OFL-W3: A One-shot Federated Learning System on Web 3.0

Linshan Jiang
linshan@nus.edu.sg
National University of Singapore

Moming Duan
moming@nus.edu.sg
National University of Singapore

Bingsheng He
hebs@comp.nus.edu.sg
National University of Singapore

Yulin Sun
rain-forest@sjtu.edu.cn
Shanghai Jiao Tong University

Peishen Yan
peishenyan@sjtu.edu.cn
Shanghai Jiao Tong University

Yang Hua
Y.Hua@qub.ac.uk
Queen's University Belfast

Tao Song
songt333@sjtu.edu.cn
Shanghai Jiao Tong University

ABSTRACT

Federated Learning (FL) addresses the challenges posed by data silos, which arise from privacy, security regulations, and ownership concerns. Despite these barriers, FL enables these isolated data repositories to participate in collaborative learning without compromising privacy or security. Concurrently, the advancement of blockchain technology and decentralized applications (DApps) within Web 3.0 heralds a new era of transformative possibilities in web development. As such, incorporating FL into Web 3.0 paves the path for overcoming the limitations of data silos through collaborative learning. However, given the transaction speed constraints of core blockchains such as Ethereum (ETH) and the latency in smart contracts, employing one-shot FL, which minimizes client-server interactions in traditional FL to a single exchange, is considered more apt for Web 3.0 environments. This paper presents a practical one-shot FL system for Web 3.0, termed OFL-W3. OFL-W3 capitalizes on blockchain technology by utilizing smart contracts for managing transactions. Meanwhile, OFL-W3 utilizes the Inter-Planetary File System (IPFS) coupled with Flask communication, to facilitate backend server operations to use existing one-shot FL algorithms. With the integration of the incentive mechanism, OFL-W3 showcases an effective implementation of one-shot FL on Web 3.0, offering valuable insights and future directions for AI combined with Web 3.0 studies.

PVLDB Reference Format:
Linshan Jiang, Moming Duan, Bingsheng He, Yulin Sun, Peishen Yan, Yang Hua, and Tao Song. OFL-W3: A One-shot Federated Learning System on Web 3.0. PVLDB, 17(12): 4461 - 4464, 2024.
doi:10.14778/3685800.3685900

1 INTRODUCTION

Federated Learning (FL) [7] marks a pioneering shift in machine learning, enabling collaborative model training directly within data silos. This innovative approach allows for the collaborative training of models across various data silos, safeguarding the privacy of local data while simultaneously building robust global models. Concurrently, the emergence of Web 3.0 revolutionizes our digital interactions and online value exchange mechanisms. Powered by blockchain technology and decentralized applications (DApps), Web 3.0 introduces significant breakthroughs in distributed video platforms and cloud storage services. Therefore, merging FL with Web 3.0 introduces novel pathways for data silos to practically engage in the collaborative machine learning process with incentives.

Standard FL algorithm FedAvg [7] requires a multitude of communication rounds for effective global model training, leading to considerable communication overhead, increased privacy risks, and a greater demand for fault tolerance. One-shot FL approaches [5, 6, 10], which streamline client-server communication into a solitary round, offer a promising yet complex solution to mitigate these challenges with a tolerable impact on global model quality. Additionally, within the context of Web 3.0 applications, the transaction speed limitations of contemporary commercial blockchains such as Ethereum (ETH) [9], coupled with the high transaction costs (e.g., gas fees) on Web 3.0, render one-shot FL a viable option.

The practical implementation of one-shot FL on Web 3.0 encounters two significant challenges. Firstly, given that Web 3.0 research is still in its nascent stages, the fusion of Web 3.0 and FL, including the functionality and roles within this integration, remains an ambiguous issue. Secondly, considering the substantial gas fees [3] associated with transactions on ETH, it necessitates the simplification of smart contract designs. In other words, complex operations and the storage of models within smart contracts should be minimized to manage costs effectively.

