FedCoin: A Peer-to-Peer Payment System for Federated Learning

Yuan Liu\textsuperscript{1*}, Shuai Sun\textsuperscript{1}, Zhengpeng Ai\textsuperscript{1} and Shuangfeng Zhang\textsuperscript{1}, Zelei Liu\textsuperscript{2}, Han Yu\textsuperscript{2}

\textsuperscript{1}Northeastern University, Shenyang, China
\textsuperscript{2}Nanyang Technological University, Singapore

\texttt{liuyuan@swc.neu.edu.cn, sunshuai.edu@gmail, im@aizhengpeng.cn, 1871161@stu.neu.edu.cn, zelei.liu@ntu.edu.sg, han.yu@ntu.edu.sg}

Abstract

Federated learning (FL) is an emerging collaborative machine learning method to train models on distributed datasets with privacy concerns. To properly incentivize data owners to contribute their efforts, Shapley Value (SV) is often adopted to fairly assess their contribution. However, the calculation of SV is time-consuming and computationally costly. In this paper, we propose FedCoin, a blockchain-based peer-to-peer payment system for FL to enable a feasible SV based profit distribution. In FedCoin, blockchain consensus entities calculate SVs and a new block is created based on the proof of Shapley (PoSap) protocol. It is in contrast to the popular BitCoin network where consensus entities “mine” new blocks by solving meaningless puzzles. Based on the computed SVs, a scheme for dividing the incentive payoffs among FL clients with nonrepudiation and tamper-resistance properties is proposed. Experimental results based on real-world data show that FedCoin can promote high-quality data from FL clients through accurately computing SVs with an upper bound on the computational resources required for reaching consensus. It opens opportunities for non-data owners to play a role in FL.

1 Introduction

Nowadays, many businesses generate large amount of data through usage, and rely on making informed decisions using machine learning (ML) based on such data in order to thrive. With changing regulatory scene, ML is facing increasingly difficult challenges with respect to the usage of such data. Data are often collected and owned by multiple distributed entities. They often contain sensitive or private information and is cannot be stored in a centralized server without violating privacy protection laws. In recent years, federated learning (FL) has emerged as a promising solution to these challenges [Yang et al., 2019a].

In FL, each entity trains its local model and contributes the local model parameter updates to a center server to help build a more powerful global FL model. Compared with centralized ML methods, FL not only reduces communication costs by transmitting model updates instead of raw data, but also reduce the computational costs of the server by leveraging computing power from the clients. Moreover, since local data never leave the data owners, FL improves user privacy [McMahan et al., 2017; Bonawitz et al., 2019].

From the above description, it is clear that FL clients are making significant contributions towards the FL model. Thus, in order to sustain an FL community, it is important for FL clients to be properly incentivized. For this to happen, FL clients must be treated fairly [Yu et al., 2020]. Existing FL incentive schemes generally agree that fair treatment of FL clients shall be based on a fair assessment of their contributions to the FL model [Kairouz et al., 2019]. Currently, the most widely adopted method for fair assessment of FL client contribution is that of Shapley Values (SVs) [Jia et al., 2019; Song et al., 2019].

SV is a popular notion in fairly distributing profits earned by a coalition among its contributors. It has been applied in various fields, ranging from economics, information theory, and ML. The reason for its broad application is that the SV divides the payoff with attractive properties such as fairness, individual rationality, and additivity. However, SV based distribution solution often takes exponential time to compute with a complexity of $O(n!)$ where $n$ is the number of data items. Even though the computational complexity can be reduced through approximating SV with marginal error guarantees [Jia et al., 2019], it is still computationally costly.

In order to help FL systems compute SVs to support sustainable incentive schemes, we propose a blockchain-based peer-to-peer payment system FedCoin. The Shapley value of each FL client, reflecting its contribution to the global FL model in a fair way, is calculated by the Proof of Shapley (PoSap) consensus protocol which replaces the traditional hash-based protocol in existing proof of work(PoW) based blockchain systems. All the payments are recorded in the block in an immutable manner. Under FedCoin, paying out incentives to FL clients does not need to rely on a central FL server. Based on this, FedCoin provides a decentralized payment scheme for FL, so that incentives for all participants can be delivered in third-party-free manner with nonrepudiation and tamper-resistance properties.

