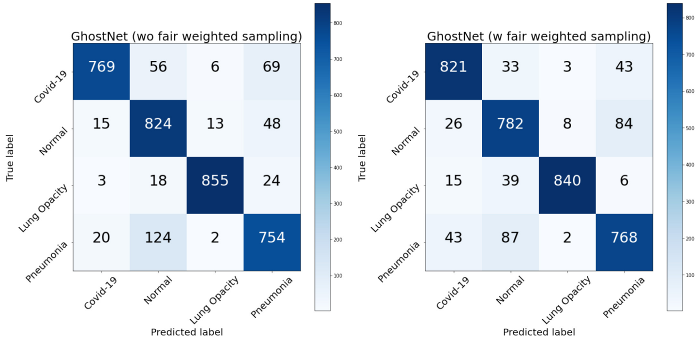

Fig. 4. Confusion matrices.

COVID X-rays will have a higher tendency to be sampled compared to normal X-rays.

We have trained and tested the ResNet50[4] and GhostNet [19] models with the hyperparameters as followed: ten federation epochs, three local epochs, a learning rate of 0.001, and ADAM as the optimizer. 12 experiments were conducted and the training losses, training accuracy, and test accuracy curves are illustrated in Fig. 3. Both the models trained with weighted fair sampled data achieved lower training losses, higher training, and test accuracy in all experiments while only one with a lower test accuracy is the ResNet50 in experiment group 2. The final readings of the training losses, training, and test accuracy are listed in Table III.

To evaluate the fairness of the models, we have created a test data set with 3600 X-rays which has an equal number of samples per classes. We used two models with the highest test accuracy: one with and one without weighted fair data sampling. The models' fairness result is presented in the confusion matrices, as shown in Fig. 4. We observed that the model without weighted fair sampling have predictions that are skewed toward normal X-rays which proves that fairness is affected by the training data distribution bias. The number of correct COVID-19 predictions of the models with weighted fair sampling is higher than the model without it and the test accuracy of the models with weighted fair sampling are also higher. Hence, the models with weighted fair data sampler are proven to have higher fairness and generalisability.

[4] https://pytorch.org/hub/pytorch_vision_resnet/

TABLE III
EXPERIMENTS RESULTS OF FEDERATED LEARNING

Exp. groups	Models	Fair weighted sampling	Training losses (%)	Training accuracy (%)	Test accuracy (%)
1	ResNet50	O	0.23	89.09	82.53
1	ResNet50	X	0.28	86.80	82.34
1	GhostNet	O	0.21	90.05	82.92
1	GhostNet	X	0.26	87.50	82.39
2	ResNet50	O	0.21	90.10	79.26
2	ResNet50	X	0.28	86.80	81.55
2	GhostNet	O	0.20	90.37	84.90
2	GhostNet	X	0.26	88.28	81.49
3	ResNet50	O	0.23	89.61	75.76
3	ResNet50	X	0.27	87.11	64.28
3	GhostNet	O	0.21	90.05	85.30
3	GhostNet	X	0.25	88.16	85.22

TABLE IV
EXPERIMENTS RESULTS OF BLOCKCHAIN OPERATION (MS)

	Single upload	Continuous upload	Parallel upload	Model retrieval
Minimum	56	532	778	3
First quartile	1142.75	2508.5	2200.25	4
Median	2274.5	4180	3283.5	4
Third quartile	3494.25	8451	4662.75	4.25
Maximum	4956	16452	6543	17
Average	2339.81	5932.92	3420.14	4.32

B. Blockchain and Smart Contract Performance

We conducted experiments to test the performance of involved blockchain operations. Our experiments examine the latency of the writing and reading model parameters via blockchain. Specifically, there are mainly three types of upload situations considering the setting of three clients in our experiments: single upload and model retrieval from one client, parallel and continuous uploads from all the clients. Hereby parallel upload means multiple models are sent to the blockchain at the same time, while continuous upload refers to serial order.

We adopted Parity consortium blockchain 1.9.3-stable, in which the consensus algorithm is Proof-of-Authority (PoA). The block gas limit is set to 80M and the block interval is configured to 5 s. The smart contracts are written in Solidity with compiler v.0.4.26. We performed four tests to measure the latency of the aforementioned blockchain operations, respectively, each test ran 100 times.

Table IV shows the results of blockchain operation latency. The upload operations include data hashing, encryption, and blockchain transaction inclusion, where the inclusion time is the dominating latency and depends on block generation interval. The average latency of the three upload scenarios is all around 5 ms which aligns with our setting of block interval. The maximum latency of continuous upload reaches 16 ms, which implies that the blockchain transactions from three clients are included in three consecutive blocks. Whilst, the maximum latency of parallel upload is still around 6 ms, which means that the three transactions are included in the same block. Retrieving model information from smart contract and decryption does not change on-chain data states and hence, no transaction is generated for inclusion, which enormously reduces the operation latency. Overall, from the experiment results, it can be observed that the blockchain and smart contract can achieve a satisfying performance to provide an accountable environment for the federated learning systems.

