BlockFed: A Novel Federated Learning Framework Based On Hierarchical Aggregation

1st
xyz

2nd
xyz

3rd
xyz

This research introduces a novel federated learning framework that utilizes a novel hierarchical aggregation strategy in which the global model is generated by the clients after some levels of aggregation. Also, a novel role definition strategy is implemented in this framework for determining the roles and tasks of the clients in each round. Furthermore, IPFS and blockchain are used for storing local models and their hash pointers respectively. The proposed framework is evaluated on a genomic breast cancer dataset from the GDC portal. The achieved results showed 98% accuracy of the global model after 10 rounds of learning.

Keywords: federated learning, hierarchical aggregation, role definition, gene data, blockchain, deep learning.

I. Introduction

Huge amounts of data are being generated these days by different tools such as home appliances and smartphones. Growing the Internet of Things technology has resulted in an enormous amount of data that can be used for training different learning models for different purposes. Based on the statistics reported in [1], the smart home market size is around 53.3 billion which is mainly enabled by IoT devices. Despite, the generation of huge datasets, one of the main challenges in using these datasets for training learning models is the lack of sufficient data. In many cases due to the privacy and sensitivity of data, data owners are reluctant to make their data public or donate it for scientific purposes. Federated learning is a promising solution for this challenge in which the privacy of the data is guaranteed since each data owner can train his/her model and share the model's parameters instead of sharing the data. In federated learning, the clients send their local model parameters, trained locally on the local devices, to a central server for aggregation. After the local models are aggregated, the clients load the global model from the central server and train their model with their local training data in the next round of learning. While in the traditional distributed learning systems, a low degree of transparency and centralization are two main drawbacks, federated learning has covered these issues by utilizing blockchain and making use of its merits including transparency. In other words, integrating blockchain with federated learning can result in secure data retrieval and accurate, traceable, and transparent model training [2]. In this research, a federated learning architecture empowered by a novel hierarchical aggregation mechanism is introduced to train a global machine learning algorithm. The efficacy of the architecture is evaluated on different datasets. The main contributions of this research are:

1) **Enhance global model safety and transparency with blockchain-based aggregator smart contract and IPFS.**
2) **Utilizing a novel hierarchical aggregation mechanism and task definition for increasing efficacy.**
3) **Improve framework efficiency with Deep learning models.**

The rest of the paper is organized as follows: In the next section, the research background is explained. After that in the third section the related works, their cons, and pros are reviewed. Section 4 is devoted to the proposed model and the last section is experimental results.

II. Research background

A. Federated learning

Despite the generation of enormous data by different devices, the lack of sufficient training data in some specific areas is the main issue for machine learning projects. In federated learning, data owners can keep the privacy of their data and at the same time participate in training a global learning model. In other words, the private data will never leave the local devices. In this process, clients, train their local models using their local private data and then send the parameters (weights and biases) of the training models to an aggregator for aggregation. When all of the clients send their local model's parameters, the aggregator aggregates the parameters by averaging them and generates the global model. Then at the beginning of the next round, each client loads the global model, trains its model with the local data, and sends the new parameters to the aggregator for aggregation. Thus federated learning consists of a predefined number of learning rounds in each of them the clients perform the following tasks respectively:

1) Loading the global model.
2) Setting the global parameters (weights and biases) in their local learning model.
3) Training their local model.
4) Sending the new parameters (weights and biases) to the aggregator for aggregation.

B. Blockchain

Generally, blockchain is a technology for data storage. This technology, which has gained attention with the introduction of Bitcoin, is used as a payment system (instead of a banking system) for executing and storing transactions. In other words, a Blockchain is a chained list of blocks each of which contains a list of transactions. In this technology, miners, add the transactions made by other nodes to their block and distribute it over the network. When a block is verified by the majority of the nodes, then it is added to the chain. The main feature of these blocks is that every user can see the transactions stored in them. So these blocks are the transparent units that store data.

C. Smart contract

A smart contract is a term referred to computer programs that can be executed in advanced blockchain such as Ethereum. In other words, a" smart contract" is a script that can be deployed in the blockchain and run by a network of mutually distrusting nodes without the need for an external trusted authority. This feature turns the blockchain into a general computational platform [3]–[5].

