

# A SECURE FEDERATED LEARNING FRAMEWORK FOR 5G NETWORKS

Yi Liu, Jialiang Peng, Jiawen Kang, Abdullah M. Ilyasu, Dusit Niyato, and Ahmed A. Abd El-Latif

## ABSTRACT

Federated learning (FL) has recently been proposed as an emerging paradigm to build machine learning models using distributed training datasets that are locally stored and maintained on different devices in 5G networks while providing privacy preservation for participants. In FL, the central aggregator accumulates local updates uploaded by participants to update a global model. However, there are two critical security threats: poisoning and membership inference attacks. These attacks may be carried out by malicious or unreliable participants, resulting in the construction failure of global models or privacy leakage of FL models. Therefore, it is crucial for FL to develop security means of defense. In this article, we propose a blockchain-based secure FL framework to create smart contracts and prevent malicious or unreliable participants from being involved in FL. In doing so, the central aggregator recognizes malicious and unreliable participants by automatically executing smart contracts to defend against poisoning attacks. Further, we use local differential privacy techniques to prevent membership inference attacks. Numerical results suggest that the proposed framework can effectively deter poisoning and membership inference attacks, thereby improving the security of FL in 5G networks.

## INTRODUCTION

Distributed machine learning has spawned a lot of useful applications, such as on-device learning and edge computing [1]. However, due to communication delays and network bandwidth limitations in fourth generation (4G) networks, mobile users holding smart devices with limited computing power cannot fully participate in distributed machine learning tasks. Fortunately, in the fifth generation (5G) networks, bottlenecks attributed to communication latency and network bandwidth will be overcome [2]. Therefore, attention can be focused on addressing the performance, and efficiency issues of distributed machine learning can be addressed. Consequently, mobile devices will be able to participate in distributed machine learning. However, traditional distributed machine learning techniques require a certain amount of private data to be aggregated and analyzed at central servers during the model training phase [3]. Such a training process would

lead to potential privacy leakage for users in 5G networks [4, 5].

To address such privacy challenges, McMahan *et al.* [6] proposed a collaboratively distributed machine learning paradigm for mobile devices called federated learning (FL). FL involves the training machine learning models over devices<sup>1</sup> or siloed data centers while maintaining the training datasets locally without sharing raw training data [7]. The procedure of FL execution is divided into three phases: initialization, aggregation, and update phases. In the initialization phase, the central aggregator presents a pre-trained global model on a public dataset (e.g., MNIST,<sup>2</sup> CIFAR-10<sup>3</sup>) to each device. Following that, each device uses 5G networks to train and improve the current global model on the local dataset in each iteration.

In the aggregation phase, the central aggregator aggregates local model updates (i.e., gradient information) from the devices. In the update phase, the central aggregator aggregates all local model updates to generate a new global model for the next iteration. Both the devices and the central aggregator repeat the above process until the global model reaches a certain accuracy or optimal convergence. This paradigm significantly reduces the risks of privacy leakage by decoupling the model training from direct access to the raw training data [7].

Although FL brings new application scenarios (e.g., edge computing and on-device learning) to devices in 5G networks, it still suffers from two critical threats: poisoning and membership inference attacks [8]. First, if a poisoning attack occurs, an unreliable device submits an “error update” and the FL model would be “poisoned” [9]. A vicious attack will cause a failure in FL global model update. Second, in the case of the membership inference attack, since the intermediate gradients in training models always contain rich semantic information (e.g., model parameters), adversaries could exploit reverse engineering techniques to access some sensitive information (e.g., raw distribution of training data) [10]. Therefore, these vulnerabilities compel the need to develop a secure and privacy-preserving framework for FL, where the global model updates can be aggregated more securely.

In this study, we propose a secure FL framework that exploits the capabilities of smart contracts in

<sup>1</sup> We use the term “device” throughout this article to describe entities in 5G networks, including edge nodes, clients, smartphones, and so on.

<sup>2</sup> <http://yann.lecun.com/exdb/mnist/>

<sup>3</sup> <http://www.cs.toronto.edu/~kriz/cifar.html>

blockchain to defend against poisoning attacks and also introduce local differential privacy techniques to mitigate inference attacks. Specifically, smart contracts allow a task publisher to reward participants who can provide well-trained local models for a specific FL task [11]. The proposed framework establishes a marketplace to train FL models in an automated and anonymous manner for each participant (i.e., mobile device) through 5G networks. The local differential privacy technique provides robust defensive measures against membership inference attacks by adding well-designed noises to the model updates during the model training phase [10]. Based on the proposed framework, a central aggregator can create a decentralized and trustless marketplace to eliminate the impact of overreliance on participants in FL.

In 5G communication systems, task publishers can typically be deployed on edge nodes with sufficient storage and data processing capabilities, and mobile devices can easily access edge nodes to train local models. The proposed framework creates a marketplace where participants that are skilled in solving FL problems can profit directly from smart contracts [11]. The main contributions of this study are summarized as follows:

- We design a blockchain-based FL framework to achieve secure and reliable FL as well as all-round defense against poisoning attacks.
- We introduce the local differential privacy technology as a firewall against membership inference attacks on FL models.
- We create a marketplace where participants endowed in solving FL problems can benefit directly from their skills. This will provide a trustworthy platform to motivate participants to create better FL models.