Extensive experiments based on real-world data show that
FedCoin is able to properly determine FL clients’ Shapley Value-based contribution to the global FL model with an upper bound on the computational resources required for reaching consensus. To the best of our knowledge, FedCoin is the first attempt to leverage blockchain technology in federated learning incentive scheme research. It opens up new opportunities for entities which has computational resources but without local data to contribute to federated learning.

2 Related Work

The incentive mechanism design is an important research direction in the field of federated learning [Kairouz et al., 2019; Yang et al., 2019a]. In [Kang et al., 2019], the contract theory is employed to improve the accuracy of model training considering the unreliable data contributors. A consortium blockchain architecture is applied to build a decentralized reputation system. In [Khan et al., 2019], a Stackelberg-game based incentive mechanism is designed to optimize the utilities of both FL clients and the FL server. These works focus on optimizing the rewards for self-interested FL clients and FL customers who pay to use the FL model. Our study is compatible with these works in terms of determining the payment budget for the FL customers.

In field of ML, SV has also been studied widely for various purpose. SV can be applied in feature selection, ranking the importance of training data, which is further applied in explaining the behavior of ML models [Li and Cui, 2019]. Since the computation complexity is $O(n^4)$, approximations of SV also attract many attentions. In [Ancona et al., 2019], a polynomial-time approximation of SV is proposed for deep neural network. Group sampling based approximation is studied in [Jia et al., 2019]. In this work, our objective is not to decrease the computational complexity, but to establish a scheme so that distributed computational resources, which are otherwise wasted, can be leveraged to help FL systems calculate SVs.

Blockchain has been widely applied in addressing the security problems in FL applications [Dillenberger et al., 2019; Kairouz et al., 2019; Yang et al., 2019a]. FLChain [Bao et al., 2019] and BlockFL [Kim et al., 2019] have been proposed to record the local model parameter updates in the temper-resistant manner. A blockchain-based FL was proposed in [Ramanan et al., 2019] so as to remove the need for an FL server. A blockchain-based trust management system was proposed in [Kang et al., 2019] to assist FL server to select reliable and high quality data owners as FL clients. These blockchain systems are used as immutable ledgers to record local gradients and aggregate them in a trusted manner. Our work will adopt the blockchain network as a computational engine and payment distribution ledger, which is the first of its kind in the current literature.

3 Preliminaries

For a typical FL scenario, we take $F_i(w) = \ell(x_i, y_i; w_t)$ as the loss of prediction on a sample $(x_i, y_i)$ with model parameters $w_t$ at the $t$-th round. The parameters $w^t$ is a $d$-dimensional vector. We assume that there are $K$ clients, and each client has a local data set $D_k$ with $n_k = |D_k|$. The overall dataset is $D = \{D_1, \ldots, D_K\}$ with $n = |D| = \sum_{k=1}^{K} n_k$. The objective function to be optimized is:

$$\min_{w \in \mathbb{R}^d} F(w) \quad \text{where} \quad F(w) = \frac{1}{n} \sum_{k=1}^{K} \sum_{i \in D_k} F_i(w) \quad (1)$$

This optimization problem is generally solved by stochastic gradient descent (SGD) [Goodfellow et al., 2016] based methods. For example, based on the current model $w_t$, the federated averaging algorithm [McMahan et al., 2017] computes the average gradient $g_k^t = \frac{1}{n_k} \sum_{i \in D_k} \nabla F_i(w_k)$ on the local data of client $k$. Each client updates its local model $w_{k+1}^t \leftarrow w_t - \eta g_k^t$, and the FL server aggregates the local models as the global FL model.

$$w_{t+1} \leftarrow A(\{w_{k+1}^t|k=1, \ldots, K\}) \quad (2)$$

where $A$ is an aggregation function.