III. Literature review

In [6] Zolotareva et al. have proposed a federated learning framework, called Flimma, for differential gene expression analysis. Their main goal was cancer detection. In this framework, each client trains a regression model and sends the corresponding parameters to the central server for aggregation.

Dipro et al, proposed a federated learning model for Parkinson's prediction in [7]. In their proposed model clients train convolutional neural networks with topography image data and share the corresponding parameters through a central server and also blockchain. Although they have used blockchain in their framework but, the global model is saved in a central server.

In [8] a federated learning architecture for post-surgery cancer recurrence prediction is proposed based on convolutional neural network and patients' clinical data. Similarly in this framework, a central server is used for aggregation. Also to keep the privacy of the patient's data a localized differential privacy mechanism is established on the user side. In fact, in this framework, they have to keep the privacy of the patients' data by utilizing federated learning.

In [9] a federated learning framework is proposed for network intrusion detection by Tang et al. Their framework is utilized to keep the privacy of the data owned by different institutions. They evaluated their model on a benchmark dataset, called CICIDS 2017. Their result revealed the promising performance of FL where the accuracy of the global model was almost the same as the centralized trained model.

In another research, Hosseini et al have used federated learning for the estimation of power generation in a grid [10]. Estimating the power generated by different renewable energy sources is a challenge due to the privacy of the data owned by different clients (BTM sites). Thus using federated learning, Hosseini et al. have trained a global model for this task. Their result shows a negligible reduction in accuracy in comparison to a centralized trained model.

Dipro et al have used federated learning to develop a global model for Parkinson's prognosis [7]. In their model. the local learners train a CNN locally with Tomography images and share the corresponding gradient descent information through a central server. Although blockchain is used for aggregation but still storing the models' parameters in a central server is a drawback due to exposing the whole system to failure problems.

In our previous work [11] we proposed a federated learning framework in which the aggregation is done by combining kfedAvg and particle Swarm Optimization (PSO). In that framework, in each learning round, each client loads other clients' models, evaluates their accuracy on his local test data, and selects the top 2 models. Then PSO is run with an initial population generated using the selected models. Finally, the best solution found by PSO is set as the weights of the local model. Although we utilized a novel idea for aggregation in the framework each client has his model and there is no global model. This restricts the usability of the framework in real-world applications in which having a global model is necessary.

The main drawbacks of the most reviewed research are centralization, lack of incentivization infrastructure, and usability restrictions. To cover these issues we propose a new framework, BlockFed, with the following features:

1) A novel hierarchical aggregation mechanism that potentially increases the security.
2) Generating an environment for the participants (clients) to bring more computation resources which improves the efficiency of the whole framework.
3) Using IPFS for saving the trained models instead of blockchain so the clients can train complex models and their corresponding huge matrices of weights without limitation.
4) Using blockchain for storing IPFS hash pointers which increases transparency and trustworthiness.

IV. Proposed Framework

The whole framework of the proposed method is shown in figure 1.

According to this figure, the participants in this framework are categorized into two main groups: learners and aggregators. Learners are responsible for training local models and sharing them through IPFS and blockchain while aggregators are responsible for aggregating the trained local models. In each round of learning, learners perform the following tasks respectively:

1) Loading the global model.
2) Training their local model using their local train data.
3) Storing their local model parameters in the IPFS.
4) Storing the corresponding IPFS hash pointer in a smart contract deployed in the Ethereum blockchain.

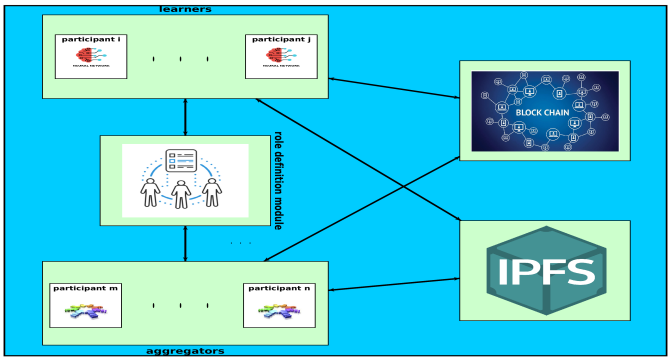

Fig. 1. Architecture of the proposed framework.