## FEDERATED LEARNING AND ITS THREATS

### FEDERATED LEARNING AND ITS APPLICATION SCENARIOS IN 5G NETWORKS

Federated learning [6, 12] is a collaborative machine learning framework that requires no centralized training data. In FL, the local devices download the global model through 5G networks from the central aggregator, and then these devices train and improve the current global model by using their local raw data [6]. Typically, each device trains its local model using the Distributed Stochastic Gradient Descent (D-SGD) algorithm [13] and uploads the model updates (i.e., gradient information) to a central aggregator [7]. The aggregator updates a new global model by collecting all local updates and calculating the mean value of these local model updates. Federated learning aims to build machine learning models based on datasets stored locally in distributed devices without compromising privacy.

Some typical FL application scenarios in 5G networks are listed here:

- The Internet of Things (IoT): ByteLAKE and Lenovo present a technique called Federated Intelligence IoT<sup>4</sup> that not only enables IoT devices in 5G networks to learn from each other, but also makes it possible to leverage local machine learning models on IoT devices. In such scenarios, FL could be applied to build a personalized voice assistant without compromising personal privacy.

- The Internet of Vehicles (IoV): The Shenzhen Municipal Government in China cooperates with commercial organizations such as DiDi Chuxing and HelloBike companies to conduct real-time traffic flow forecasting [1] based on IoV in 5G networks. Meanwhile, General Data Protection Regulation (GDPR)<sup>5</sup> prohibits any organization from directly trading personal data due to concerns about privacy leakage. Therefore, FL can be applied to build a traffic flow prediction model that not only improves the prediction accuracy, but also protects personal privacy.

### THREATS AND COUNTERMEASURES

Although FL has many promising applications in scenarios related to 5G networks, two threats have been identified as obstacles to its full-scale deployment. Noting that it is assumed that the central aggregator is not compromised or malicious, we enunciate the two threats as follows:

- *Threat 1: Participants with malicious behaviors.* Participants may have intentional or unintentional malicious behaviors during the FL training phase [7]. Intentional malicious participants can submit wrong model updates resulting in failure of the update of the FL model. Unintentional malicious participants may upload model updates that may have negative effects on global model updates as they utilize low-quality training data. When the central aggregator gathers these local model updates to update the global model, it eventually leads to low accuracy or even a useless global model. All of these intentional or unintentional malicious behaviors may “poison” the FL model. In a word, the current FL models rely on a trust mechanism, which makes it vulnerable to poisoning attacks [14].
- *Threat 2: Disclosure of sensitive information.* In FL, each participant uploads the updated parameters of its own local training model to the central aggregator. However, existing studies show that adversaries can still initiate a membership inference attack to obtain sensitive information from the updated parameters associated with the local data. Specifically, adversaries can utilize the sensitive information disclosed by network snooping attacks in 5G networks to threaten the privacy of an FL model.

To address the poisoning attack, in [7], Kang *et al.* explored using blockchains to build a decentralized market using identity and reputation systems to deter poisoning attacks. However, such a method favors the gradient direction for updates uploaded from participants with high reputation value, which leads to the poor generalization ability of FL. Moreover, for local sensitive information leakage problems, existing studies show that the differential privacy technique is one of the best solutions [10]. Therefore, we introduce a local differential privacy technique to address the poisoning attack issues. Guided by the above considerations, we outline the main objectives in the proposed FL framework.

- *Establishing a fairness trading market for participants in a blockchain-based FL framework.* Based on Ethereum, the proposed framework can create a market for collaborative training

Federated learning is a collaborative machine learning framework that requires no centralized training data. In FL, the local devices download the global model through 5G networks from the central aggregator, and then these devices train and improve the current global model by using their local raw data.

<sup>4</sup> <https://becominghuman.ai/theres-a-better-way-of-doing-ai-in-the-iot-era-feabb-bc1b589>