4 FedCoin

There are two networks of participants in our system: 1) the FL network, and 2) the peer-to-peer blockchain network (Figure 1). A FL model requester or FL training task requester

![Figure 1: Overview of the Proposed Model](image-url)

refers to the entities who need to train an FL network and with a budget of $V$. In the FL network, there is a centralized server, referred as FL server, in coordinating the executing of model training and receiving payment $V$ from FL model requester.

The distributed data owners, called as FL clients, participate in a collaborative training task and receive a payment $V$. Each FL client trains its local model and submits the parameter updates to the FL server. The FL server plays three roles. Firstly, it publishes a training task to FL clients with price TrainPrice. Secondly, it aggregates local updates through a secure aggregation protocol [Bonawitz et al., 2017] and earns a computation payment (ComPrice). Thirdly, it transfers a processing fee to the blockchain network to enlist its members’ help in calculating the FL model. The total payment of the task (TrainPrice+ComPrice+SupPrice) should be greater than $V$ in order to sustain payment balance without relying on external transfer of values into this system.

After each global model update, the FL server publishes a task to calculate the contribution by each FL client. The
consensus nodes in blockchain network then collaboratively calculate SVs, and the block winner receives a payment of TrainPrice+SapPrice. The winner then divides ComPrice to FL clients according to their respective SVs by creating transactions in the blockchain. In our current design, we only reward clients with positive contributions, but refrain from penalizing clients with negative contributions. All the transactions are recorded in the new block and further updated to the chain.

Therefore, the connection between FL network and blockchain network is a special type of task. A task includes the received local update set \( W = \{w_k | k = 1, \ldots, K\} \), the aggregation function \( \mathcal{A} \), the loss function \( \mathcal{F}(w) \), and values SapPrice and TrainPrice for each update round. Note that SapPrice and TrainPrice decreases as the number of training rounds increases, and the total payment for training can be divided among the rounds equally or not. Without loss of generality, the following description focuses on a single training round.

### 4.1 Shapley Value Based Blockchain Consensus

Upon receiving a Shapley value calculation task from the FL network, the miners in the blockchain network are to calculate the SV vector \( S = [s_k]_{k=1}^K \) where \( s_k \) is the SV of the client in providing \( w_k \in W \). Each miner independently calculates the SV vector following Algorithm 1. Since the objective of the mining process is to competitively calculate SV vectors so as to prove the miner’s computation power, we name the algorithm as “Proof of Shapley (PoSap)”. The input of Algorithm 1 comes from the task specifications from the FL network. The output is a new generated block.

In Algorithm 1, a miner first initializes the SV vector as an all-zero vector, and sets the calculation iteration number as 0 (Line 1-2). The SV computation continues as long as one of the following two conditions are satisfied: 1) there is no new block received; or 2) the received block fails to pass the verification which is specified in Algorithm 2 (Line 3).

The SV calculation process is described in Line 4 to Line 13. The miner initializes a temporary SV vector \( S_t \) to record the calculated value in this iteration (Line 4). Then, the miner randomly generates a rank of the \( K \) FL clients (Line 5). According to the rank, an SV of the first entity is calculated as in Line 6, which is the contribution of the entity to the loss function (Line 6)). For the next entity \( i \), the Shapley value is calculated as its marginal contribution (Line 7-10). \( S \) is updated by averaging all the previous iterations and the current \( S_t \) (Line 11). The iteration time is then incremented by 1 (Line 12). Then, the entity broadcasts \( S \) and time (Line 13).

Whenever a miner receives \( S \) and time, the miner calculates the average results \( \overline{S} \) of all the received \( S \) (Line 16). Then, the miner calculates the \( P \)-distance between its own \( S \) and \( \overline{S} \). When the distance is no greater than the mining difficulty \( D \), the miner becomes the winner and generates a new block \( Blk \) (Line 18). The difficulty \( D \) is adapted dynamically as explained in Section 4.2. The illustration of the Shapley based verification is shown in Figure 2. The new block is then appended to the current longest chain.

Whenever a miner receives a new block \( Blk \), the miner verifies this block according to Algorithm 2. Once the verification passes, the block is appended to the miner’s chain, and the mining process terminates (Line 23-28).

