筛选chain上的节点，并将模型存储到IPFS中，细节不是很清楚，也没有代码，有点水

FL-MAB: Client Selection and Monetization for Blockchain-Based Federated Learning

Zahra Batool
École de technologie supérieure
Montréal, Québec, Canada
zahra.batool.1@ens.etsmtl.ca

kaiwen Zhang
École de technologie supérieure
Montréal, Québec, Canada
kaiwen.zhang@etsmtl.ca

Matthew Toews
École de technologie supérieure
Montréal, Québec, Canada
matthew.toews@etsmtl.ca

ABSTRACT

Federated Learning (FL) is a promising solution for training using data collected from heterogeneous sources (e.g., mobile devices) while avoiding the transmission of large amounts of raw data and preserving privacy. Current FL approaches operate in an iterative manner by selecting a subset of participants each round, asking them to training using their latest local data over the most recent version of the global model, before collecting these local model updates and aggregating them to form the next iteration of the global model, and so forth until convergence is reached. Unfortunately, existing FL approaches typically select randomly the set of clients to use each round, which can negatively impact the quality of the model trained, as well the training round time due to the straggler problem. Moreover, clients, especially mobile devices with limited resources, should be incentivized to participate as federated learning is essentially a form of crowdsourcing for AI which requires monetization. We argue that the integration of blockchain and smart contract technologies to FL can solve the two aforementioned issues. In this paper, we present FL-MAB (FL- Multi-Auction using Blockchain), a client selection mechanism for FL operating in a smart contract which rewards clients for their participation using cryptocurrencies. FL-MAB employs a multidimensional auction mechanism for selecting users based on the compute and network resources offered by each client, as well as the quality of their local data. This auction is realized in a reliable and auditable manner through a smart contract. This allows FL-MAB to measure the relative contribution of each client by calculating a Shapley value, and allocating rewards accordingly. We have implemented FL-MAB using Solidity and tested on the Ethereum blockchain with various popular datasets. Our results show that FL-MAB outperforms existing baseline schemes by improving accuracy and reducing the no. of FL rounds.

CCS CONCEPTS

• **Security and privacy** → **Privacy-preserving protocols**; • **Computer systems organization** → **Peer-to-peer architectures**; • **Computing methodologies** → **Artificial intelligence**;

KEYWORDS

Federated Learning, Blockchain, Smart Contracts, Auction, Client Selection, Monetization

ACM Reference Format:
Zahra Batool, kaiwen Zhang, and Matthew Toews. 2022. FL-MAB: Client Selection and Monetization for Blockchain-Based Federated Learning, . In *The 37th ACM/SIGAPP Symposium on Applied Computing (SAC '22), April 25–29, 2022, Virtual Event, .* ACM, New York, NY, USA, Article 4, 9 pages. https://doi.org/10.1145/3477314.3507050

SAC '22, April 25–29, 2022, Virtual Event,

ACM ISBN 978-1-4503-8713-2/22/04…$15.00
https://doi.org/10.1145/3477314.3507050

1 INTRODUCTION

Federated Learning (FL) emerged from Machine Learning, which overcome the privacy issue by eliminating need of centralized storage and enables a shared model with distributed data to be collaboratively trained [9]. Although FL brings new application on-device learning scenarios, but it still suffers from many challenges in mobile edge computing (MEC) environment [1, 9, 10, 15]. In traditional FL setting, the client selection is random for training in each FL round [3, 9, 16]. However, In MEC scenario, many competent participants will likely remain idle even though they can contribute more data and are willing to take part in FL training [14]. The randomly selected contributors in FL process might have heterogeneous resources, and the server has no information about the clients availability i.e., battery level, network condition and their device is idle for training or not [4]. Therefore, during the model training process, any client within a network may go offline. This may produce a significant number of straggler clients in FL MEC environment. Consequently, slowing down the overall process which will decrease the model accuracy as clients are not able to share their model on time. Moreover, to motivate resource-rich clients to participate, there is a need to develop fair monetization mechanism based on their contribution.

Recently, many studies have emerged that focuses on addressing best possible client selection problem and to motivate them through compensation in traditional FL [6, 16, 17, 20, 21]. However, they relied on a centralized server for communication which possibly could lead to an inappropriate global model because the server can be biased and malfunctioning. Moreover, there are also some malicious clients who intentionally want to jeopardize the FL ecosystem by intimidating global model through attacks [4]. Eventually, by providing wrong data it makes the global model less accurate. To overcome these limitations in FL system, it is imperative to develop a secure, privacy-preserving FL framework that also provide reward to users according to their contributions and penalize negative behavior.

With the inception of blockchain technologies, it is now possible to eliminate the trusted third party. Moreover, a blockchain

system is a distributed tamper-proof database which provides auditability and accountability [7]. Therefore, a blockchain-backed federated learning network promotes the federation of more devices by providing highly fault-tolerant system using decentralization, immutability, transparency, efficiency, security and anonymity that helps in making secure FL System [2, 7, 13].

We present multi-dimensional auction for client selection with the best resources to participate in FL model training. The client will specify their resources i.e., data size, bandwidth, internet connectivity, and corresponding reward while submitting their bid. Therefore, this study aims to build a smarter FL model by developing a fair and monetizable client selection protocol using blockchain based auction model. The integration of blockchain provides integrity, non-repudiation and helps to defend against malicious updates from both clients and server[4]. Furthermore, it provides a transparent platform to the clients who want to contribute in global model training with more data and better resources and encourage them by providing reward in form of cryptocurrency.