<sup>5</sup> <https://eugdpr.org/>

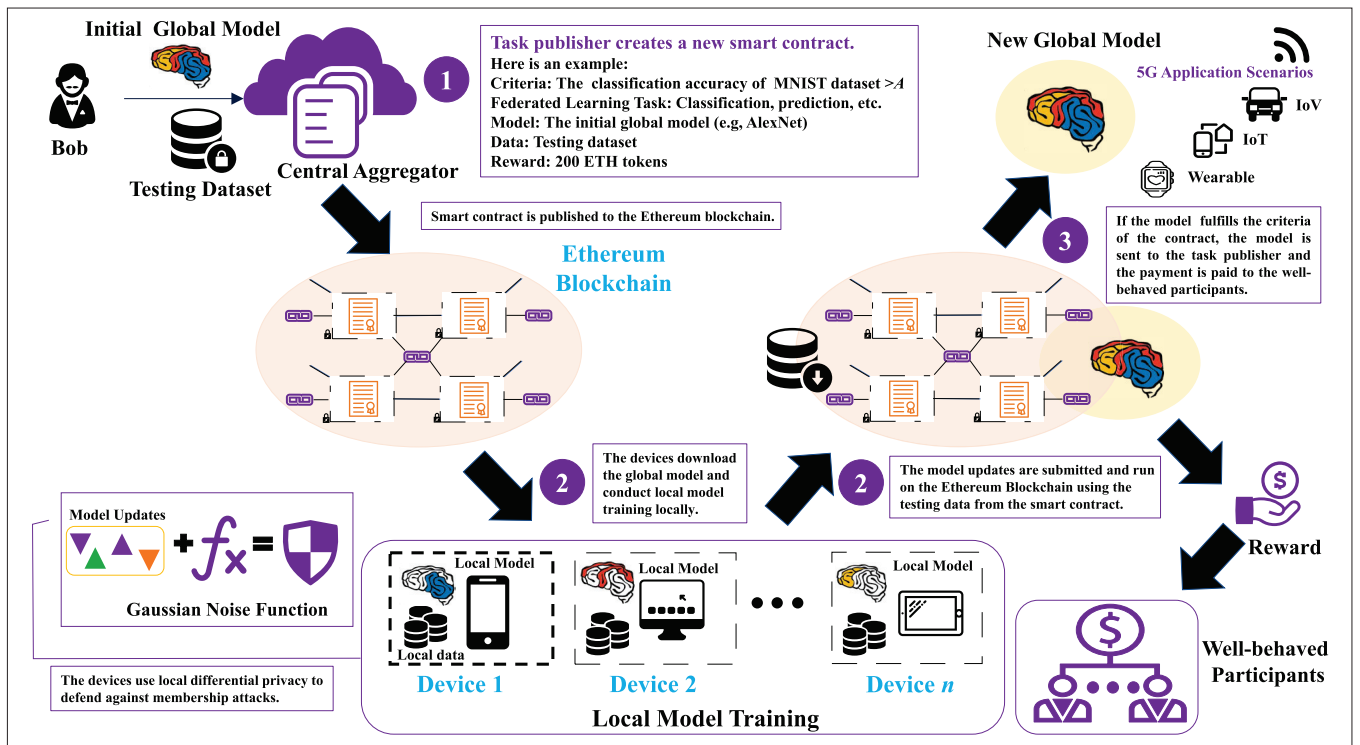


FIGURE 1. An overview of the proposed secure federated learning framework.

of FL models anonymously. Note that the proposed framework can also be applied to other blockchain platforms. In this marketplace, honest participants can make profits by submitting correct solutions to FL problems. To ensure the security of the framework, the central aggregator verifies the updates of local models based on smart contracts. The need for mutual trust between participants in this framework is eliminated because the protocol uses cryptographic authentication to secure all transactions.

- *Protecting sensitive information in FL.* In the proposed framework, participants add well-designed noises to their model updates by applying the local differential privacy technique. It utilizes some noise-adding mechanisms (e.g., Gaussian noise mechanism or Laplace noise mechanism) to secure model parameters. Therefore, even if adversaries gain access to the noise-adding gradient information, they cannot recover the original model parameters and local data.

## SECURE FEDERATED LEARNING FRAMEWORK

We first describe the related concepts and terms used in the proposed secure federated learning (SFL) framework:

- *Smart contracts:* Blockchain-based Ethereum implementation of a smart contract
- *Wallet address:* an escrow account of Ethereum that stores the rewards
- *Task publisher:* anyone who can publish FL tasks and interact with the central aggregator
- *Central aggregator:* an entity responsible for aggregating model updates uploaded by devices and interact with the task publishers
- *Device:* a physical entity in 5G networks involved in FL
- *Local model:* the model that a device trains with its local data

- *Global model:* an initial global model that is issued by the task publisher for the devices; updated by the central aggregators during model iterations
  - *Model update:* the gradient information generated by local model training
  - *Data points:* points made up of inputs and outputs
  - *Data group:* a matrix made up of data points
- Figure 1 further illustrates the pipeline of the proposed SFL framework. In this framework, we introduce the smart contract technique to deter the poisoning attacks (Fig. 2), and the local differential privacy technique is applied to curtail the inference attacks (Fig. 3).

- *Phrase 1, Initialization.* A task publisher named Bob creates a smart contract including a testing dataset, an initial global model, evaluation criteria, and a reward amount. The accuracy performance is used as the evaluation criterion to evaluate the training quality of the FL model. The reward is a monetary reward, such as ETH tokens. The above smart contract is published to the Ethereum blockchain. The central aggregator sends the initial global model to participants in Ethereum.
- *Phrase 2, Aggregation.* Each participant downloads the initial global model to train the local model using its local dataset. When a device succeeds in training a local model, it submits its local model updates to the Ethereum blockchain. All the Ethereum blockchain miners use the evaluation function in the smart contracts to evaluate all uploaded model updates and generate an average value of model quality for each uploaded model. We note that, similar to [6], a federated averaging algorithm (i.e., average aggregation) is used in the proposed framework. Participants use a differential privacy technique to add well-designed noise to the

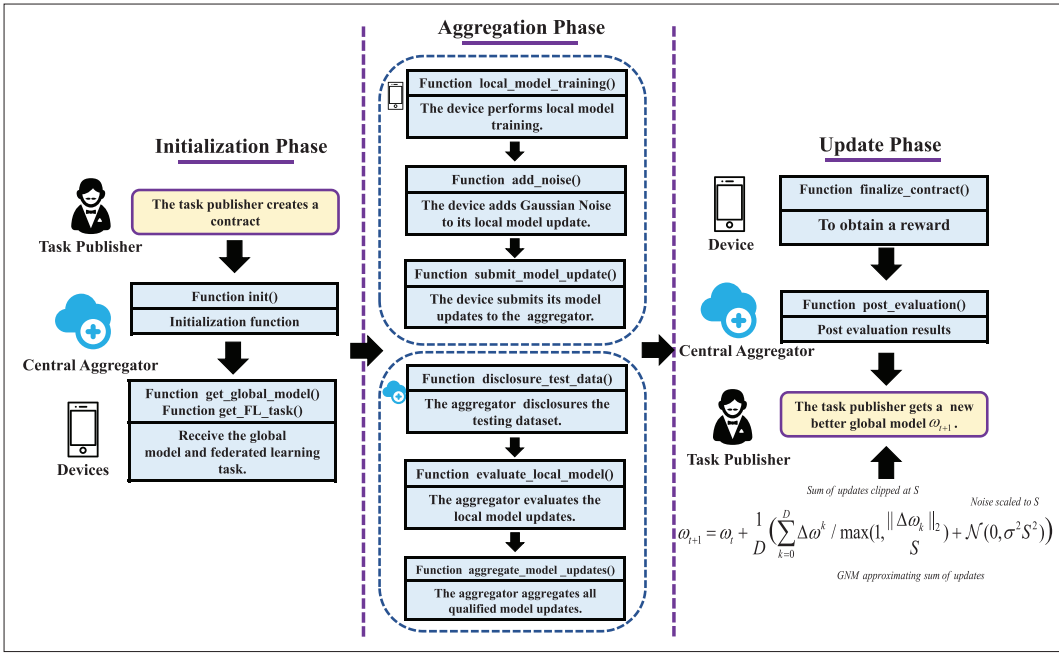


Figure 2. The workflow in the proposed secure federated learning framework.