To solve these challenges, in this demonstration, we present OFL-W3, a novel one-shot Federated Learning (FL) system optimized for Web 3.0. OFL-W3 categorizes data silos into two roles: model buyers, who lead the one-shot FL process and supply tokens for robust models, and model owners, who use their private data to contribute models to the one-shot FL process in exchange for tokens. To address storage and smart contract complexity challenges on Web 3.0, we leverage the Inter-Planetary File System (IPFS) [2] for efficient model sharing. Furthermore, we employ PFNM [10] as the one-shot FL algorithm and Leave-one-out as the incentive mechanism for illustration. To showcase our system, OFL-W3 includes a

Proceedings of the VLDB Endowment, Vol. 17, No. 12 ISSN 2150-8097.
doi:10.14778/3685800.3685900

distributed application (DApp) built with React for the front end and Flask for backend services, integrated with the Google Chrome browser and MetaMask wallet extension. This configuration enables model owners to participate in the FL system and receive token rewards with no prior blockchain or Web 3.0 knowledge, while model buyers can access decentralized models to build robust global models, while maintaining data privacy and security. Our contribution can be summarized as follows.

Our system offers a user-friendly DApp that allows data silos to participate in the one-shot FL learning system, either as model buyers or model owners. Designed for simplicity and ease of use, OFL-W3 enables anyone, regardless of their knowledge of blockchain or Web 3.0, to share their models or obtain high-quality ML models.

2 RELATED WORKS

One-shot Federated Learning. One-shot Federated Learning (FL) represents a cutting-edge and promising avenue of research, distinguished by its notably low communication cost. The initial exploration into one-shot FL [6] presents a method that aggregates local models into an ensemble to formulate the final global model, followed by the application of knowledge distillation utilizing public data. Researchers introduce PFNM [10], a Bayesian probabilistic framework specifically tailored for multi-layer perceptrons. Lastly, FedOV [5] ventures into tackling cases of label skew, marking another step forward in the evolution of one-shot FL approaches.

Blockchain-enabled Federated Learning. Blockchain-enabled Federated Learning (FL) has conventionally addressed the privacy and security challenges inherent in FL frameworks. For instance, Blockchain-based PPFL [1] leverages blockchain technology to trace models and prevents tampering by unauthorized individuals. BlockFlow [8] addresses concerns related to dishonest participants by employing blockchain and consensus mechanisms.

These approaches to FL for Web 3.0 have certain limitations, including their dependency on local blockchains, lack of public code, which hinders system evaluation, and potentially inaccurate estimated gas fees for contemporary commercial blockchains on Web 3.0. The absence of DApps restricts engagement to Web 3.0 specialists. Additionally, sharing models directly on the blockchain, as seen in several studies, increases numerical execution costs on smart contracts, challenging their widespread adoption. Moreover, relying on traditional FL algorithms introduces substantial overhead from multi-round communication over the blockchain.

3 SYSTEM

3.1 System Overview

As shown in Fig. 1, OFL-W3 consists of the following two entities. **Model Buyers.** Model buyers have demands for high-quality ML models. They aggregate the shared models on a one-shot FL algorithm through OFL-W3 to improve model quality.

Model Owners. Model owners can participate in model aggregation via OFL-W3, which requires sufficient incentives. Note that the models may come from the local training if the model owner also performs as the data owner, or fine-tuned/transferred from existing backbone models on their own techniques.

In our system, model buyers benefit from improved model quality via the one-shot FL paradigm, at the cost of spending digital tokens, including transaction fees or gas fees. Model owners gain by acquiring tokens, but face costs from training models with private data or adapting existing models, in addition to gas fees.