The main contributions of this paper are as follows:

(1) A multi-dimensional auction based Federated Learning protocol is proposed to select competent clients that incorporates blockchain for authenticity and traceability. (Section 5.1)
(2) We formulated smart contract-based based secure and fair monetization by calculating client's contribution in every round of FL using Shapley values. The system will reward clients based on their contribution in the form of FL tokens. (Section 5.3)
(3) We provide a reference implementation of our solution written in Solidity deployed over Ethereum, and carried out an analysis to evaluate the performance and efficiency of our client selection and monetization protocols compared to the baseline solutions. (Section 6)

To the best of our knowledge, this is the first paper which proposes a solution to orchestrate a Federated Learning process while providing both client selection and fair monetization using smart contracts.

The rest of the paper is organized as follows: Section 2 presents the literature review of studies that have explored federated learning and blockchain based federated learning to study client's selection problem and incentive mechanism. The traditional Federated Learning, and Shapley value is introduced in Section 3. This is followed by Section 4, which present an overview of a proposed solution, preliminaries and operation flow. Section 5 explains the proposed auction and rewarding mechanism. Moreover, this section also presents a walk-through example. Section 6 will presents the results and their comparison with existing solution. Finally, Section 7 concludes the findings of this study.

2 RELATED WORK

This section will provide a brief review of studies focusing on client selection problem in MEC and reward mechanism using traditional federated learning [6, 16, 17, 21]. Moreover, we also discuss federated learning studies which leverages blockchain for rewarding and selecting clients [2, 5, 7, 11, 13, 22].

2.1 Client Selection

Federated Learning is facing challenges for heterogeneous resource client selection, data security, and incentivizing participants in MEC environment since it gained attention in 2016. A client selection protocol FedCS have been proposed by Nishio & Yonetani. They employed greedy algorithm for selecting clients iteratively that take the least time for model update and upload [16].

A hybrid-FL method as an extension of FedCS, which gathers data from few clients with their consent and updates the model along with the set of clients to train model with their data, are presented by Yoshida et al. [20]. The researchers employed heuristic algorithms for client selection. However, FL supposed to avoid data sharing which is still requesting by the server.

Zeng et al., employed the multi-dimensional procurement auction model for selecting and incentivizing multiple eligible devices for FL model training [21]. The authors advocated the expected utility theory for providing guidance to a server to get expected resources. However, a server may involve in nepotism that implies to endorse its peers by selecting and rewarding them.

In the aforementioned studies, clients and server are assumed to be trustworthy, but in real world scenario, clients can disobey the contract and server can be biased and malicious [4]. Although some studies have considered the clients selection and heterogeneous resource issue, they also ignored the reliability of clients. But they are not considering all resources while selection of clients which hinder the best clients to contribute positively. Moreover, these systems are highly prone to different types of attacks, e.g., collusion, backdoor, false name, and free rider attacks [12]. The mechanism for rewarding user according to their contribution is not considered in previous works on client selection.

2.2 Incentive Mechanisms for Clients

An incentive mechanism using contract theory, to encourage high quality data owners in FL, has been presented by Kang et al [6]. The participants with more reliable data, higher accuracy and more resource contributions can obtain greater rewards.

Song et al., proposed an efficient method for calculating contribution, which requires less computation than Shapley Value and it works by reconstructing the approximate models on different subsets of data during the training process[17].

Due to lack of secure monetization mechanism resource rich clients are demotivated to participate in collaborative training [4]. Contrary to these previous works, our proposed solution FL-MAB provides transparency and reliability by executing client selection and monetization on a blockchain using smart contracts.

2.3 Security in Federated Learning

There is a risk of model poisoning when collaboratively training a model in a group of untrusted participants. Furthermore, FL security will be at risk if multiple trainers halt training or compromise the global model by delivering malformed local models. There exist several studies which tried to address the above challenges in federated learning by integrating blockchain technology [2, 5, 7, 11, 13, 19, 22].

Kang et al., presented a blockchain-based client selection and reward mechanism in FL using a game theoretic approach (contract

theory) [5]. The reputation mechanism works by selecting reliable clients to overcome malicious model updates.

Another study proposed a protocol using the EOS blockchain to establish data security and fair reward in FL [13]. The protocol is intended to track user updates and to reward them for the data they have used in computation.

Bao et al. proposed blockchain based federated learning framework which provides incentive mechanism and misbehavior detection [2]. However, the framework has no client selection mechanism and the global model is exposed to unauthorized devices for attacks.

Liu et al. introduced a blockchain-based payment system called FedCoin that does not rely on a central FL server to reward FL clients [11]. The researchers used a Shapley value (SV) to measure the contribution of each client using consensus nodes in the blockchain network after each update of global model to reward them.

Another blockchain-based federated learning system is proposed to assist IoT device manufacturers to predict customer's behavior by collaboratively training an ML model by Zhao et al [22]. However, they are assuming that the blockchain is permissioned and their participants are honest. This system design is not suitable for public blockchain and there are more chances of participants to be malicious in the public blockchain. Building upon these earlier works, we formulated a novel multi-dimensional auction-based context aware client selection and rewarding mechanism.

3 BACKGROUND

In this section, we briefly introduce different concepts required to understand this paper.