VI. RELATED WORK

The broad use of AI of building next-generation applications [20]–[23] generates concern about the use of AI systems that is human centered and trustworthy. For trustworthiness in federated learning systems, the questions often being asked are "Can the local model provided by the client devices be trusted to be nonadversarial?" "Is the local model provided by the client device genuinely trained by its local data?" and "Can the client trust the central server for the global model it provides?" The accountability challenges faced by federated learning systems are the ability to audit the data used to train each local model, the different local model provided by multiple client devices, and the global models created out of these local models.

Many research works have been done in addressing the accountability and auditability issues of federated learning systems and a great majority of them leveraged blockchain. For instance, Bao *et al.* [24] proposed an FLChain to build an auditable decentralized federated learning system to reward the honest trainer and detect the malicious nodes. Zhang *et al.* [25] proposed a blockchain-based federated learning approach for IIoT device failure detection which enables verifiable integrity and maintains the accountability of client data. Kang *et al.* [26] developed a reliable worker selection scheme using blockchain for reputation management of the trainers to defend against unreliable model updates. Kim *et al.* [10] proposed a blockchained federated learning architecture for the exchange and verification of local model updates.

Multiple frameworks and technical tools have been proposed by large private companies to implement trustworthy AI systems that focus on the fairness principle. For instance, Microsoft introduced Microsoft Fairlearn [27] to assess and improve the fairness of machine learning models through visualization dashboard and bias mitigation algorithms. IBM proposed IBM AI Fairness 360[5] to detect and mitigate unwanted bias in machine learning models and data sets.

[5] https://aif360.mybluemix.net/

Recent rapid growth of the COVID-19 pandemic has been elevated to a global crisis. This increases the usage of medical diagnostic images to determine COVID-19 cases [28] and triggered the usage of AI systems to detect COVID-19 infections through analysis of medical diagnostic images (e.g., X-ray and CT scans). However, medical data are highly privacy sensitive and the high-quality training data are limited. Federated learning has been adopted to connect isolated medical institutions to train classification and prediction models for medical diagnosis. Choudhury *et al.* [29] used federated learning to predict adverse drug reactions and Vaid *et al.* [30] used federated learning for mortality prediction in hospitalized COVID-19 patients. Liu *et al.* [31] showcased the conventional federated learning for COVID-19 detection using X-ray chest images. Kumar *et al.* [32] utilized federated learning for COVID-19 detection using the CT imaging and blockchain technology to further enhance data privacy. Zhang *et al.* [33] introduced a model fusion algorithm to improve the federated learning model performance and training efficiency on COVID-19 X-ray and CT images.

VII. Conclusion

This article proposed a blockchain-based federated learning approach to improve trustworthiness for medical diagnostic images analyses to detect COVID-19. This work is limited to only focusing on the fairness and accountability aspects of trustworthy AI. The registries built using blockchain and smart contract improve the accountability of the federated learning system. The weighted fair training data sampler approach has improved the fairness of the federated model trained. Overall, the evaluation results show that the proposed approaches are feasible and have achieved better performance than the conventional setting of federated learning in terms of accuracy, fairness, and generalisability. For future work, we will explore ways to improve fairness and trustworthiness through incentive mechanisms for federated learning systems using blockchain and smart contract.

References

[1] S. K. Lo, C. S. Liew, K. S. Tey, and S. Mekhilef, “An interoperable component-based architecture for data-driven IoT system,” *Sensors*, vol. 19, no. 20, p. 4354, 2019.

[2] A. Jobin, M. Ienca, and E. Vayena, “The global landscape of AI ethics guidelines,” *Nat. Mach. Intell.*, vol. 1, no. 9, pp. 389–399, Sep. 2019.

[3] S. Thiebes, S. Lins, and A. Sunyaev, “Trustworthy artificial intelligence,” *Electron. Markets*, vol. 31, pp. 447–464, Jun. 2021. [Online]. Available: https://doi.org/10.1007/s12525-020-00441-4

[4] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, “Communication-efficient learning of deep networks from decentralized data,” 2017, *arXiv:1602.05629*.

[5] S. K. Lo, Q. Lu, L. Zhu, H.-Y. Paik, X. Xu, and C. Wang, “Architectural patterns for the design of federated learning systems,” 2021, *arXiv:2101.02373*.

[6] S. K. Lo, Q. Lu, H.-Y. Paik, and L. Zhu, “FLRA: A reference architecture for federated learning systems,” in *Software Architecture*. Cham, Switzerland: Springer Int. Publ., 2021, pp. 83–98.