In figure 2 the pseudo-code of the procedure done by each learner is demonstrated.

```
global_IPFS_hash_pointer = Trainer.deployed_contract.read_global_model()
global_weights = Trainer.read_from_IPFS(global_IPFS_hash_pointer)
Trainer.set(global_weights)
Trainer.train()
updated_weights = Trainer.get_weights()
hash_pointer = Trainer.write_on_IPFS(updated_weights)
Trainer.deployed_contract.update(hash_pointer)
```

Fig. 2. Pseudo-code of the tasks performed by each learner in each learning round.

On the other side, aggregators aggregate the locally trained models, stored by the learners, in a hierarchical scenario. This means that the aggregators are divided into different groups based on the level of aggregation. In this scenario, the aggregators in each level, accept the models from the previous level, aggregate them, and pass the aggregated model to the next level. The last level has only one aggregator which calculates the global model by aggregating the received models from the aggregators in the previous level. Imagine that we have two levels of aggregation. The first-level aggregators accept the models of the learners, aggregate, and send them to the second level for aggregation. The aggregator in the second level averages the previous aggregated models and generates the global model. Similar to the learners, each aggregator after averaging the received models from the previous level, stores the aggregated model in the IPFS and also stores the corresponding hash pointers in the smart contract. Thus the following steps are done by each aggregator at each level of aggregation:

1) Loading the aggregated models stored in IPFS by the previous level aggregators.
2) Averaging the loaded models.
3) Storing the averaged models in IPFS.
4) Storing the corresponding IPFS hash pointer in the smart contract.

```
IPFS_hash_list = Aggregator.deployed_contract.read_previous_round()
for IPFS_hash in IPFS_hash_list:
    weights.append(Aggregator.read_from_IPFS(IPFS_hash))
averaged_weights = Aggregator.average(weights)
IPFS_hash = Aggregator.write_on_IPFS(averaged_weights)
Aggregator.deployed_contract.update(IPFS_hash)
```

Fig. 3. Pseudo-code of the tasks performed by each aggregator in each learning round.

Thus each participant's task is defined based on his role in the federated learning procedure. Furthermore, the proposed framework is empowered by a role definition module which redefines the role of each participant for the next round at the end of the current round. In other words, at the end of each round, the role of each participant (learner or aggregator) is defined for the next round. This role definition can be done based on different evaluations. For example, the participants who bring more computation resources to the whole procedure, or the participants who have more training data, can be selected as learners to train the local model while the others can be assigned as aggregators. In our simulations, we did this based on the accuracy of the trained models by each participant. So at the end of each learning round, each participant (even aggregators) sends the accuracy of their trained model to the contract and then the role definition function of the contract determines their role based on the received accuracy scores.

The main benefit of the dynamic role definition is that it makes the incentivization mechanism more effective. In the proposed incentivization mechanism, the participants receive rewards based on their contribution to the whole procedure. This means that the learners receive more rewards since their trained models have better accuracy on the test data in the previous round. The dynamic role definition idea gives the chance to aggregators to change their roles and receive more rewards in the next rounds. In other words, this novel role definition idea increases the efficiency of the federated learning by creating a competition between the participants during the learning rounds.

V. Experimental Results

To evaluate the performance of the proposed framework, we conducted several experiments in each of them the framework is simulated with the following settings:

1) In each round of learning four trainers are selected among the clients to train their local models. The selection of the trainers is done based on the accuracy of their models on the global test data.