3.1 Federated Learning

FL is a decentralized machine learning paradigm which aggregate global machine learning model from distributed resource-constraints clients data [14].

$$\sum_{k=1}^{k} \left(\frac{n^k}{n} W_{t+1}^k \right) \tag{1}$$

Let suppose k data owners are deployed to train their local W_k in form of weights. The server learns by receiving updates W_k from clients in form of weights and aggregating those updates in to one global model. In first round of FL, randomly selected clients will receive an initial global model parameter W_t to initiate the process. Every client in FL trains its local model W_k using their private data d_i and send to server to form one global model as shown in Fig. 1. FL works in multiple rounds by updating the aggregated global model after every round. Subsequently, all local updates $W_t^1, ..., W_t^k$ are combined by server to form one global parameters W_{t+1}.

3.2 Shapley Value

Shapley value proposed by Lloyd Shapley is the most popular methods for calculating reward for every participant in coalition games by investigating their contribution [17]. It provides a direction to measure client influence and fairly divide the earned reward among the FL participants, on basis of their individual contribution to the total payoff.

In our case, we have a set of clients $k = \{k_1, k_2, \ldots . k_n\}$, and S is subset of clients k and their contribution function u which depicts the contribution of every client k to our global model in every round in form of accuracy. The Shapley Value investigated the marginal contribution of what a client i add to the coalition that don't have i. The method to calculate Shapley Value is:

$$\phi_i = \frac{1}{n} \sum_{S \subseteq N \setminus \{i\}} \frac{1}{n-1_{|S|}} u(W_{S \cup i}) - u(W_S) \tag{2}$$

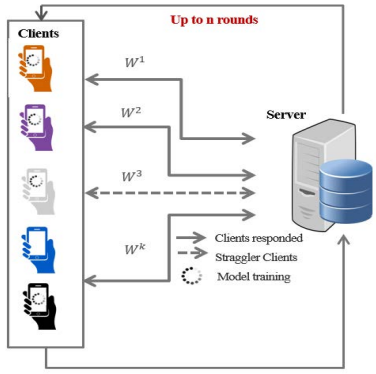


Figure 1: Federated Learning Process

4 PROPOSED SYSTEM MODEL

In this section, we elaborated the each component of our proposed blockchain-based solution i.e., Server, Clients, Blockchain network. Moreover, this section also explains our proposed system model and operation flow.

4.1 Main Components

Our blockchain based federated learning protocol consist of three entities, i.e., clients, server and blockchain network. The role of entities involved in the system is given below.

(1) *Server:* The server is responsible for initiating FL process by starting auction phase. After device selection, it will send initial global model parameters to every participant. In addition to, it will aggregate all local model updates received from clients to form one global model.
(2) *Clients:* Clients are smart devices in MEC that are agreed to use their resources for model training. At the start of every auction phase, interested client will submit bids with (resources, price, nonce and deposit amount) with an intention to participate in FL model training process.
(3) *Blockchain Network:* The blockchain network enables auditability in a FL system by storing everything on-chain. The client selection and monetization will be done through blockchain-based smart contracts. It will also provide integrity, and non-repudiation in auction mechanism. Moreover, the smart contracts act as an intermediary, enable interaction between clients and server and restrict them to be fair by eliminating the need of a third party.

4.2 System Model

In this paper, we take as an case study a MEC infrastructure where n number of mobile phones are in a FL scenario with a server O. The server and clients communicate through smart contracts with each other. In our Ethereum reference implementation, we have specific addresses for Server O and different clients $k = \{1, 2, \ldots, n\}$. These devices are deployed to train their local model W_t^k by using their private data d. These edge devices may vary in terms of their resources r i.e., data size, internet connectivity, and bandwidth, which may hinder in their local model training. To address this problem, the server will select devices with good resources through auction to join federated learning. Finally, the server will aggregate all local model received from clients to form one global model W_{t+1} and rewarded them according to their contribution. The frequently used variables are listed in Table 1.

Table 1: List of Variables

Variables	Description
N	Total no of clients
O	Server
d	Data samples
ω	Winners
S(.)	Scoring function
r	Quality vector of resources
P	Bid price
$\hat{\omega}$	Whitelisted participants
η	Nonce
ϕ	Contribution index
∂	Deposited amount
W_t	Local model
W_{t+1}^k	Global model

4.3 Overview of the Operation Flow

The proposed client selection protocol FL-MAB consist of three phases i.e., the auction, federated learning, and the reward phase as illustrated in Fig. 2.

Auction Phase: The auction is initiated to select best clients to improve the accuracy of the federated model. The clients will bid with their resources i.e., data size, internet connectivity, and bandwidth. Finally, clients with having good resources will be nominated as winners and only they are allowed to participate in FL process.

Federated Learning Phase: The FL phase is the implementation of traditional FL in which server starts FL process by sending initial parameters to clients for model training. Similarly, In FL-MAB, the server will send initial parameters to whitelisted clients instead of randomly selected clients. We implemented FL by creating N clients and their id's are mapped with Ethereum addresses.

Rewarding Phase: In the end, the contribution of every client is calculated in the global model. Finally, the clients will be incentivized based on their performance using cryptocurrency.

5 DETAILS OF EACH PHASE

In this section, we present our blockchain-based proposed multi-dimensional auction model to select resource rich clients for model training and providing them motivation to participate by giving reward. Moreover, this section also explains a walk-through example of the winner selection process. The pseudocode of the entire FL-MAB process is given in Algorithm 1.