[7] M. Mohri, G. Sivek, and A. T. Suresh, “Agnostic federated learning,” in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 4615–4625.

[8] W. Du, D. Xu, X. Wu, and H. Tong, “Fairness-aware agnostic federated learning,” in *Proc. SIAM Int. Conf. Data Min. (SDM)*, 2021, pp. 181–189.

[9] F. Sattler, S. Wiedemann, K. Müller, and W. Samek, “Robust and communication-efficient federated learning from non-i.i.d. data,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020.

[10] H. Kim, J. Park, M. Bennis, and S. Kim, “Blockchained on-device federated learning,” *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1279–1283, Jun. 2020.

[11] S. K. Lo *et al.*, “Analysis of blockchain solutions for IoT: A systematic literature review,” *IEEE Access*, vol. 7, pp. 58822–58835, 2019.

[12] J. Weng, J. Weng, J. Zhang, M. Li, Y. Zhang, and W. Luo, “DeepChain: Auditable and privacy-preserving deep learning with blockchain-based incentive,” *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 5, pp. 2438–2455, Sep./Oct. 2021.

[13] Y. Xu, C. Zhang, Q. Zeng, G. Wang, J. Ren, and Y. Zhang, “Blockchain-enabled accountability mechanism against information leakage in vertical industry services,” *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1202–1213, Apr.–Jun. 2021.

[14] A. Boudguiga *et al.*, “Towards better availability and accountability for IoT updates by means of a blockchain,” in *Proc. IEEE Eur. Symp. Security Privacy Workshops (EuroS PW)*, 2017, pp. 50–58.

[15] R. Neisse, G. Steri, and I. Nai-Fovino, “A blockchain-based approach for data accountability and provenance tracking,” in *Proc. 12th Int. Conf. Availability Rel. Security*, 2017, p. 14. [Online]. Available: https://doi.org/10.1145/3098954.3098958

[16] “Tools for Trustworthy AI.” Org. Econ. Co-oper. Devel. 2021. [Online]. Available: https://www.oecd-ilibrary.org/content/paper/008232ec-en

[17] W. Du, D. Xu, X. Wu, and H. Tong, “Fairness-aware agnostic federated learning,” 2020, *arXiv:2010.05057*.

[18] X. Xu, C. Pautasso, L. Zhu, Q. Lu, and I. Weber, “A pattern collection for blockchain-based applications,” in *Proc. 23rd Eur. Conf. Pattern Lang. Programs*, 2018, p. 3. [Online]. Available: https://doi.org/10.1145/3282308.3282312

[19] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, “GhostNet: More features from cheap operations,” 2020, *arXiv:1911.11907*.

[20] M. Q. Raza and A. Khosravi, “A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings,” *Renew. Sustain. Energy Rev.*, vol. 50, pp. 1352–1372, Oct. 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032115003354

[21] X. Zhou, X. Xu, W. Liang, Z. Zeng, and Z. Yan, “Deep-learning-enhanced multitarget detection for end-edge-cloud surveillance in smart IoT,” *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12588–12596, Apr. 2021.

[22] X. Zhou, W. Liang, J. She, Z. Yan, and K. I.-K. Wang, “Two-layer federated learning with heterogeneous model aggregation for 6G supported Internet of Vehicles,” *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 5308–5317, Jun. 2021.

[23] X. Zhou, X. Yang, J. Ma, and K. I.-K. Wang, “Energy efficient smart routing based on link correlation mining for wireless edge computing in IoT,” *IEEE Internet Things J.*, early access, May 6, 2021, doi: 10.1109/JIOT.2021.3077937.

[24] X. Bao, C. Su, Y. Xiong, W. Huang, and Y. Hu, “FLChain: A blockchain for auditable federated learning with trust and incentive,” in *Proc. 5th Int. Conf. Big Data Comput. Commun. (BIGCOM)*, Aug. 2019, pp. 151–159.

[25] W. Zhang *et al.*, “Blockchain-based federated learning for device failure detection in industrial IoT,” *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5926–5937, Apr. 2021.

[26] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, “Reliable federated learning for mobile networks,” *IEEE Wireless Commun.*, vol. 27, no. 2, pp. 72–80, Apr. 2020.

[27] S. Bird *et al.*, “Fairlearn: A toolkit for assessing and improving fairness in AI,” Microsoft, Redmond, WA, USA, Rep. MSR-TR-2020-32, May 2020. [Online]. Available: https://www.microsoft.com/en-us/research/publication/fairlearn-a-toolkit-for-assessing-and-improving-fairness-in-ai/

[28] I. Blažić, B. Brkljačić, and G. Frija, “The use of imaging in COVID-19—Results of a global survey by the international society of radiology,” *Eur. Radiol.*, vol. 31, no. 3, pp. 1185–1193, 2021.