2) The global model is updated in two levels of aggregation in each round.
3) Two clients are assigned as first-level aggregators and one client is assigned as second-level aggregators.
4) Each first-level aggregator, calculates the average of the weights of two models (out of four models) trained by the trainers and uploaded in the IPFS and blockchain.
5) The second level aggregator, averages the two aggregated models from the first level of aggregation and uploads it as the global model in the IPFS and blockchain.
6) Each trainer in each learning round, trains a Convolutional Neural Network (CNN) with two layers containing 500 neurons and a Relu activation function in addition to a softmax activation function in the final layer.
7) All of the simulations are done using the Breast cancer genomic data, we used in our previous work [11]. We downloaded the same data so that we can compare the performance of our new framework with the previous one. This dataset is described completely in the next subsection.
8) At the end of each learning round, the role definition module determines the roles of the clients (trainer, first-level aggregator, or second-level aggregator) for the next round. This role definition is done based on the accuracy score of the clients' locally trained models on the global test data. So at the end of each learning round, all of the clients (even the first-level and second-level aggregators) train a CNN model on their local data and evaluate its accuracy on the global test data. After that, the role definition module, first sorts the the clients based on their accuracy scores in a descending form, and then:
   - Assigns the first four clients as trainers for the next round.
   - Assigns the next two clients as the first-level aggregators for the next round.
   - Assigns the last client as the second-level aggregator for the next round.

A. Dataset Description

We downloaded the Breast cancer gene dataset from the GDC portal using TCGAbiolink and DT packages in the R programming language. These packages download the corresponding patient files by sending queries with specific filter values and generate a tabular dataset by extracting FPKM values for each gene of each patient. The selected filters and the corresponding values are shown in table I.

TABLE I
FILTERS AND THEIR VALUES

Project	TCGA-BRCA
Data.category	Transcriptome Profiling
Data.type	Gene Expression Quantification
Experimental.strategy	RNA-Seq
Workflow.type	STAR-Counts
Sample.type	Primary Tumor/Solid Tissue Normal

In this dataset, which has 200 rows, each row corresponds to a patient which is classified in two classes: Primary Tumor and Solid normal tissue. Thus the learning problem is a binary classification problem. In addition, the dataset contains 60677 columns each of which contains the FPKM values of a specific gene in each patient.

B. Experiments And Results

We evaluated the efficacy of the proposed framework by conducting three experiments in which we simulated the framework with the aforementioned settings in different numbers of rounds. The corresponding results are shown in tables II to VI. These tables contain the accuracy scores of the clients' trained models as well as the accuracy score of the global model in each learning round. Table II shows the results achieved for the first experiment in which the clients trained the global model in 8 rounds.

TABLE II
FIRST EXPERIMENT: ACCURACY SCORES OF THE CLIENTS AND THE GLOBAL MODEL IN DIFFERENT ROUNDS IN A SIMULATION WITH 8 ROUNDS

Round	2	3	4	5	6	7	8
Client 1	50	55	68	78	80	85	86
Client 2	51	61	65	71	75	85	86
Client 3	51	53	58	70	76	76	83
Client 4	53	53	53	55	55	55	58
Client 5	50	55	66	76	81	83	88
Client 6	53	56	73	76	83	86	90
Client 7	55	60	58	56	56	60	58
Global model	48	50	61	71	76	81	86

This table clearly shows that at the end of the simulation, the accuracy of the global model has reached 86% on the global test data, which is outstanding. Also, this table shows that round by round, not only each client's local model's accuracy is improved but also the global model's accuracy has been increasing during the simulation. This indicates the evolving nature of the proposed framework. This concept is also demonstrated in figure 4. In this figure, the accuracy scores of the clients and also the accuracy score of the global model, during the simulation, are shown through different graphs

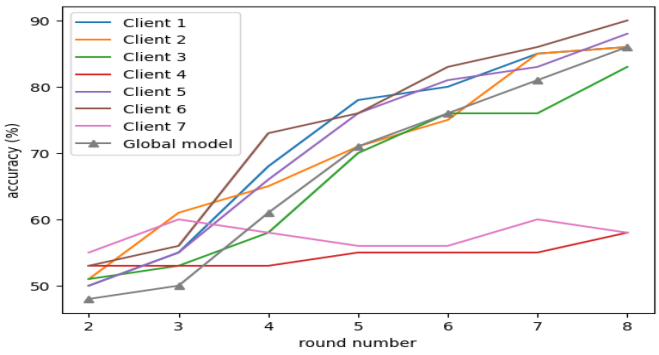

Fig. 4. Accuracy scores of the clients and the global model in each round of the first experiment.