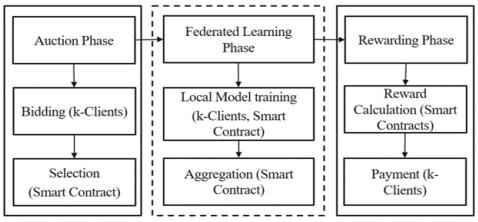


Figure 2: System Operation Flow

5.1 Auction Phase:

The server will communicate with client devices and smart contracts through an Ethereum blockchain network where bidders send bids to contract. The FL Server smart contract is deployed on the blockchain along with the Auction smart contract to initiate FL process by broadcasting that they need devices for ML model training with resources r. The FL Server smart contract initiates the *newround* function and calls the Auction smart contract to start the new round. Specifically, let $T = \{0, 1, \ldots, t\}$ denote the number of rounds of an FL process. At the start of every round t, the server will announce the scoring rule $S(r_1, \ldots, r_m, p)$, where $r = \{r_1, \ldots, r_m\}$ is the quality vector of resources, and p is the payment that the client devices expect to receive. The resources r required to include in bid is total samples of local data, bandwidth, internet connectivity, along with claiming price p and some deposited amount. The deposited amount will be collected for security so that clients cannot negate their bid. The server will return the deposited amount to clients once the bidding period is over. The data samples for every client are denoted by $d_i = \{1, 2, \ldots, n\}$. The scoring function $s(.)$ will be set as utility function for server to obtain best resource clients and defined as:

$$S(r_{i1}, r_{i2}, \ldots, r_{im}, p_i) = s(r_{i1}, r_{i2}, \ldots, r_{im}) - pi \tag{3}$$

where $\{i = 1, 2, \ldots, N\}$ is the index of clients. When a bid request with the scoring function $S(.)$ is received by every client, they will start bidding upon receiving the scoring rule. The N different bidders will submit their sealed bid $S(r_i, p_i)$ using a *CreateBid* function with nonce within the specified time duration. The sealed bid is used to protect the bid b_i from being viewed by other bidders. It will only be known by the server and the client i who submitted the bid and will be revealed after the bidding period is over.

Once the bidding period T_b is over, the client must call *RevealBid* function to reveal their bids along with the nonce to verify their validity. The server will close the bid selection process when it has received enough bids and the predefined reveal period T_r has expired. The *withdraw* function will then be called by the Auction smart contract, which will return the deposited amount to each client.

After collecting all bids, the Auction smart contract will start to determine the winners using *calculatescore* function. The clients

with higher score can be determined by Auction smart contract based on their credibility or ranking score calculated using Eq. 3. In our solution, we extended the classic multi-dimensional auction to multiple winners [21]. Among top scorer, only 50% of the bidders will be selected as winners ω among the bidders using the *GetWinners* function. The winner calculation is performed on-chain and blockchain will send the winner list to the server and clients.

At the end of Auction phase, the FL Server smart contract will whitelist winning participants so they can participate in training and submit their local model update once (per round).

5.2 Federated Learning Phase:

In FL phase, a python script for FL is used to connect with blockchain using web3.py, which enable interaction with the blockchain network. This will get a list of all winners using get *ListOfAllbiddersRevealed* function and get winner score through *getWinnerScore* function to start FL process. After receiving initial parameters from the server, each client $k_i \in k$ will start training their local model W_i^k by using their private data d_i and resources r. In our reference implementation, the whitelisted clients will train and push their model hash W_i^k to IPFS (off-chain storage) by calling *AddModel* function of the FL Server smart contract. Each client has their model stored on an off-chain storage using the *AddToIpfs* function and hash of their model on-chain. This will reduce cost overhead and helps in maintaining the history of model off-chain [8]. When selected clients will finish local training, the global aggregation will then be performed by the server. The server will retrieve all local models from IPFS by comparing with their on-chain hashes to perform aggregation for global model W_{t+1}^k using Eq. 1.

5.3 Rewarding Phase:

The contribution index of every client is calculated in the global model to reward them accordingly using coalition game-based method i.e., Shapley value. In every round, the power set $S \subseteq k = \{1, 2, \ldots, n\}$ for every client is computed and then we compute the local model for every combination of clients $W_k | S \subseteq k$. For example, suppose we have to train our model $W_{1,2}$ for a subset 1, 2 using the data from client 1 and client 2. The server will approximate the global model $W_{ki} = \{1, 2, \ldots, n\}$ for every subset $S \subseteq k$ using the technique in [17]. Finally, we calculate a marginal contribution of every client using Shapley value.

For rewarding clients, the bid price that every client mentioned while bidding in auction phase is considered by multiplying the bid price with a contribution index to reward them using cryptocurrency. We employ cryptocurrency tokens to reward clients. The *RewardDistribution* function executed by server after each round and takes the addresses of clients, reward amount and round number which they are rewarded for. The client can claim reward anytime by calling *collectReward* function. After the end of the every round, the server calls the Auction smart contract to re-deploy it for starting a new round.

$$Rewardamount = P_i * \phi_i \tag{4}$$

We also implemented alternative methods to calculate the contribution of every client in the global model, i.e., exact Shapley, and influence function. The comparison of these implemented methods

Algorithm 1 FL-MAB Algorithm

Input: nodes set $K = \{1, 2, \ldots, n\}$, local data size $d = \{d_1, \ldots, d_n\}$
Output: global model W_{t+1}, ϕ_i