The structure of a block is shown in Figure 3, including block header and block body. The block header includes seven pieces of information, Table 1 presents the explanation about each header item. The block body records two types of data: (1) The task specification including all the inputs for Algorithm 1; and (2) The transactions in the blockchain network. Here, a transaction is denoted as a certain amount of FedCoins transferred from a user account to another, which is similar to that in BitCoin [Nakamoto and others, 2008]. The block winner has the privilege to create special transactions: transferring TrainPrice from its own account to the FL clients according to \( S \). The detailed design is explained in Section 4.3.

The verification procedure is described as in Algorithm 2. Three conditions must be satisfied for a block to successfully pass the verification. The first condition is \( \| S_t - \overline{S} \|_p \leq D \) which aims to verify whether the winner has generated the block with a valid SV calculation result. The second con-
Figure 2: Shapley Valued based Consensus Protocol

Figure 3: Block Structure in FedCoin

| Type          | Explanations                                                                 |
|---------------|-----------------------------------------------------------------------------|
| Block ID      | The block height                                                            |
| Winner’s ID   | Identity of the block generator                                              |
| Averaged $S$  | The calculated Shapley vector in Line 15 of Algorithm.1                     |
| Previous Hash | Hash value of previous block based on a hash function, e.g. SHA 256        |
| Winner’s $S$  | The Shapley vector calculated by the winner                                 |
| Difficulty    | The required difficulty $D$                                                  |
| Merkel Tree Root | Root of Merkel tree organized with transactions in block body               |

4.2 Dynamic Mining Difficulty

The difficulty level in mining new blocks can be adapted dynamically. There are two main factors influencing the difficulty updates: 1) the total mining power of the miners and 2) the speed of generating a block. Given the same mining power, the difficulty level should be decreased as the block generation speed increases. Given the same block generation speed, the difficulty level should be increased as the mining power increases. Difficulty update can be achieved by deploying a smart contract. For example, in BitCoin, a block is generated in every ten minutes and the difficulty level is updated in every two-week duration.

4.3 The Payment Scheme

With the FedCoin system in place, an FL model requester starts by depositing $V$ FedCoins in the FL server. The value of $V$ shall be no greater than the value of the FL model for the requester. To divide $V$ among FL clients, blockchain miners, and the FL server, all the entities should register a transaction account. The value of $V$ is then divided into three parts.

- **TrainPrice**: payments to the FL clients;
- **ComPrice**: payment to the FL server for processing the model aggregation;
- **SapPrice**: payments to the blockchain network miners for calculating the Shapley value of each client.

The division can be determined by a pre-agreed smart contract. For example, the division contract could specify that TrainPrice:ComPrice:SapPrice=7:1:2. Then, TrainPrice=0.7$V$, ComPrice=0.1$V$, and SapPrice=0.2$V$. The specific payment scheme is shown in Algorithm 3.

In Algorithm 3, a model training task is successfully accepted by the FL server whenever the server receives payment $V$ from the FL model requester. The payment of $V$ is confirmed when the transfer transaction (requester $\rightarrow$ server) is recorded in the blockchain. The server then calculates TrainPrice and SapPrice and leaving ComPrice=$V$-TrainPrice-SapPrice as its own payment for processing the task (Line 2). The training task is then published to FL clients with

Algorithm 2: VerifyBlock (new Blk)

**Input:** Blk: Received new block; $ar{S}_i$: Local average of received Shapley Value; $D$: Difficulty in Mining;

**Output:** ValuationResult: True OR False

1. $S_i = Blk.S; \bar{S}_i = Blk.S$;
2. if $\|\bar{S}_i - \bar{S}\|_p \leq D$ then
3.   if $\|S - \bar{S}\|_p \leq D$ then
4.     if $\bar{S}_i > longest chain length$ then
5.       return ValuationResult=true;
6.     else
7.       return ValuationResult=false;
8.   end
9. else
10. return ValuationResult=false;
11. end
12. end
13. else
14. return ValuationResult=false;
15. end
16. end
17. end
Algorithm 3: The Payment Scheme in FedCoin