uploaded model updates. The high-quality (i.e., high accuracy) model updates with larger values, and if they are larger than a given threshold determined by task publishers, they will be sent to the central aggregator.

- *Phrase 3, Update.* The task publisher gets the current global model from the central aggregator, and then prepares for the next round of iterative training.

The proposed SFL framework allows participants in the blockchain to implement local model training for rewards in a trustless manner. Moreover, the participants will be rewarded by the efforts of their high-accuracy model updates for FL tasks. As presented in Fig. 2, the detailed steps of the phases mentioned above are described as follows.

### INITIALIZATION PHASE

**Step 1:** The task publisher creates a smart contract with the following operations.

- The initialization function `init()` is called to initialize the SFL framework.
- The number of iterations is based on the size of the blocks.
- The task publisher hashes the testing dataset into the hashed data groups using a certain hash function.

**Step 2:** The task publisher hashes the indices of the data groups and sends this hashed result as the seed of random numbers to the central aggregator. The task publisher stores the reward in an encrypted address. Next, the central aggregator calls the `init()` function to initialize the SFL framework.

**Step 3:** The central aggregator verifies the testing data with the hash value previously provided for the data groups and writes the testing data groups and the random numbers into the smart contracts. The devices in 5G networks call the `get_global_model()` and `get_FL_task()` functions to receive the global model and FL task sent by the central aggregator, respectively.

At this point, the smart contracts initialization

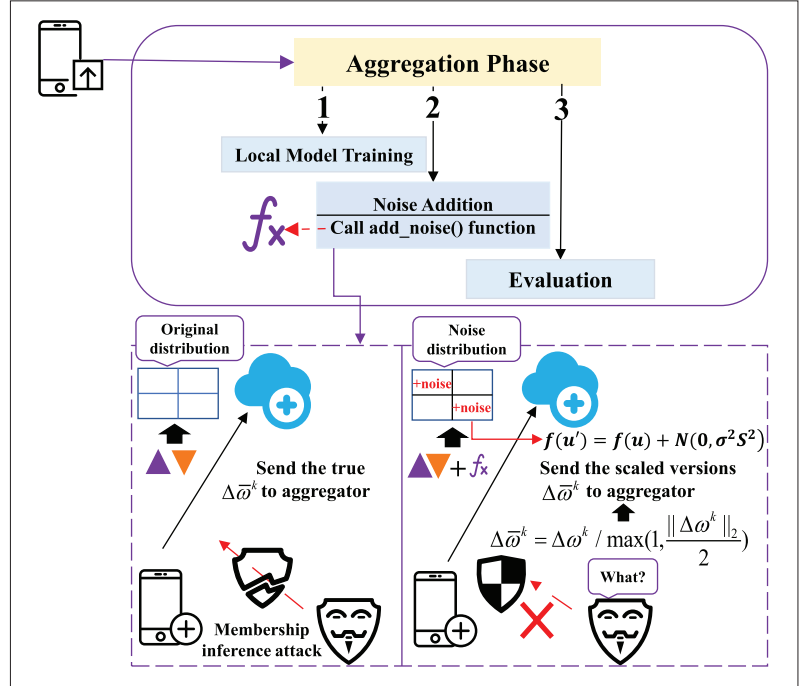


FIGURE 3. Illustration of three steps of the aggregation phase: (1) each device performs local model training and prepares to submit its local model update; (2) each device uses the differential privacy technique to add Gaussian noise on the local model update; (3) each device executes the `submit_model_update()` function to submit the model update.

in the SFL framework has been completed. Devices can begin to train their local models for a particular FL task.

### AGGREGATION PHASE

**Step 1: Local model training.** In this step, each device calls the `local_model_training()` function to use its local data to train the local model. Without loss of generality, it is assumed that the D-SGD algorithm [13] is used to update the local model updates.



Participants are compensated for the costs of executing transactions in the blockchain. These costs are called “gas” in the Ethereum blockchain. To avoid missing the mining block tasks, participants set the limit of gas themselves. Therefore, the upper limit of the gas in SFL framework will result in many devices missing the evaluation of the central aggregator.

**Step 2: Noise Addition.** As shown in Fig. 3, each device uses a Gaussian noise mechanism (GNM) to add noise on the local model updates prior to submission to the central aggregator. To defend against membership inference attacks, each device  $d_i$  first perturbs its model updates  $u_i$  using a randomized perturbation function  $f$  based on GNM. Next,  $d_i$  sends the perturbed model updates  $u_i'$  to the aggregator instead of its original model updates  $u_i$ . To achieve this goal, we require that the function  $f$  satisfies the requirement of  $(\epsilon, \delta)$ -local differential privacy  $(\epsilon, \delta)$ -LDP [10], where  $\epsilon > 0$  is a privacy parameter to make a trade-off between security and availability, and  $\delta$  is defined as the probability that plain  $\epsilon$ -LDP is broken.  $f(u')$  is denoted as a fixed-scale Gaussian noise added to  $f(u)$ , and the sensitivity  $S$  is defined as the maximum absolute distance between  $f(u)$  and  $f(u')$ . A device calls the `add_noise()` function, which combines the above  $(\epsilon, \delta)$ -LDP method and D-SGD algorithm to improve the privacy of FL. Meanwhile, the `add_noise()` function contains two main operations:

- Local Update. Let  $D$  be the total number of devices participate in FL. In each iteration, each device calls the `local_model_training()` function to compute its local model update. After execution of this function, a device  $k$  obtains its local models  $\{\omega^k\}_{k=0}^D$ .
- GNM Implementation. Each device can enforce a certain sensitivity by using scaled versions  $\Delta\omega^k$  instead of the true updates, where  $\forall k, \|\Delta\omega^k\|_2 < S$ . Dividing the GNM’s output by  $d$  yields an approximation to the real average of all device updates, while preventing the leakage of crucial information about an individual, as shown in Fig. 3.

Each device executes the `submit_model_update()` function to submit its model update with both the noise-adding gradient information and the payee address for the task publisher.

**Step 3: Evaluation.** The central aggregator can evaluate a single local model by the following operations. The central aggregator calls the `disclosure_test_data()` function that applies the testing dataset for local model evaluation. If the central aggregator fails to call the `disclosure_test_data()` function, the central aggregator uses its training dataset to evaluate the local model.

At this point, we note that once the evaluation begins to perform in the central aggregator, it no longer accepts model updates from devices. The central aggregator calls the `evaluate_local_model()` function to select the qualified devices whose model updates meet the evaluation criteria. It supports the central aggregator to observe the model aggregation results of each round before deciding whether to stop training, thereby reducing the consumption of computing resources and gas. Next, the central aggregator calls the `aggregate_model_update()` function to aggregate all local model parameters in the qualified devices to update the current global model. A new global model  $\omega_{t+1}$  is allocated by adding this approximation to the current global model  $\omega_t$  (Fig. 2). Finally, the central aggregator calls the `evaluate_model()` function to evaluate the updated global model and send the results to the task publisher to complete the current iteration.

## UPDATE PHASE

After the convergence of the global model update or once the number of iterations reaches the upper limit, the qualified device calls the `finalize_contract()` function to obtain a reward from the Ethereum blockchain. The central aggregator calls the `post_evaluation()` function to post the results of this FL task. Finally, the task publisher obtains a new, better global model.

## SOLUTIONS TO POTENTIAL ISSUES

Trust and fairness are critical to any marketplace, which necessitates the design of an FL framework where no user can cheat or gain an advantage over other users. Therefore, we further present the following solutions to potential issues related to the proposed framework.

### OVERFITTING PROBLEMS

Overfitting is one of the most common problems in machine learning. It reduces the generalization ability of machine learning models. Our proposed framework is not immune since overfitting problems could occur when:

1. The local training data contains additional noises.
2. The local training data is insufficient.
3. The local model is over-trained.
4. The task publisher releases the testing dataset in advance.

For problems 1, 2, and 3, the central aggregator can call the `evaluate_local_model()` function based on the testing dataset to solve. For problem 4, the task publisher and the central aggregator should keep the testing dataset unreleased during the model update phase.

### GAS COSTS PROBLEMS

Participants are compensated for the costs of executing transactions in the blockchain. These costs are called “gas” [11] in the Ethereum blockchain. To avoid missing the mining block tasks, participants set the limit of gas themselves. Therefore, the upper limit of the gas in the SFL framework will result in many devices missing the evaluation of the central aggregator. This means that the smart contracts would never be finalized [15].

To solve this problem, we perform the following two steps:

- We allow participants to call evaluation functions locally for evaluating their models. This approach reduces the computation required to evaluate the model, thus avoiding the problem of insufficient gas costs.
- The accuracy of the model needed for the device to receive rewards is announced in real time. Therefore, the device can decide whether to submit a model update based on its local model evaluation result.

## PERFORMANCE EVALUATION AND ANALYSIS

### EXPERIMENT SETTING

To evaluate the performance of the proposed SFL framework, we use Google Tensorflow for the simulation experiments on MNIST dataset and CIFAR-10 dataset, respectively. Without loss of generality, we assume that the task publisher releases the classification tasks on MNIST and

CIFAR-10, respectively. For the two tasks, the participants work together to train the FL model, AlexNet.<sup>6</sup>

The size of participant set  $P \in \{10, 100, 200, 300\}$ . It is assumed that  $P$  contains 20 percent malicious participants that launched a poisoning attack, 20 percent unreliable participants with low-quality data, and 60 percent well-behaved participants. The training set of well-behaved participants is randomly selected by a uniform distribution of 10 categories. It is assumed that malicious participants mislead the training model by modifying the labels of the training samples to carry out a poisoning attack. We use  $\lambda$  as the percentage of the modified labels that indicate the strength of poisoning attack, where  $\lambda$  in the experiments is set as 5, 10, 15, and 20 percent, respectively. The Earth Mover's Distance (EMD) represents the probability distance between a participant's data distribution and the actual distribution for the whole population, which is used as a metric to measure training data quality of the unreliable participants [7].

We deploy the proposed SFL framework on the Ethereum blockchain. Further, each participant applies the D-SGD algorithm to update model parameters iteratively. In the experiments, mini-batch size  $m = 128$ , learning rate  $\alpha = 0.0001$ , and dropout rate = 0.5. As in [10], we adopt a privacy budget of  $\epsilon = 8$ ,  $\delta = e^{-1}, e^{-3}, e^{-5}, e^{-6}$  to apply the  $(\epsilon, \delta)$ -LDP method, where  $\delta$  is the acceptable probability when the differential privacy is unbroken [13].