Phase 1: Auction
for $t = 1, 2, \ldots, T$ **do**
  The aggregator sends scoring rule S(q, p) to all the nodes;
  for *node* $i \in K$ in *parallel* **do**
    Node k submit sealed bid $(r_i, p_i, \partial, \eta)$ to the server
  end for
  for *node* $i \in k$ in *parallel* **do**
    Node k reveal bid $(r_i, p_i, \partial, \eta)$ to the server
  end for
  The server nominates winners through solving Eq 3 and sorts all the scores in descending order.
  $i \in k \leftarrow withdraw()$
  $Server \leftarrow Whitelist()$
Phase 2: Federated Learning
  The server sends initial global model parameter W_t to nodes.
  for *node* $i \in k$ in *parallel* **do**
    Node k train W_t with d_i
    Node k send W_{t+1}^k to the server
  end for
  The server computes global model W_{t+1} according to Eq 1
Phase 3: Calculating client's contribution and reward distribution
  Calculate possible combination of nodes set $k = \{1, 2, \ldots, n\}$.
  The server sends initial global model parameter W_k
  for *eachS* $\subseteq k$ **do**
    $W_S \leftarrow \frac{|d_i|}{\sum_{i \in S} |d_i|} \sum_{i \in S} W_t$
  end for
  for *iteration* $= 1, 2, \ldots$ **do**
    $\phi_i = C \cdot \sum_{S \subseteq k \setminus \{i\}} \frac{u(\tilde{W}_{S \cup i}) - u(\tilde{W}_S)}{n-1_{|S|}}$
  end for
end for
return $W^{(R)}$ and ϕ_i

没有任何chain的描述，也没有IPFS的描述，说明不清

in term of cost and time is discussed in (Section 6.5). We need to train additional coalition models to compute Shapley value using the exact method which is an expensive solution. Therefore, for calculating coalition model, we used heuristic solution similar to [17], which is capable of approximating coalition model from client local models. To compare our approach, we have also implemented influence function and exact Shapley method for calculating contribution [17, 18]. The performance of the implemented methods is evaluated by measuring the execution time and calculating the Euclidean distance. The reward is distributed among clients by multiplying bid price with a Shapley based contribution index. Moreover, influence function is implemented to evaluate the fairness and cost for calculating a client's contribution. To calculate the influence of every client on the global model, we need to retrain the model by eliminating client for which we need to calculate contribution and

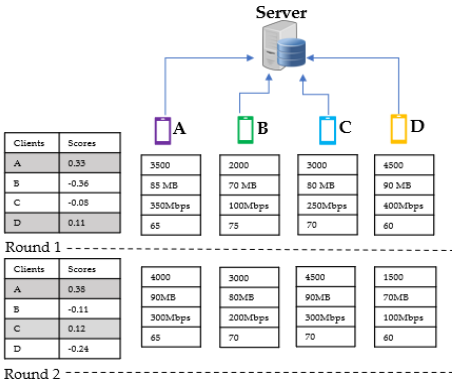


Figure 3: Walk-Through Example for Four Clients

measure how much the prediction of retrained model changes.

$$Influence_i = \frac{1}{n}\sum_{j=1}^{n} |\hat{W}_n - \hat{W}_n^{-i}| \tag{5}$$

where n is the size of dataset, W_n is the global model trained on all data, and $W_{S\{i\}}$ is global model trained by eliminating the ith client.

Let the contribution index obtained from native Shapley be denoted as $< \phi^* = \phi_1^*, \phi_2^* \ldots, \phi_n^* >$ and by the heuristic algorithms$< \phi = \phi_1, \phi_2 \ldots, \phi_n >$. The Euclidean distance can be calculated by following equation:

$$d = \sqrt{\sum_{i=1}^{n} \left(\phi_i^* - \phi_i\right)^2} \tag{6}$$

5.4 A Walk-Through Example

Let consider an example with four clients k= {A, B, C, D} having resources, i.e. data size, bandwidth with a bid price as shown in Fig. 3. The data size, bandwidth, and internet connectivity are randomly divided among clients in each round over the range of [1000, 5000], [5, 100MB] and [10, 500$Mbps$] respectively. We will calculate winners by using eq no 3 i.e. S (q, p) = $\alpha_1 r_1 + \alpha_2 r_2 + \alpha_3 r_3$ − p where coefficients α_1, α_2 and α_3 value adjusted to 0.4, 0.3, 0.3 which is used to balancing different types of resources. We normalized r_1, r_3, and r_3 values using min-max normalization for simplicity, i.e.,

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{7}$$

During the auction phase for the first round of training, the four bidders will submit their bids (r_1, r_2, r_3, p) as (3500, 85Mb, 350$Mbps$, 65), (2000, 70Mb, 100$Mbps$, 70), (3000, 80Mb, 250$Mbps$, 70), (4500, 90Mb, 400$Mbps$, 60). When bidding and revealing period are over, the server starts calculating winners by computing scores. The server selects the top 50% of the bidders as winners $\omega = \{A, D\}$ with scores (0.33, 0.11) and sort them in descending order in an array. Then the server sends initial parameters to these three winners for start local model training. After completion of local model training, the server aggregates all local models into one global model. Finally, this round is completed after paying every client according to their contribution.

In the second round, there may be new client devices joining to participate in collaborative learning. Furthermore, existing nodes may not have same resources as in round 1 due to their dynamic nature. Suppose the bids for round 2 are (4000, 90Mb, 300$Mbps$, 65), (3000, 80Mb, 200$Mbps$, 70), (4500, 90Mb, 300$Mbps$, 70), (1500, 70Mb, 100$Mbps$, 60). This time, the client devices $\omega = \{A, C\}$ are the winners with scores (0.38, 0.12) by following the winner criteria selection similar to the previous round. The FL phase and the reward phase will be executed after every auction phase until desired threshold to achieve model accuracy.

6 PERFORMANCE EVALUATION

In this section, we describe our experimental setup, dataset settings and distribution, and the neural network configurations we used to evaluate FL-MAB.