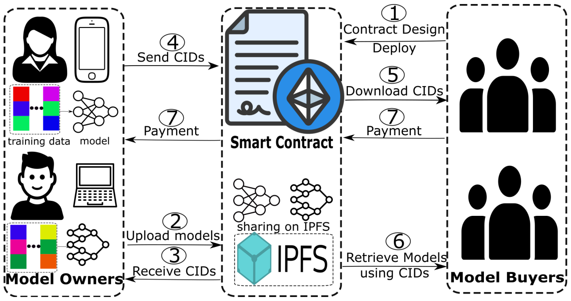
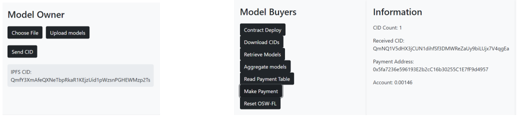


Figure 1: The System Overview of OFL-W3.

```
pragma solidity ^0.8.7;
contract CidStorage {
  uint256 public cidCount;
  ...
  function uploadCid(string memory cid) public {
    cids[cidCount] = cid;
    cidCount++;
    emit CidUploaded(cid);}
  ...
  function getCid(uint256 index) public view returns
 (string memory) {
    require(index < cidCount, "Invalid CID index");
    return cids[index];
  }
  ...
}

```

Figure 2: The partial example solidity codes of smart contract.



IPFS CID:
QmfY3XmAfeQXNeTbpRkaR1KEjzUid1pWzsnPGHEWMzp2Ts

CID Count: 1

Received CID:
QmNQ1V5dHX3jCUN1dihfSf3DMWReZaUy9biLUjx7V4qgEa

Payment Address:
0x5fa7236e596193E2b2cC16b30255C1E7fF9d4957

Account: 0.00146

(a) Model Owners **(b) Model Buyers**

Figure 3: Interfaces in OFL-W3.

3.2 Workflow

Step 1. Contract Design and Deploy. Model buyers design and deploy a smart contract tailored to a specific one-shot FL algorithm on the Sepolia ETH test network[1], specifying ML tasks, model structures, initial models, and necessary auxiliary information if the one-shot FL algorithm requires. They outline the payment in tokens and launch the contract on a commercial blockchain.

Step 2. Upload Models. Model owners find the smart contract using its address, agree to participate in the one-shot FL system, and prepare models according to the contract's specifications, including any necessary auxiliary information. They then upload these prepared models to the IPFS, with or without additional data.

[1] Note that the deployed contract can be directly transferred on the ETH mainnet since they use the same standard. However, due to the high price of ETH, we mainly show our system on SepoliaETH, one of ETH testnet. Now, 1 Sepolia ETH is around \$0.00006874 while 1 ETH is around \$3, 466.

Step 3. Receive CIDs. In the IPFS, a distributed file system, models are assigned 32-byte Content Identifiers (CIDs) through cryptographic hashes. This system ensures the unique accessibility and integrity of uploaded content, allowing for efficient model retrieval.
Step 4. Send CIDs. After receiving CIDs from IPFS, model owners submit these identifiers to the blockchain through the smart contract. This method conserves on-chain space, with each model occupying only 256 bits. As a comparison, at least Kb-level storage is needed if directly saving the model on the blockchain [1, 4], which proves to be impractical within the ETH network.
Step 5. Download CIDs. Model owners download the CIDs of all models shared via the smart contract, involving a process free of gas fees since it makes no data modification on the blockchain.
Step 6. Retrieve Models. After receiving the CIDs, the model owners can retrieve models with/without any auxiliary information. The retrieved models are used for the one-shot FL algorithm.
Step 7. Payment. The model buyers aggregate the retrieved model using its own one-shot FL algorithm, as denoted in the smart contract. The model buyers can adopt their own backend workstation/server to accelerate one-shot FL algorithm by using Flask to interact with the backend workstation. In this demonstration, we adopt PFNM [10] to aggregate the models. Then it assesses each participant's marginal contribution, like Leave-one-out (LOO), to pay the calculated tokens.

For the Dapp, the buyer's interface including Step 2 and Step 4 is illustrated in Fig. 3b, while the owner's interface including Step 1,2,5,6 and 7 is illustrated in Fig. 3a. The simplicity of the interface enables anyone with/without any knowledge of blockchain or Web 3.0 to use OFL-W3 by clicking buttons.