[29] O. Choudhury, Y. Park, T. Salonidis, A. Gkoulalas-Divanis, I. Sylla, and A. Das, “Predicting adverse drug reactions on distributed health data using federated learning,” in *Proc. AMIA Annu. Symp.*, vol. 2019, 2019, pp. 313–322.

[30] A. Vaid *et al.* “Federated Learning of Electronic Health Records Improves Mortality Prediction in Patients Hospitalized with COVID-19.” medRxiv. 2020. [Online]. Available: https://doi.org/10.1101/2020.08.11.20172809

[31] B. Liu, B. Yan, Y. Zhou, Y. Yang, and Y. Zhang, "Experiments of federated learning for COVID-19 chest X-ray images," 2020, *arXiv:2007.05592*.

[32] R. Kumar *et al.*, "Blockchain-federated-learning and deep learning models for COVID-19 detection using CT imaging," 2020, *arXiv:2007.06537*.

[33] W. Zhang *et al.*, "Dynamic fusion-based federated learning for COVID-19 detection," *IEEE Internet Things J.*, vol. 8, no. 21, pp. 15884–15891, Nov. 2021.

Sin Kit Lo received the B.S. degree in electronics and electrical engineering from Sungkyunkwan University, Seoul, South Korea, in 2017, and the M.S. degree in computer science from the University of Malaya, Kuala Lumpur, Malaysia, in 2020. He is currently pursuing the computer science Ph.D. degree in software engineering with AI Team, Data61, CSIRO, Sydney, NSW, Australia, and the School of Computer Science and Engineering, University of New South Wales, Sydney.

His research interests include federated learning and decentralised artificial intelligence, specifically in software architecture design.

Yue Liu received the M.S. degree in computer science from the China University of Petroleum (East China), Dongying, China, in 2020. He is currently pursuing the Ph.D. degree in computer science with the Architecture and Analytics Platforms Team, Data61, CSIRO, Sydney, NSW, Australia, and the School of Computer Science and Engineering, University of New South Wales, Sydney.

His research interests include blockchain as a service, blockchain governance, and self-sovereign identity.

Qinghua Lu (Senior Member, IEEE) received the Ph.D. degree from the University of New South Wales (UNSW), Kensington, NSW, Australia, in 2013.

She is a Principle Research Scientist and the Team Leader of Software Engineering with the AI Team, Data61, CSIRO, Sydney, NSW. She is also a Conjoint Senior Lecturer with UNSW. She formerly worked as a Researcher with NICTA, Sydney. She has published more than 100 academic papers in the international journals and conferences. Her recent research interests include software engineering for AI, responsible AI, software architecture, and blockchain.

Chen Wang received the Ph.D. degree in computer science from Nanjing University, Nanjing, China.

He is a Principal Research Scientist with Data61, CSIRO, Sydney, NSW, Australia. He leads and develops machine learning and data analytics systems for various domains, including radio astronomy, health, and agriculture as well as smart grids. His current research interests are on the interpretability, robustness, and scalability of data-driven systems.

Xiwei Xu received the Ph.D. degree from the University of New South Wales (UNSW), Sydney, NSW, Australia, in 2012.

She is a Principal Research Scientist with the Architecture and Analytics Platforms Team, Data61, CSIRO, Sydney. She is also a Conjoint Senior Lecturer with UNSW. She has been started working on blockchain since 2015. She is doing research on blockchain from software architecture perspective, for example, tradeoff analysis, and decision making and evaluation framework. Her main research interest is software architecture. She also does research in the areas of service computing, business process, and cloud computing and dependability.

Hye-Young Paik (Member, IEEE) received the Ph.D. degree in computer science from the University of New South Wales (UNSW), Sydney, NSW, Australia, in 2004.

She is a Senior Lecturer with the School of Computer Science and Engineering, UNSW. She collaborates with the Architecture and Analytics Platforms Team, Data61, Commonwealth Scientific and Industrial Research Organization, Sydney, as a Visiting Academic. Her research interests include service-oriented software design and architecture and distributed data/application integration.

Liming Zhu received the Ph.D. degree in software engineering from the University of New South Wales (UNSW), Sydney, NSW, Australia, in 2007.

He is the Research Director of Data61, Commonwealth Scientific and Industrial Research Organization, Sydney, NSW, Australia, and a Conjoint Full Professor with UNSW. He has published more than 150 academic papers on software architecture, secure systems, and data analytics infrastructure. His research program focuses on big data platforms, computational science, blockchain, regulation technology, privacy, and cybersecurity.

Prof. Zhu is the Chair of Standards Australia's Blockchain and Distributed Ledger Committee.