6.1 Setup

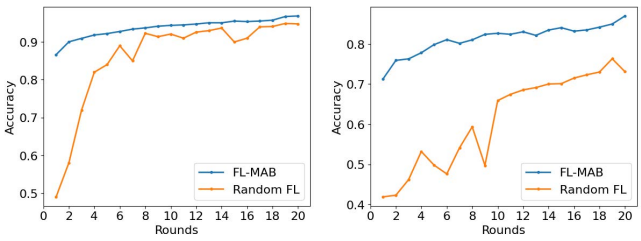
We implemented our blockchain-based federated learning environment on a system equipped with an Intel®Core^{TM} i7-8650U @ 2.0 GHz CPU, and 8 GB RAM running on Window 10. To simulate the federated learning environment, we implemented FedAvg algorithm using Python 3.6.5 with TensorFlow 2.3.1 library. The smart contract is implemented using Solidity programming language and is deployed on an Ethereum local blockchain network using Ganache. The Truffle framework is used for smart contracts development and testing. Moreover, the Python library web3.py is used to enable interaction with the Ethereum network. For off-chain storage, we used Inter Planetary File System (IPFS) through the Infura API [8]. Infura allows connectivity to the Ethereum network using HTTPS and WebSockets. We evaluated our system with one FL server and N= 40 clients.

The auction phase selects winners ω=20 i.e., 50% of total clients to participate in the collaborative learning. The total number of clients N and the winner proportion are taken in the same way as in the baseline solution [21]. The traditional federated learning baseline is named RandomFL, and our proposed solution is FL-MAB. For random selection results, we simulated bandwidth, and internet connectivity for edge nodes in our system with realistic values.

6.2 Datasets

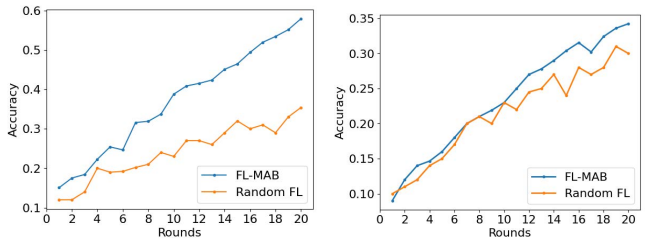
We conducted experiments on our proposed FL environment by dividing four centralized datasets (i.e., MNIST-O, F-MNIST, CIFAR-10, and HPnews) among all 40 client devices. The MNIST-O, F-MNIST dataset consists of 10 different classes of 28 x 28 pixels of 60,000 images. MNIST-O data consists of different handwritten digit images from 0 to 9. Similarly, F-MNIST dataset consists of different grayscale images of fashion objects. The CIFAR-10 consist of 32 x 32 pixel color images of 10 different classes i.e., dog, horse, plane, car etc. Moreover, we also evaluated our system on textual data i.e., Huff post news category dataset. The HP news dataset consist of 41 different classes of 200,000 news headlines from 2012 to 2018 i.e., entertainment, politics, arts, education etc.

For each dataset, the training and testing split is done using ratio 9:1. We have implemented two different Convolutional Neural Network (CNN) settings, one for MNIST-O, F-MNIST and one for CIFAR-10 similar to [21]. The news classification is done by using 2



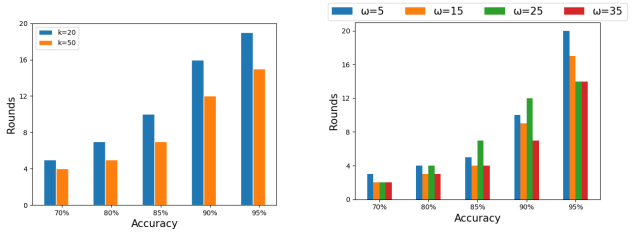
(a) MNIST-O (b) F-MNIST

Figure 4: MNIST-O and F-MNIST Accuracy



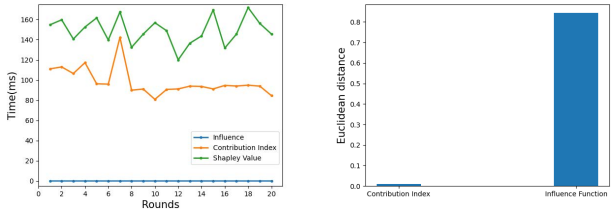
(a) CIFAR-10 (b) HP News

Figure 5: CIFAR-10 and HPnews Accuracy



(a) Number of clients K (b) Number of winners ω

Figure 6: Sensitivity Analysis



(a) Processing Time (b) Euclidean Distance

Figure 7: Comparison of Reward Distribution Methods

layers LSTM on Hp news dataset using a similar structure defined in [21]. We have evaluated our system based on Non-independent and identically distributed (Non-IID) setting. In Non-IID, client gets data of only limited classes that represents only 1 or 2 classes from all classes of data, which is a more realistic scenario in MEC as edge data is Non-IID. This theoretically means that a single client data cannot be used to infer the overall distribution.

6.3 Performance Metrics

To evaluate the performance of our proposed system, we will use following performance metrics: accuracy, number of FL rounds, throughput, time, and gas cost estimation. The higher the accuracy value the more efficient our system will be. The number of training rounds will represent how much FL rounds our model need to converge. Throughput should be increasing as more clients are introduced to demonstrate scalability. Finally, the higher time taken by function to execute on blockchain means it will consume more gas as well. Thus, the function that taking more gas and time are too expensive to execute.