### PERFORMANCE ANALYSIS

Figure 4 illustrates the accuracy performance of the AlexNet model with different  $\lambda$  and EMDs on the MNIST dataset and CIFAR-10 dataset, respectively, when  $P = 100$ . It means that the three factors  $\lambda$ , EMD, and number of attackers influence the accuracy performance. The increase in the accuracy of the model is accompanied by a decrease in any one of the above factors. The results show that the proposed framework can deter poisoning attacks. Specifically, as the attack intensity increases, the accuracy of the proposed framework remains unchanged. As shown in Fig. 4b, for example, the learning accuracy is 65.2 percent for the CIFAR-10 classification task with 20 percent malicious participants exhibiting the strongest poisoning attack on the AlexNet model (i.e.,  $\lambda = 20$  percent,  $\text{EMD} = 1.5$ ). The central aggregator without defensive measures aggregates a large number of low-quality model updates as the attack intensity increases, thereby reducing the accuracy of the model. It implies failure of the AlexNet model in FL to perform image classification tasks under the poisoning attacks.

Meanwhile, the well-behaved participants call the `add_noise()` function to add noise to the gradient information, resulting in loss of model accuracy. Thus, these well-behaved participants may be misidentified as malicious participants because they degrade the accuracy performance of the model. To avoid the confusion of these well-behaved participants, there is a need to distinguish them from malicious participants in the proposed SFL framework. Figure 5 presents the highest accuracy (HA), minimum accuracy (MA), and average accuracy (AA) by 10 simulations on

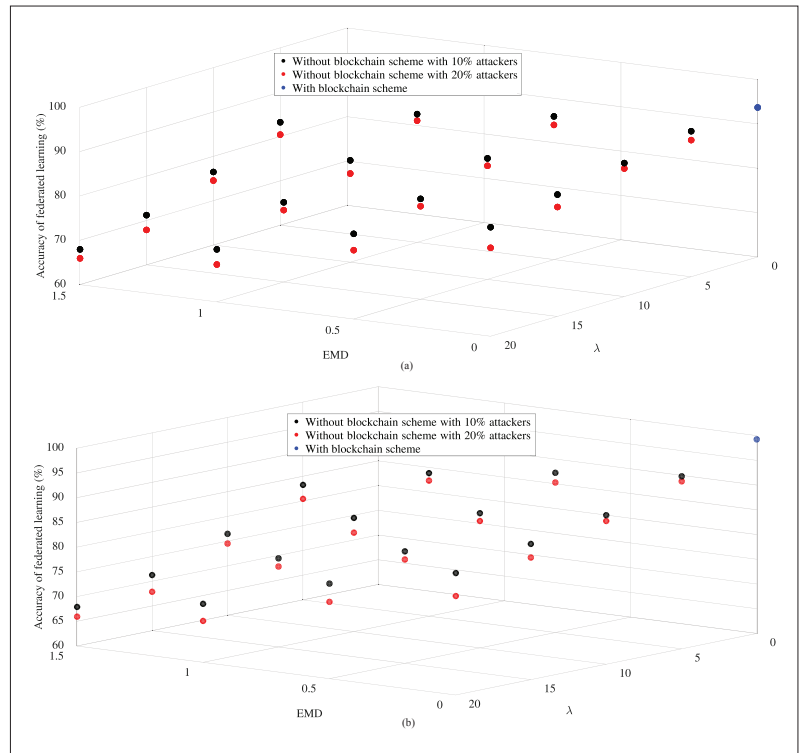


FIGURE 4. Accuracy performance with respect to  $\lambda$  and EMD on the datasets: a) MNIST; b) CIFAR-10.

the MNIST dataset and CIFAR-10 dataset, respectively. We adopt AA as the evaluation criteria to determine malicious participants in an FL task. For example, as presented in Fig. 5a, when  $P = 100$ , the lowest AA is obtained as 90.32 percent, which is the same as the accuracy of the model in Fig. 5a with  $\lambda = 5$  and  $\text{EMD} = 1.5$ . Here, AA (i.e.,  $\lambda = 5$ ,  $\text{EMD} = 1.5$ ) can be used as the threshold to determine malicious participants. The central aggregator calls the `evaluate_local_model()` function on the testing dataset to evaluate the local model uploaded by the participants. If the result of the above function is lower than AA, a reward will not be paid to the participant, who may be malicious. If not, the participant calls the `finalize_contract()` function to get a reward.

For Fig. 5, we compare the accuracy of the proposed framework with that of a typical FL framework (i.e., non-differentially private, non-DP, model) at different defense strengths of inference attacks. Numerical results show a trade-off between the protection from inference attacks and the accuracy of the model. Specifically, the accuracies of the non-DP model are 99.21 and 93.72 percent on MNIST and CIFAR-10 datasets, respectively. When  $\delta = e^{-6}$ ,  $P = 300$ , Fig. 5 illustrates that the accuracy of the proposed framework is close to that of the scheme without defensive measures, which indicates that the local differential privacy mechanism prevents inference attacks while sacrificing some accuracy. Besides, the central aggregator uses a batch average aggregation mechanism (Fig. 2) to mitigate the influence of noises on the accuracy of the model. Therefore, within a threshold of a certain number of participants (i.e., in this study, the threshold is set as 300), the more participants involved in the FL, the better the performance achieved.