Table III shows the performance of the role definition module in each round of the first experiment.

TABLE III
FIRST EXPERIMENT: ROLES OF THE CLIENTS IN DIFFERENT ROUNDS

Rounds	Trainers	Level 1 aggregators	Level 2 aggregator
2	2,3,4,7	1,5	6
3	3,4,6,7	2,5	1
4	2,5,6,7	1,4	3
5	1,2,5,6	3,7	4
6	1,2,5,6	3,7	4
7	1,3,5,6	2,7	4
8	1,2,5,6	3,7	4

According to this table, in the second round, clients 2, 3, 4, and 7 were assigned as trainers, clients 1 and 5 as level 1 aggregators, and client 6 as level 2 aggregators. In the last round, clients 1, 2, 5, and 6 were the trainers, clients 3 and 7 were the level 1 aggregator, and client 4 was the level 2 aggregator. As mentioned before, in our simulations, the role definition module defines the roles of each client in each round based on its performance in the previous round. So we can see that client 4 which has shown the poorest performance in round 7 is assigned as the level 2 aggregator in round 8 while clients 1, 2, 5, and 6 which have the best performance in round 7 are assigned as trainers for the last round. Since the roles in the first round are defined randomly, they are ignored in table III

Tables IV and V show the accuracy scores of the corresponding roles in each round of the second experiment.

TABLE IV
SECOND EXPERIMENT: ACCURACY SCORES OF THE CLIENTS AND THE GLOBAL MODEL IN DIFFERENT ROUNDS IN A SIMULATION WITH 10 ROUNDS

Round	2	3	4	5	6	7	8	9	10
Client 1	46	46	46	50	61	65	78	86	91
Client 2	53	70	80	85	86	88	88	90	91
Client 3	80	88	91	93	93	95	98	98	98
Client 4	85	96	95	96	95	95	96	98	96
Client 5	95	93	93	93	96	96	96	95	95
Client 6	78	78	85	91	91	93	93	93	93
Client 7	53	58	66	73	73	73	75	75	80
Global model	65	95	93	93	95	96	96	96	96

According to table IV we can see that the final accuracy of the global model has reached 96% while it was 86% in the previous experiment. Also, figure 5 indicates how the accuracy of the local models and the global model has changed during the simulation in this experiment. Like the previous experiment, in this experiment, we can also see the improving pattern in the performance of the local clients which has resulted in an improving pattern of the global model. This indicates the efficiency of the proposed aggregation mechanism.

According to table V we can see that clients 3, 4, 5, and 7 are selected as trainers in all rounds. Table IV shows that these clients have shown the best performance among other clients in all learning rounds.

Tables VI and VII contain the same results for the third experiment in which the proposed framework is simulated

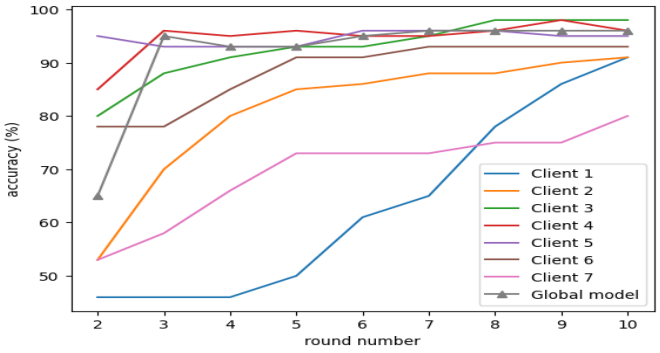

Fig. 5. Accuracy scores of the clients and the global model in each round of the second experiment.