6.4 Selection Algorithm Results

Accuracy Baseline Comparison: The goal of the proposed method is to improve performance of model by selecting and motivating high quality nodes to take part in a federated learning process. The performance of model training can be seen in two aspects i.e., the accuracy of the model and time taken by the model for training.

Fig. 4a and 4b depict that FL-MAB increases the convergence speed and accuracy of model for MNIST-O and F-MNIST after the 20th round of training as compared to RandomFL. However, in random selection, the accuracy jitters are prominent due to dropout clients. This is due to the fact that FL-MAB handle dropouts and straggler clients to improve accuracy and training speed of the model.

Fig. 5a and 5b illustrate the increase in model performance of CIFAR-10. In comparison with RandomFL for CIFAR-10 the accuracy is increased by 15% after the 20th round of training. Moreover, the HP News accuracy improved from 30% to 34%. The training task for HP News is quite challenging, as there are total 41 classes in News data and there is a class imbalance problem.

We can conclude that our approach outperforms the baseline and RandomFL by selecting clients with good resources in the client selection process.

Sensitivity Analysis: We investigate the performance of our system by varying the total number of clients K, and winners ω.

Fig. 6a depicts the accuracy comparison of proposed solution for $K = 20$ and $K = 50$. By increasing the number of clients in a MEC scenario, we encourage diverse data distribution nodes to participate which will improve the accuracy of the system. This is because of the competition between nodes with high quality resources which reduces convergence time and improves model performance. By comparing $K = 50$ and $K = 20$, we can see that the number of training rounds has been reduced by 20% to achieve final accuracy i.e., 95%. It can be seen that the accuracy with $K = 50$ is higher than with the $K = 20$ due to the data diversity.

Another important factor which we used to estimate performance of our system is by using different proportion of winners ω among bidders. We conducted experiments with a proportion of 10%, 30%, 50% and 70% of winners by using ω values ranging from 5-35 with $K = 50$. As shown in Fig. 6b, a low value for ω will

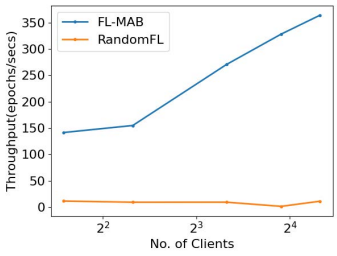


Figure 8: Client Scalability

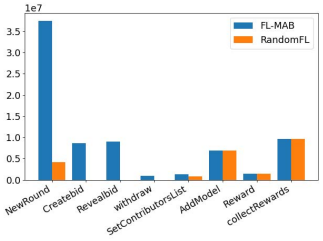


Figure 9: Gas Usage

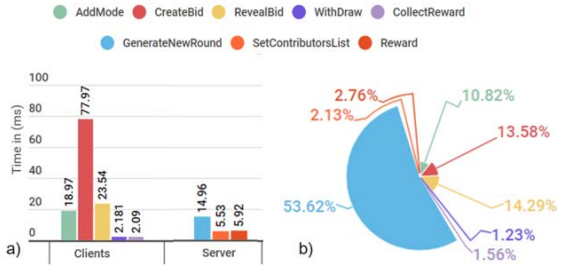


Figure 10: Smart Contract Operations - (a) Execution Time (b) Relative Gas Cost

degrade the performance of the system even if K is a high value. It takes 20 rounds of training round for ω= 5 to achieve accuracy of 95%, while 15 rounds are enough for ω = 25. Similarly, for ω = 25 and ω = 35, the system achieves the desired accuracy rapidly.

6.5 Reward Distribution

The time taken, by different algorithms, to calculate the contribution of each client in the global model is shown in Fig.7a. We can see that the Exact Shapley method is the most accurate and expensive one, since it requires re-computation of the coalition models for each subset of clients. Besides, our heuristic method, i.e., the Shapley-based contribution index is around 34% faster as it calculates the marginal contribution by approximating the coalition models. The influence function, on the other hand, takes the least amount of time compared to both of the Shapley value-based methods. This is because it does not involve expensive computations like Shapley based methods.

Next, we have examined the Euclidean distance of Contribution index, and Influence function with the Exact Shapley method. In Fig. 7b, it can be seen that the reward payment for our proposed solution (contribution index) is approximately the same as the exact Shapley. However, there is a significant difference in the reward distribution between exact Shapley and Influence function. Hence, we can conclude that the contribution index method is fairer and closer to exact Shapley, while being significantly faster to execute.

6.6 Scalability Results

We calculated the throughput of the system by iteratively increasing round t and client N. Fig. 8 shows the estimation of how much rounds achieved per second for FL-MAB and RandomFL with the number of clients. We can see that the RandomFL is fixed when increasing number of clients. On the other hand, the FL-MAB throughput is constantly increasing with a larger number of clients, which demonstrates the benefits of collaborative learning. With 20 clients, FL-MAB exhibits a 15× throughput increase over RandomFL.

6.7 Smart Contract Gas Cost comparison:

We compared the gas cost for RandomFL and FL-MAB. The graph analysis (Fig. 9) shows that RandomFL significantly reduced the gas cost for the *GenerateNewRound* function since no auctions are conducted. The *Createbid*, *RevealBid* and *withdraw* functions are not implemented in RandomFL so it incurred zero gas cost. Moreover,

Table 2: Gas Cost Analysis

Operations	Gas Usage	Fast		Standard	
		Gas price (Gwei)	Gas cost (ethers)	Gas price (Gwei)	Gas cost (ethers)
GenerateNewRound	1872553	89	0.166657217	75	0.140441475
Createbid	430076	89	0.038276764	75	0.0322557
Revealbid	452519	89	0.040274191	75	0.033938925
withdraw	47163	89	0.004197507	75	0.003537225
SetContributorsList	67458	89	0.006003762	75	0.00505935
AddModel	342720	89	0.03050208	75	0.025704
Reward	48264	89	0.006225995	75	0.005246625
collectRewards	965289	89	0.004295496	75	0.0036198

the gas usage for *Addmodel*, *Reward*, and *CollectRewards* will remain the same.