4 DEMONSTRATIONS AND EXPERIMENTS

In our demo, we simulate a scenario with ten model owners and a model buyer using a server with two NVIDIA RTX A5000 GPUs to run the PFNM one-shot FL algorithm, targeting to develop a high-quality model with a total cost of 0.01 ETH (approximately $34). The experiment utilizes the MNIST dataset and a neural network with three multi-layer perceptron layers (784, 100, 10). To mimic realistic non-IID data distributions, we use the data partitioning techniques in PFNM [10]. The local model training settings include a batch size of 64, a learning rate of 0.001, and 10 local epochs.

4.1 Model Performance

Figure 4 presents the quality of local models as evaluated by their test performance. This highlights the issue where, if a model owner is unable to effectively aggregate models from all participants, an individually trained model suffers from inadequate training data, leading to suboptimal performance. Conversely, the aggregated model demonstrates a test accuracy of 93.87%, surpassing the least effective single model by an impressive margin of 58.87%.

4.2 Transaction Costs

Our demonstrations outline critical interactions with the smart contract during Steps 1, 4, and 7, each incurring specific gas fees. Model owners deploy the smart contract by clicking the button, which triggers MetaMask to authorize the deployment. Submitting CIDs to the blockchain and transferring ETH to model owners are also facilitated by MetaMask, with model buyers covering the gas fees for these transactions.

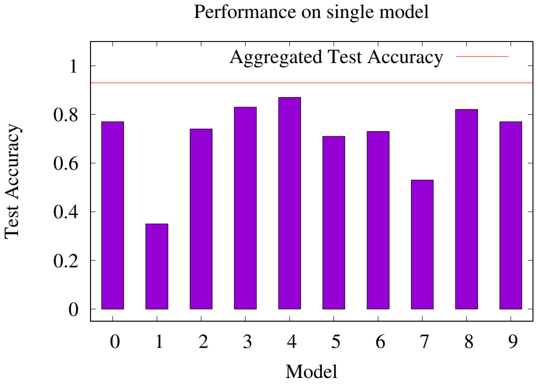
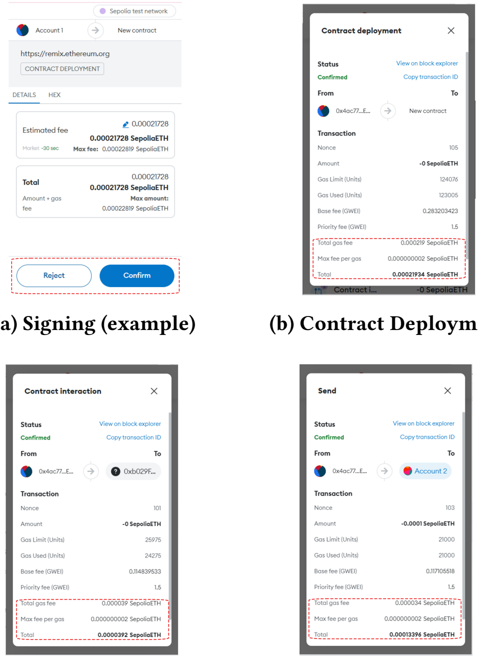


Figure 4: Single local model quality among 10 model owners.



Sepolia test network
Account 1 → New contract
https://remix.ethereum.org
CONTRACT DEPLOYMENT
DETAILS HEX
Estimated fee 0.00021728
0.00021728 SepoliaETH
Market -30 sec Max fee: 0.00022819 SepoliaETH
Total 0.00021728
0.00021728 SepoliaETH
Amount + gas fee Max amount: 0.00022819 SepoliaETH

(a) Signing (example)

Contract deployment
Status View on block explorer
Confirmed Copy transaction ID
From To
0x4ac77...E... → New contract
Transaction
Nonce 105
Amount -0 SepoliaETH
Gas Limit (Units) 124076
Gas Used (Units) 123005
Base fee (GWEI) 0.283203423
Priority fee (GWEI) 1.5
Total gas fee 0.000219 SepoliaETH
Max fee per gas 0.000000002 SepoliaETH
Total 0.00021934 SepoliaETH
Contract i... -0 SepoliaETH