<sup>6</sup> <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks>

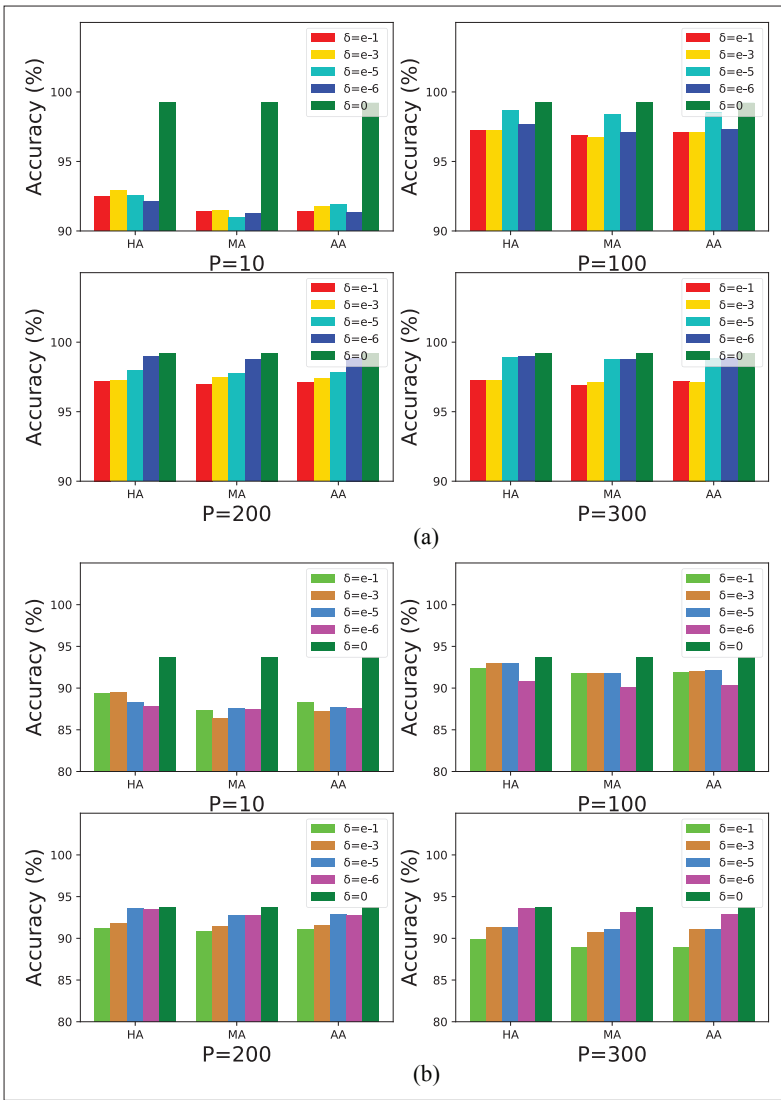


FIGURE 5. The accuracy performances with respect to  $P$ ,  $\epsilon$ , and  $\delta$  on the datasets: a) MNIST; b) CIFAR-10.

## CONCLUSIONS AND FUTURE DIRECTIONS

In this article, we address data privacy leakage issues related to ensuring secure FL in 5G networks. We propose a blockchain-based framework to defend against poisoning attacks. In the proposed framework, a market is created to trade model updates based on smart contracts in blockchain to validate the model updates against poisoning attacks automatically. Additionally, we introduce the local differential privacy technique in smart contracts to prevent membership inference attacks. Specifically, we add well-designed Gaussian noises to the model updates uploaded by the participants to defend against membership inference attacks. We validate our proposed security framework on two datasets, which yielded secure FL to 5G networks. The study has several open questions that need to be further investigated. First, because the federated averaging algorithm is used to aggregate the model updates, we encounter unfair usage to certain devices. Second, efficiency issues arise since the use of deep learning algorithms increases the computational demands of the proposed system. Third, the more participants in the proposed framework, the

greater the communication overhead, which in turn reduces the accuracy of FL. These are among the many issues that need to be addressed on the road to fair and efficient blockchain-based FL frameworks of the future.

## ACKNOWLEDGMENTS

This work is supported in part by the Science and Technology Project of State Grid Corporation of China (Grant No. SGHL0000DKJS1900883), in part by AMI, which acknowledges by funding from the Prince Sattam Bin Abdulaziz University, Saudi Arabia via the Deanship for Scientific Research funding for the Advanced Computational Intelligence & Intelligent Systems Engineering (ACISE) Research Group Project Number 2019/01/9862, in part by the National Research Foundation (NRF), Singapore, the Singapore Energy Market Authority (EMA), Energy Resilience, NRF2017EWT-EP003-041, Singapore NRF2015-NRF-ISF001-2277, Singapore NRF National Satellite of Excellence, Design Science and Technology for Secure Critical Infrastructure NSoE DeST-SCI2019-0007, A\*STAR-NTU-SUTD Joint Research Grant on Artificial Intelligence for the Future of Manufacturing RGANS1906, WASP/NTU M4082187 (4080), Singapore MOE Tier 2 MOE2014-T2-2-015 ARC4/15, NTU-WeBank JRI (NWJ-2020-004), and MOE Tier 1 2017-T1-002-007 RG122/17.