Input: $V$: The value paid by a model requester; $S$: The final aggregated Shapley Value;
Output: An allocation of $V$

1. while FL server receives $V$ from a model requester do
2.   Calculate TrainPrice and SapPrice;
3.   Publish training task with price TrainPrice;
4.   if The model is well trained then
5.     Publish a Shapley task to blockchain network with price SapPrice;
6. end
7. end
8. while a new block is mined do
9.   FL server transfers TrainPrice+SapPrice to the block winner; for each FL client $i$ do
10.   if $S_i > 0$ then
11.     $p_i = \frac{S_i}{\sum_{j \geq 0} S_j} \cdot $TrainPrice;
12.     block winner transfers $p_i$ to $i$;
13. end
14. end
15. end

price TrainPrice (Line 3). When the training task is completed, the server then publishes a SV calculation task to the blockchain network with price SapPrice (Line 4-6). As the blockchain network completes the task by successfully mining a new block, the server creates a transaction to transfer TrainPrice+SapPrice to the block winner. The block winner creates the transactions in dividing TrainPrice to the FL clients with positive Shapley value. All the transactions as well as unconfirmed transactions are stored in the new block.

5 Analysis

Under FedCoin, the decentralized payment scheme is reliable based on the security of proposed PoSap consensus protocol. Each miner who successfully calculates the sufficiently converging SV is allowed to record a set of transactions and receive payment from the FL server. The more mining power (i.e., resources) a miner applies, the higher its chances to become a block winner. PoSap provides incentives for miners to contribute their resources to the system, and is essential to the decentralized nature of the proposed payment system.

The security of PoSap is also similar to that of the BitCoin system. Empirical evidence shows that Bitcoin miners may form pools in order to decrease the variance among their incomes. Within such pools, all members contribute to the solution of each cryptopuzzle, and share the rewards proportionally to their contributions. Ideally, a feasible payment system should be designed to resist the formation of large mining pools. Bitcoin system has been shown to be vulnerable when a mining pool attracts more than 50% of the miners. Similarly, the proposed system can also only resist up to 50% of the miners colluding.

Next we discuss how our system fares against the selfish mining strategy [Eyal and Sirer, 2018].

Observation 1 When the FL server processes FL training model requests sequentially, it is not rational for colluders to follow the selfish mining strategy.

According to Algorithm 3, each public block winner is paid by the FL server before creating a new block containing the block reward payment transactions. When the training task is processed one by one, if a selfish miner becomes the winner but does not publish this result immediately, it cannot receive the block rewards. Meanwhile, the selfish miner cannot mine the next block without publishing its private block since the next SV task must wait for the completion of payment in the current block in the setting of sequentially training models.

Observation 2 When the FL server processes FL training model requests in parallel, and all the miners have the same block propagation delay to the FL server; the expected revenue for selfish miner is greater than that for honest miners when the selfish pool attracts more than 25% of the total mining power in the blockchain network.

If the tasks are published in parallel, a selfish miner can reserve a block and continue to mine the next block. The state transition and revenue analysis is same as that in [Eyal and Sirer, 2018], resulting the condition of the threshold of selfish pool’s mining power to be 1/4 under the condition of the same propagation delay to the FL server. Thus, processing FL training model requests in parallel under the proposed scheme is not recommended.

6 Experimental Evaluation

To evaluate the proposed FedCoin, we set up a blockchain environment and design a federated learning task based on a real-world dataset. The objective of the experiments is to verify whether FedCoin can promote high quality data from distributed clients and whether the computational cost of PoSap is feasible.

6.1 Experiment Settings

We design our experiment based on a well-known digit classification dataset - MNIST with 70,000 images and a widely used software environment - TensorFlow - to perform federated digit classification tasks.

We set up a FL server and 100 FL clients. The MNIST dataset is divided among the clients such that their data quality vary. We set there are 10 groups of 10 clients each. Clients from each group own datasets which belong to one of the 10 preset quality levels. We refer to this quality level as a client’s type. Each client of type $T_j$ is randomly assigned a training set following a uniform distribution over the $10 - j$ class labels ($j = 0, 1, \ldots, 9$) as its local dataset. The FL training model in our experiments is classical neural network.