TABLE V
SECOND EXPERIMENT: ROLES OF THE CLIENTS IN DIFFERENT ROUNDS

Rounds	Trainers	Level 1 aggregators	Level 2 aggregator
2	3,4,5,6	2,7	1
3	3,4,5,6	2,7	1
4	3,4,5,6	2,7	1
5	3,4,5,6	2,7	1
6	3,4,5,6	2,7	1
7	3,4,5,6	2,7	1
8	3,4,5,6	2,7	1
9	3,4,5,6	1,2	7
10	3,4,5,6	1,2	7

TABLE VI
THIRD EXPERIMENT: ACCURACY SCORES OF THE CLIENTS AND THE GLOBAL MODEL IN DIFFERENT ROUNDS IN A SIMULATION WITH 12 ROUNDS

Round	2	3	4	5	6	7	8	9	10	11	12
Client 1	98	98	98	98	98	95	98	98	98	98	98
Client 2	95	96	98	98	98	98	98	98	98	98	98
Client 3	96	98	98	98	98	98	98	98	98	98	98
Client 4	96	86	98	93	95	96	93	91	98	98	98
Client 5	93	91	85	98	98	98	98	98	98	96	98
Client 6	98	95	93	98	98	98	98	98	98	98	98
Client 7	95	98	95	98	98	98	98	98	98	98	96
Global model	95	96	98	98	98	98	98	98	98	98	98

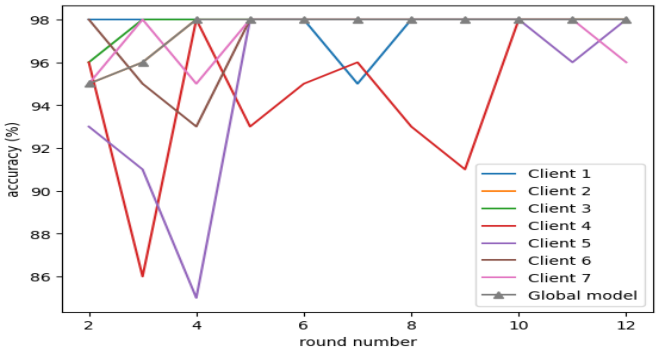

Fig. 6. Accuracy scores of the clients and the global model in each round of the third experiment.

in 12 rounds. From table VI it can be seen that the global model has reached 98% accuracy from the second round of

TABLE VII
SECOND EXPERIMENT: ROLES OF THE CLIENTS IN DIFFERENT ROUNDS

Rounds	Trainers	Level 1 aggregators	Level 2 aggregator
2	3,4,5,6	2,7	1
3	1,3,4,6	2,7	5
4	1,2,3,7	5,6	4
5	1,2,3,4	7,7	5
6	3,5,6,7	1,2	4
7	3,5,6,7	1,2	4
8	3,5,6,7	2,4	1
9	3,5,6,7	1,2	4
10	3,5,6,7	1,2	4
11	4,5,6,7	2,3	1
12	3,4,6,7	1,2	5

learning. Also figure 6 shows the increasing pattern of the global model's accuracy despite the small fluctuations in the accuracy of some models like clients 4 and 5.

Table VIII contains our previous results reported in [11]. In our previous work, we proposed a framework using a PSO-based aggregation mechanism in which each client performs the aggregation independently on his local device using PSO. Although the previous results show a slight superiority of the previous framework over BlockFed (clients 5 and 6), but the average accuracy scores of the clients in BlockFed are higher which shows that the overall performance of BlockFed is better. Table IX shows the average accuracy scores for the two methods.

TABLE VIII
OUR PREVIOUS RESULTS REPORTED IN [11]

Round	2	3	4	5	6	7	8	9	10
Client 1	41	16	95	95	95	91	95	95	95
Client 2	79	79	20	91	79	87	87	91	91
Client 3	50	91	50	95	91	95	95	95	95
Client 4	29	91	87	91	91	87	29	70	75
Client 5	62	100	100	100	100	100	100	100	100
Client 6	45	45	100	100	100	100	100	100	100

TABLE IX
AVERAGE PERFORMANCE OF THE CLIENTS IN TWO METHODS

Round	Our framework	Framework in [11]
2	95.87	51
3	94.57	70.33
4	95	75.33
5	97.28	95.33
6	97.57	92.83
7	97.28	93.33
8	97.28	84.33
9	97	91.83
10	98	92.66

VI. CONCLUSION AND FUTURE WORKS

In this research, we proposed a federated learning framework with a novel hierarchical aggregation mechanism that moves the computations from the central smart contract to the clients. In this hierarchical scenario, the aggregation of the local models is calculated by several clients at different levels. This mechanism increases the security of the whole framework in which the performance of the honest clients in each level of aggregation will cover the performance of the malicious ones. Also, the achieved results indicate the efficiency of the proposed aggregation strategy.