(b) Contract Deployment

Contract interaction
Status View on block explorer
Confirmed Copy transaction ID
From To
0x4ac77...E... → 0xb029F...
Transaction
Nonce 101
Amount -0 SepoliaETH
Gas Limit (Units) 25975
Gas Used (Units) 24275
Base fee (GWEI) 0.114839533
Priority fee (GWEI) 1.5
Total gas fee 0.000039 SepoliaETH
Max fee per gas 0.000000002 SepoliaETH
Total 0.0000392 SepoliaETH

(c) Contract Interaction

Send
Status View on block explorer
Confirmed Copy transaction ID
From To
0x4ac77...E... → Account 2
Transaction
Nonce 103
Amount -0.0001 SepoliaETH
Gas Limit (Units) 21000
Gas Used (Units) 21000
Base fee (GWEI) 0.117105518
Priority fee (GWEI) 1.5
Total gas fee 0.000034 SepoliaETH
Max fee per gas 0.000000002 SepoliaETH
Total 0.00013396 SepoliaETH

(d) Payment

Figure 5: The transaction details shown on MetaMask.

Figure 5 shows the transaction process via MetaMask, where Figure 5a details the transaction confirmation phase. Figures 5b, 5c, and 5d illustrate the three different transaction types on the blockchain, each with varying total gas fees. From Figures 5b, 5c, and 5d, we can see deployment transactions carry the heaviest gas fees (e.g., 0.002 ETH) due to the need to write all functions on the blockchain. For our contract, gas fees for submitting 32-byte CIDs are similar to payment transactions as both involve writing to the blockchain. Downloading CIDs from the blockchain does not incur gas fees since they don't require data writing.

4.3 Payment

After retrieving the models, the model buyers aggregate them and then utilize incentive functions to compute payments. Figure 6

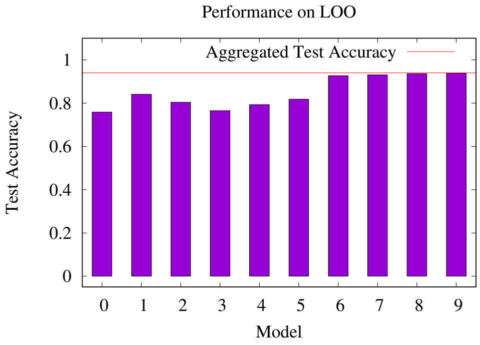


Figure 6: The test accuracy on Leave-one-out (LOO).

shows the test accuracy while any model is dropped. Thus, high test accuracy denoted on model *i* means less contribution for the model owner *i*. From the figure, we can see that model 7 is the most useless for the aggregated model.

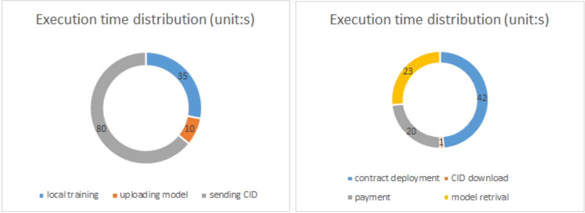
Table 1: Payment Table

Wallet Address [2]	Payment (ETH)
0xbC43368F3062Ba8605A17341d6054CFD649271dD	0.00162366
0x5fa7236e596193E2b2cC16b30255C1E7fF9d4957	0.00106922
0x5C892779A6DB3dA3716852Fa2e890B6A9626F159	0.00131720
0x7a305a674Fd11Ad96B56661A6CCe54266f7e2f56	0.00157930
0x0Ea87D03b7C394570000ed84777DeD7468A6Ad48	0.00139046
0xa3Df0eE2026f0448D309Cd8627a8b55Db20e814D	0.00122177
0x90341327A3B2Bbe2dDA305d6227d3e3ac6E363D0	0.00049194
0xED0F6C1A47F673A3D087016d48bc1FAf2b557d74	0.00046640
0xeB9865C6FAa7D146C8537005480BeC76d9AF1E03	0.00042876
0x981aDf746f0aF9717CF6f3f42Ad4Cef1b716cEe9	0.00041129

Table 1 shows the payment table computed from LOO payment function for 10 model owners. In detail, we allocate the payment based on each participant's contribution, as measured by LOO.