Gas Cost and Time Analysis: We investigated the estimated time and gas cost required to execute our smart contract. Fig. 10a depicts the average execution time taken by each function to execute on blockchain in FL-MAB. Most notably, the function *CreateBid* is taking 77.97 *ms*, because it requires waiting and it has to create bids for all selected clients. This is followed by *RevealBid*, which is taking 23.54 *ms* of average total time because it also requires the user to wait. This is followed by *Addmodel* (18.97 *ms*), which is used to add a local models hash to IPFS.

Fig. 10b pie chart shows that the *GenerateNewRound* function listed as the most expensive by consuming 53.62% of the total cost. This is because that *GenerateNewRound* function has to run the Auction smart contract for every new FL round.

Ethereum allows users to send transaction at the desired speed and cost through different gas prices. We calculated fast and standard gas cost estimation for FL-MAB operations as shown in Table 2. The mean confirmation time for transaction using standard gas price is 27717*sec* and for fast gas is 2757*sec*. In FL-MAB, the auction phase is time-sensitive so we need faster execution as users should submit their bid in certain time period. If users are not able to submit their bid on time, they have to wait for the next FL round to start. As a result, the submission of local model will be delayed.

7 CONCLUSION

In this paper, we have studied the client selection and monetization problem in federated learning by implementing blockchain-based

multi-dimensional auction model. Our proposed approach uses an auction to choose the best clients based on their resources. We also calculated the contribution of every clients in global model using a Shapley value-based contribution index. We implemented FL-MAB using smart contracts and evaluated on four different datasets to check the performance of our proposed system. The results show that FL-MAB reduced the training rounds by eliminating straggler clients and improved the accuracy of the model by up to 15%. Furthermore, our reward heuristic provides fair monetization while being significantly faster to execute than the baseline.

In future, we plan to extend our work by implementing FL-MAB on hardware. In addition, we will look into implementing other techniques to ensure feasibility on large scale and to reduce gas cost.

REFERENCES

[1] Mohammed Aledhari, Rehma Razzak, Reza M Parizi, and Fahad Saeed. 2020. Federated learning: A survey on enabling technologies, protocols, and applications. *IEEE Access* 8 (2020), 140699–140725.

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

[3] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, H Brendan McMahan, et al. 2019. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046* (2019).

[4] Ahmed Imteaj, Urmish Thakker, Shiqiang Wang, Jian Li, and M Hadi Amini. 2020. Federated learning for resource-constrained iot devices: Panoramas and state-of-the-art. *arXiv preprint arXiv:2002.10610* (2020).

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

[6] Jiawen Kang, Zehui Xiong, Dusit Niyato, Han Yu, Ying-Chang Liang, and Dong In Kim. 2019. Incentive design for efficient federated learning in mobile networks: A contract theory approach. In *2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*. IEEE, 1–5.

[7] Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. 2019. Blockchained on-device federated learning. *IEEE Communications Letters* 24, 6 (2019), 1279–1283.

[8] Randhir Kumar and Rakesh Tripathi. 2019. Implementation of distributed file storage and access framework using IPFS and blockchain. In *2019 Fifth International Conference on Image Information Processing (ICIIP)*. IEEE, 246–251.

[9] Qinbin Li, Zeyi Wen, and Bingsheng He. 2019. Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection. (2019).

[10] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine* 37, 3 (2020), 50–60.

[11] Yuan Liu, Zhengpeng Ai, Shuai Sun, Shuangfeng Zhang, Zelei Liu, and Han Yu. 2020. Fedcoin: A peer-to-peer payment system for federated learning. In *Federated Learning*. Springer, 125–138.

[12] Lingjuan Lyu, Han Yu, and Qiang Yang. 2020. Threats to federated learning: A survey. *arXiv preprint arXiv:2003.02133* (2020).

[13] Ismael Martinez, Sreya Francis, and Abdelhakim Senhaji Hafid. 2019. Record and reward federated learning contributions with blockchain. In *2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*. IEEE, 50–57.

[14] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.

[15] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review* (2008), 21260.

[16] Takayuki Nishio and Ryo Yonetani. 2019. Client selection for federated learning with heterogeneous resources in mobile edge. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 1–7.

[17] Tianshu Song, Yongxin Tong, and Shuyue Wei. 2019. Profit allocation for federated learning. In *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2577–2586.

[18] Guan Wang, Charlie Xiaoqian Dang, and Ziye Zhou. 2019. Measure contribution of participants in federated learning. In *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2597–2604.

[19] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 2 (2019), 1–19.

[20] Naoya Yoshida, Takayuki Nishio, Masahiro Morikura, Koji Yamamoto, and Ryo Yonetani. 2019. Hybrid-fl: Cooperative learning mechanism using non-iid data in wireless networks. *arXiv preprint arXiv:1905.07210* (2019).

[21] Rongfei Zeng, Shixun Zhang, Jiaqi Wang, and Xiaowen Chu. 2020. Fmore: An incentive scheme of multi-dimensional auction for federated learning in mec. In *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 278–288.

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