## REFERENCES

- [1] T. Alexander et al., "Mobile Communications and Networks," *IEEE Commun. Mag.*, vol. 57, no. 4, Apr. 2019, pp. 94–94.
- [2] A. A. EL-Latif et al., "Efficient Quantum-Based Security Protocols for Information Sharing and Data Protection in 5G Networks," *Future Generation Computer Systems*.
- [3] J. Cosmas et al., "A 5G Radio-Light SDN Architecture for Wireless and Mobile Network Access in Buildings," *2018 IEEE 5G World Forum*, July 2018, pp. 135–40.
- [4] K. Fan et al., "Blockchain-Based Efficient Privacy Preserving and Data Sharing Scheme of Content-Centric Network in 5G," *IET Commun.*, vol. 12, no. 5, pp. 527–532, 2017.
- [5] K. Cabaj et al., "SDN-Based Mitigation of Scanning Attacks for the 5G Internet of Radio Light System," *Proc. 13th Int'l. Conf. Availability, Reliability and Security*, ser. ARES 2018, ACM, 2018, pp. 49:1–49:10; <http://doi.acm.org/10.1145/3230833.3233248>
- [6] H. B. McMahan et al., "Communication-Efficient Learning of Deep Networks from Decentralized Data," *arXiv preprint arXiv:1602.05629*, 2016.
- [7] J. Kang et al., "Reliable Federated Learning for Mobile Networks," *IEEE Wireless Commun.*, vol. 27, no. 2, Apr. 2020, pp. 72–80.
- [8] R. Shokri et al., "Membership Inference Attacks Against Machine Learning Models," *2017 IEEE Symp. Security and Privacy*, May 2017, pp. 3–18.
- [9] Y. Wu et al., "Dominant Data Set Selection Algorithms for Electricity Consumption Time-Series Data Analysis Based on Affine Transformation," *IEEE Internet of Things J.*, vol. 7, no. 5, 2020, pp. 4347–60.
- [10] C. Xu et al., "Ganobfuscator: Mitigating Information Leakage under gan via Differential Privacy," *IEEE Trans. Info. Forensics and Security*, vol. 14, no. 9, Sept. 2019, pp. 2358–71.
- [11] A. B. Kurtulmus and K. Daniel, "Trustless Machine Learning Contracts; Evaluating and Exchanging Machine Learning Models on the Ethereum Blockchain," *arXiv preprint arXiv:1802.10185*, 2018.
- [12] T. Li et al., "Federated learning: Challenges, Methods, and Future Directions," *IEEE Signal Processing Mag.*, vol. 37, no. 3, 2020, pp. 50–60.
- [13] R. C. Geyer, T. Klein, and M. Nabi, "Differentially Private Federated Learning: A Client Level Perspective," *arXiv preprint arXiv:1712.07557*, 2017.
- [14] J.-S. Weng et al., "Deepchain: Auditable and Privacy-Preserving Deep Learning with Blockchain-Based Incentive," *IACR Cryptology ePrint Archive*, vol. 2018, 2018, pp. 679:1–679:16.

- [15] K. Cabaj *et al.*, “Network Threats Mitigation Using Software-Defined Networking for the 5G Internet of Radio Light System,” *Security and Commun. Networks*, vol. 2019, 2019.

## BIOGRAPHIES

YI LIU [S'19] (97liuyi@ieee.org) received his B.Eng. degree in network engineering from Heilongjiang University, Harbin, China, in 2019. He will pursue a Ph.D. degree at the Faculty of Information Technology, Monash University, Melbourne, Australia, in 2020. His research interests include security and privacy, federated learning, edge computing, and blockchain.

JIALIANG PENG (jialiangpeng@hlju.edu.cn) received his B.S. and M.S. degrees in computer science from Heilongjiang University and his Ph.D. degree in computer science at Harbin Institute of Technology, China. He is an associate professor in the School of Data Science and Technology, Heilongjiang University. He worked as a postdoctoral scholar at the Norwegian University of Science and Technology, Gjøvik, Norway. His research interests include biometric recognition, data management, and cyberspace security.

JIAWEN KANG (kavinkang@ntu.edu.sg) received his M.S. degree from Guangdong University of Technology, China, in 2015, and his Ph.D. degree from the same school in 2018. He is currently a postdoctoral researcher at Nanyang Technological University, Singapore. His research interests mainly focus on blockchain, and security and privacy protection in wireless communications and networking.

ABDULLAH M. ILIYASU (AKA ABDUL M. ELIAS) (abdul-m-elias@ieee.org) received his M.E., Ph.D., and Dr.Eng. degrees in computational intelligence and intelligent systems engineering from Tokyo Institute of Technology (Tokyo Tech.), Japan. Concur-

rently, he holds full-time and visiting faculty positions with the College of Engineering, Prince Sattam El Abdulaziz University, Saudi Arabia, the School of Computing, Tokyo Tech, and the School of Computer Science and Technology, Changchun University of Science and Technology, China. His research interests mainly focus on computational intelligence, quantum cybernetics, quantum image processing, quantum machine learning, cyber and information security, hybrid intelligent systems, the Internet of Things, 4IR, health informatics, and electronics systems reliability.

DUSIT NIYATO [M'09, SM'15, F'17] (dniyato@ntu.edu.sg) is currently a professor in the School of Computer Science and Engineering, Nanyang Technological University. He received his B.Eng. from King Mongkut's Institute of Technology Ladkrabang, Thailand, in 1999 and his Ph.D. in electrical and computer engineering from the University of Manitoba, Canada, in 2008. His research interests are in the area of energy harvesting for wireless communication, the Internet of Things, and sensor networks.

AHMED A. ABD EL-LATIF (aabdellatif@nu.edu.eg) received his B.Sc. degree with honors in mathematics and computer science and M.Sc. degree in computer science from Menoufia University, Egypt. He received his Ph.D. degree in computer science from Harbin Institute of Technology. He is an associate professor of computer science at Menoufia University and the School of Information Technology and Computer Science, Nile University, Egypt. He is a Fellow of the Academy of Scientific Research and Technology, Egypt. His areas of interests are multimedia content encryption, secure wireless communication, IoT, applied cryptanalysis, perceptual cryptography, secret media sharing, information hiding, biometrics, forensic analysis in digital images, and quantum information processing.