4.4 Overhead Measurement

We assess the computation and communication overheads of the entire process. Note that on the blockchain, 32-byte CIDs are transmitted, with the models in our experiments occupying 317Kb.



(a) Model Owners

(b) Model Buyers

Figure 7: Execution time distribution on owners and buyers.

The total time costs are evaluated from both the model owners' and buyers' perspectives. For model owners, it comprises local model training, model uploading, and sending CIDs to the contract. Model buyers' total time involves contract deployment, CID downloading, model retrieval, and payment processing, where payment calculation precedes the actual transaction.

Figure 7 presents the time distribution for both model owners and buyers within a unified campus area network, illustrating that the bulk of time consumption is attributed to blockchain interactions. While traditional FL systems may require at least 100 iterations, resulting in significant overhead, our findings endorse that one-shot FL is suitable for Web 3.0 applications.

5 CONCLUSION

In this paper, we introduce OFL-W3, a novel one-shot FL system tailored for Web 3.0 architecture, integrating blockchain technology with smart contracts for efficient transaction management and utilizing the IPFS for decentralized model sharing. Designed to bypass the limitation of the transaction speed and smart contract latency challenges prevalent in existing blockchain frameworks, OFL-W3 demonstrates a viable and innovative approach to implementing one-shot FL in the Web 3.0 context, offering unique insights and potential future directions. Our work not only showcases the practicality of FL applications in a new era of the internet but also sets the stage for further exploration and development within the AI + Web 3.0 domain, promising a transformative impact on both fields.

ACKNOWLEDGMENTS

This research / project is supported by the National Research Foundation, Singapore and Infocomm Media Development Authority under its Trust Tech Funding Initiative, and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-RP2020-018), and partially supported by the HighTech Support Program from STCSM (No.22511106200), Intel Corporation (UFunding 12679). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore, Infocomm Media Development Authority, STCSM and Intel Corporation.

REFERENCES

[1] Sana Awan, Fengjun Li, Bo Luo, and Mei Liu. 2019. Poster: A Reliable and Accountable Privacy-Preserving Federated Learning Framework Using the Blockchain. In *CCS*.

[2] Juan Benet. 2014. IPFS - Content Addressed, Versioned, P2P File System. *arXiv preprint arXiv:1407.3561*.

[3] David Carl and Christian Ewerhart. 2020. Ethereum Gas Price Statistics. *University of Zurich, Department of Economics, Working Paper No. 373*.

[4] Harsh Bimal Desai, Mustafa Safa Ozdayi, and Murat Kantarcioglu. 2021. Blockfla: Accountable Federated Learning via Hybrid Blockchain Architecture. In *CODASPY*.

[5] Yiqun Diao, Qinbin Li, and Bingsheng He. 2023. Towards Addressing Label Skews in One-Shot Federated Learning. In *ICLR*.

[6] Neel Guha, Ameet Talwalkar, and Virginia Smith. 2019. One-Shot Federated Learning. *arXiv preprint arXiv:1902.11175*.

[7] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *AISTATS*.

[8] Vaikkunth Mugunthan, Ravi Rahman, and Lalana Kagal. 2020. Blockflow: An Accountable and Privacy-Preserving Solution for Federated Learning. *arXiv preprint arXiv:2007.03856*.

[9] Gavin Wood and Gavin Parity. 2014. Ethereum: A Secure Decentralised Generalised Transaction Ledger. *Ethereum Project Yellow Paper*.

[10] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. 2019. Bayesian Nonparametric Federated Learning of Neural Networks. In *ICML*.

[2] Note that all wallet addresses are real and can be tracked on the Sepolia Etherscan. https://sepolia.etherscan.io/.