Furthermore, a role definition module is utilized which defines the roles of the clients (trainer, level *i* aggregator) in each round based on their performance in the previous round. This can generate a competitive environment that motivates the clients to bring more computational resources to the whole framework to earn more rewards.

We also compared our results with our previous work. The average results show the slight superiority of our new framework although we reached higher local accuracy scores in our previous work. Furthermore, our previous framework doesn't have a global model, instead, each client proposes a local model. This potentially restricts its applicability. For future works, we suggest:

1) Using weighted average in each level of aggregation instead of simple averaging.
2) Determining the roles of the clients based on the amount of local data they possess.
3) Using Zero-Knowledge proof to increase the security.

REFERENCES

[1] Yang Zhao, Jun Zhao, Linshan Jiang, Rui Tan, Dusit Niyato, Zengxiang Li, Lingjuan Lyu, and Yingbo Liu. Privacy-preserving blockchain-based federated learning for iot devices. *IEEE Internet of Things Journal*, 8(3):1817–1829, 2020.

[2] Yunlong Lu, Xiaohong Huang, Yueyue Dai, Sabita Maharjan, and Yan Zhang. Blockchain and federated learning for privacy-preserved data sharing in industrial iot. *IEEE Transactions on Industrial Informatics*, 16(6):4177–4186, 2019.

[3] Shailendra Rathore, Byung Wook Kwon, and Jong Hyuk Park. Blockseciotnet: Blockchain-based decentralized security architecture for iot network. *Journal of Network and Computer Applications*, 143:167–177, 2019.

[4] Shuai Wang, Liwei Ouyang, Yong Yuan, Xiaochun Ni, Xuan Han, and Fei-Yue Wang. Blockchain-enabled smart contracts: architecture, applications, and future trends. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(11):2266–2277, 2019.

[5] Toqeer Ali Syed, Ali Alzahrani, Salman Jan, Muhammad Shoaib Siddiqui, Adnan Nadeem, and Turki Alghamdi. A comparative analysis of blockchain architecture and its applications: Problems and recommendations. *IEEE access*, 7:176838–176869, 2019.

[6] Olga Zolotareva, Reza Nasirigerdeh, Julian Matschinske, Reihaneh Torkzadehmahani, Tobias Frisch, Julian Späth, David B Blumenthal, Amir Abbasinejad, Paolo Tieri, Nina K Wenke, et al. Flimma: a federated and privacy-preserving tool for differential gene expression analysis. *arXiv preprint arXiv:2010.16403*, 2020.

[7] Sumit Howlader Dipro, Mynul Islam, Md Abdullah Al Nahian, and Moonami Sharmita Azad. *A federated learning approach for detecting Parkinson's disease through privacy preserving by blockchain*. PhD thesis, Brac University, 2022.

[8] Zezhong Ma, Meng Zhang, Jiajia Liu, Aimin Yang, Hao Li, Jian Wang, Dianbo Hua, and Mingduo Li. An assisted diagnosis model for cancer patients based on federated learning. *Frontiers in Oncology*, 12:860532, 2022.

[9] Zhongyun Tang, Haiyang Hu, and Chonghuan Xu. A federated learning method for network intrusion detection. *Concurrency and Computation: Practice and Experience*, 34(10):e6812, 2022.

[10] Paniz Hosseini, Saman Taheri, Javid Akhavan, and Ali Razban. Privacy-preserving federated learning: Application to behind-the-meter solar photovoltaic generation forecasting. *Energy Conversion and Management*, 283:116900, 2023.

[11] Reza Nourmohammadi, Iman Behravan, and Kaiwen Zhang. Privacy-preserving genomic analysis via pso-driven federated learning on blockchain. In *2023 3rd Intelligent Cybersecurity Conference (ICSC)*, pages 17–25. IEEE, 2023.