We adopt the popular FedAvg as the FL aggregation function, which averages the collected local model parameters to derive the global FL model parameters. Each local client trains the model for 20 iterations.

The consensus nodes are generated based on Docker. Each consensus node can independently communicate with each other by sending messages, performing Shapley value calculation tasks following PoSap, and verifying blocks. The total
computational power is equal to that of our simulation platform (CPU Inter i7-7700, GPU 2G, RAM 8g, ROM 1t, SSD 256M). We set \( p = 2 \) (Euler distance) in PoSap for comparing mining difficulty.

### 6.2 Data Quality Evaluation

We adopt Earth Mover’s Distance (EMD) as a metric to measure the data quality from the perspective of data reliability in this experiment [Zhao et al., 2018]. EMD captures the distance for a client’s training data distribution compared to a given distribution. Here, we use the distribution of the whole MINIST dataset as the comparison benchmark. A high EMD value for a given dataset indicates that the dataset is of low quality. The data quality measured by EMD for each client type is shown in Table 2. We can observe that the data quality linearly decreases from \( T_0 \) to \( T_9 \).

| Data Type | \( T_0 \) | \( T_1 \) | \( T_2 \) | \( T_3 \) | \( T_4 \) |
|-----------|--------|--------|--------|--------|--------|
| Quality (EMD) | 0.00 | 0.02 | 0.04 | 0.06 | 0.08 |

### 6.3 Results and Discussion

The average Shapley values calculated by FedCoin for the clients belonging to each type are shown in Figure 4. The results in Figure 4 show that the proposed FedCoin approach can accurately compute clients’ Shapley Values which are critical to incentive mechanisms designed for federated learning. We can also observe that the computed Shapley values of client type \( T_0 \) is higher than all the other clients with lower data quality. The Shapley values decrease with the quality level decays from \( T_1 \) to \( T_9 \). Moreover, only the values for type \( T_0 \) to \( T_4 \) are positive, indicating that only half of the clients can positively contribute model accuracy improvement. It also shows that our PoSap can effectively promote high quality data in collaboratively FL application scenario. The negative Shapley values for type \( T_5 \) to \( T_9 \) means the clients can mislead the model training.

To empirically explain why there is negative Shapley values, we train the same model with clients from each type separately. The testing accuracy of the model trained by 10 clients in each type is presented in Figure 5. It has been found that the model accuracy based data from clients with negative Shapley values is indeed lower than the accuracy achieved in the FL model based on the 100 clients. In our system, we will not reward the clients with negative Shapley values, discouraging these clients in misleading the model training.

The block generation time with different mining difficulty values is shown in Figure 6. It is measured in the unit of Shapley calculation iterations, each of which executing Lines 3 to 14 in Algorithm 1 once. The \( x \)-axis contains the difficulty \( D \) measured in the unit of \( log(D) \) (e.g., \( x = 2 \) means \( D = 1e-2 \)). The \( y \)-axis contains the block generation time as the mining difficulty increases. The block generation time converges at about 2,500 Shapley calculation iterations. This shows that the computational cost for the consensus nodes is upper-bounded. In other words, the computational cost of PoSap is feasible.

### 7 Conclusions

In this paper, we propose FedCoin - a blockchain-based payment system to enable a federated learning system. It can mobilize free computational resources in the community to perform costly computing tasks required by FL incentive schemes. The Shapley value of each FL client, reflecting its contribution to the global FL model in a fair way, is calculated by the proof of Shapley (PoSap) consensus protocol. The proposed PoSap replaces the traditional hash-based protocol in existing Bitcoin based blockchain systems. All the payments are recorded in the block in an immutable manner. Under FedCoin, paying out incentives to FL clients does not need to rely on a central FL server.

Experimental results show that FedCoin is able to properly determine FL clients’ Shapley Value-based contribution to the global FL model with an upper bound on the computational resource required for reaching consensus. To the best of our knowledge, FedCoin is the first attempt to leverage blockchain technology in federated learning incentive scheme research. Thereby, it opens up new opportunities for non-data owners to contribute to the development of the FL ecosystem [Yang et al., 2019b].
References

[Ancona et al., 2019] Marco Ancona, Cengiz Öztireli, and Markus H. Gross. Explaining deep neural networks with a polynomial time algorithm for shapley value approximation. In Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, pages 272–281, 2019.

[Bao et al., 2019] Xianglin Bao, Cheng Su, Yan Xiong, Wenchao Huang, and Yifei Hu. Flchain: A blockchain for auditable federated learning with trust and incentive. In Proceedings of 5th International Conference on Big Data Computing and Communications, BIGCOM, pages 151–159, 2019.

[Bonawitz et al., 2017] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In Proceedings of ACM SIGSAC Conference on Computer and Communications Security CCS, pages 1175–1191, 2017.

[Bonawitz et al., 2019] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloé Kiddon, Jakub Konecný, Stefano Mazzocchi, H. Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. Towards federated learning at scale: System design. CoRR, abs/1902.01046, 2019.

[Dillenberger et al., 2019] Donna Dillenberger, Petr Novotny, Qi Zhang, Praveen Jayachandran, Himanshu Gupta, Sameep Mehta, Sandeep Hans, Supriyo Chakraborty, Matthew Walli, John Thomas, et al. Blockchain analytics and artificial intelligence. IBM Journal of Research and Development, 2019.

[Eyal and Sirer, 2018] Ittay Eyal and Emin Gün Sirer. Majority is not enough: bitcoin mining is vulnerable. Communications of the ACM, 61(7):95–102, 2018.

[Goodfellow et al., 2016] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.

[Jia et al., 2019] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J. Spanos. Towards efficient data valuation based on the shapley value. In Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics AISTATS 2019, pages 1167–1176, 2019.

[Kairouz et al., 2019] Peter Kairouz, H. Brendan McMahan, Brendan Avent, et al. Advances and open problems in federated learning. CoRR, abs/1912.04977, 2019.

[Kang et al., 2019] Jiawen Kang, Zehui Xiong, Dusit Niyato, Shengli Xie, and Junshan Zhang. Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory. IEEE Internet of Things Journal, 6(6):10700–10714, 2019.

[Khan et al., 2019] Latif U. Khan, Nguyen H. Tran, Shashi Raj Pandey, Walid Saad, Zhu Han, Minh N. H. Nguyen, and Choong Seon Hong. Federated learning for edge networks: Resource optimization and incentive mechanism. CoRR, abs/1911.05642, 2019.

[Kim et al., 2019] Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Blockchained on-device federated learning. IEEE Communications Letters, 2019.

[Li and Cui, 2019] Yadong Li and Xin Cui. Shapley interpretation and activation in neural networks. CoRR, abs/1909.06143, 2019.

[McMahan et al., 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arca. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics AISTATS, pages 1273–1282, 2017.

[Nakamoto and others, 2008] Satoshi Nakamoto et al. Bitcoin: A peer-to-peer electronic cash system. Working Paper, 2008.

[Ramanan et al., 2019] Paritosh Ramanan, Kiyoshi Nakayama, and Rtnesh Sharma. Baffle: Blockchain based aggregator free federated learning. arXiv preprint arXiv:1909.07452, 2019.

[Song et al., 2019] Tianshu Song, Yongxin Tong, and Shuyue Wei. Profit allocation for federated learning. In Proceedings of the 2019 IEEE International Conference on Big Data (IEEE BigData’19), 2019.

[Yang et al., 2019a] Qi Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology (TIST), 10(2):12, 2019.

[Yang et al., 2019b] Qi Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, and Han Yu. Federated Learning. Morgan & Claypool Publishers, 2019.

[Yu et al., 2020] Han Yu, Zelei Liu, Yang Liu, Tianjian Chen, Mingshu Cong, Dusit Niyato, and Yang Qiang. A fairness-aware incentive scheme for federated learning. In Proceedings of the 3rd AAAI/ACM Conference on Artificial Intelligence, Ethics, and Society (AIES-20), 2020.

[Zhao et al., 2018] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. CoRR, abs/1806.00582, 2018.