# SecCL: Securing Collaborative Learning Systems via Trusted Bulletin Boards

Zhichao Zhang, Ke Xu, Qi Li, Xin Liu, Lin Li, Bo Wu, and Yunzhe Guo


## Abstract

Massive and diverse data is crucial to train a general deep learning model, while the data collection for model training is difficult, especially training on sensitive data (e.g., medical data and face imaging). The emerging collaborative learning addresses this issue well by allowing participants to train a global model by uploading a subset of parameter changes, instead of the entire training data, to a centralized server. However, this privacy-preserving method can effectively enable privacy protection only when the involving entities are trusted (i.e., they honestly follow the protocol). Otherwise, the method may still leak private data. In this article, we propose a secure collaborative learning system named SecCL, which leverages a trusted bulletin board built on blockchain to enable strong privacy protection in collaborative learning by ensuring authentic and correct message interaction during the training process. Also, we develop a novel smart contract for SecCL so that participants can achieve consensus to restrain malicious behaviors. Therefore, SecCL ensures that the server cannot deceive participants and that participants behave well during the training process. We implement a prototype to evaluate its performance, and the promising experimental results demonstrate that SecCL can throttle malicious behaviors of participants and parameter servers while ensuring the accuracy of the global model.


## Introduction

Deep learning has been widely applied in various artificial intelligence fields (e.g., computer vision and natural language processing). In order to achieve higher learning accuracy, a large amount of diverse training data is vital to train a learning model. Unfortunately, many privacy-sensitive scenarios (e.g., medical and financial data learning) do not allow direct collection of such data due to the privacy issue. Therefore, privacy-preserving learning methods are required to realize secure learning on private data. To solve the issue above, Shokri and Shmatikov [1] proposed a collaborative learning system called PPDL, where participants collaborate with each other to train a global model through uploading a subset of parameter changes to a server instead of centralized data collection. Recently, Google developed an instance of collaborative learning, that is, a federated learning framework [2], which enables millions of smart devices to train a joint prediction model while keeping datasets private. Moreover, homomorphic encryption helps to prevent a parameter server's malicious behavior [3]. However, these existing systems cannot effectively preserve data privacy when the server or participants manipulate the learning process. For example, the current collaborative learning systems are still vulnerable to data leakage, and malicious participants can still steal the private training data by training a generative adversarial network (GAN) [4]. Colluders can poison the global model by training their delicately designed backdoor data [5], which can significantly decrease the accuracy of global models. In practice, it is difficult to ensure that both servers and participants are trusted, that is, they correctly follow learning protocols. Thus, it is necessary to develop a secure collaborative learning system.

It is challenging to develop such a system in a completely untrusted environment. To the best of our knowledge, there is no effective countermeasure to throttle the attacks above. In this article, we aim to answer the question: *is it possible to develop a secure collaborative learning system even if the server and participants are untrusted?* The answer is positive. Collaborative learning [1] (including federated learning [2]) is a typical distributed system, that is, multiple participants work together training a global model without exposing sensitive data. We can utilize the blockchain as a trust infrastructure to secure collaborative learning.

In this article, we propose a blockchain-based secure collaborative learning system named SecCL, which can restrict the malicious behaviors of participants and parameter servers while ensuring the accuracy of the global model. First, we design a mechanism for participants to verify the behaviors of parameter servers. By leveraging the information recorded on a trusted bulletin board (TBB) built on blockchain, any malicious behaviors of the server against the uploaded parameters will be traced by participants (e.g., tampering with the uploaded parameters). Thus, SecCL ensures that parameter servers must correctly perform the message interaction protocol. Second, we develop a novel consensus smart contract (CSC) for SecCL to restrict behaviors of malicious participants. CSC is built on multi-winner election rules and enables consensus among participants

The authors propose a secure collaborative learning system named SecCL, which leverages a trusted bulletin board built on blockchain to enable strong privacy protection in collaborative learning by ensuring authentic and correct message interaction during the training process.

*Zhichao Zhang, Ke Xu, Qi Li, and Yunzhe Guo are with Tsinghua University and the Beijing National Research Center for Information Science and Technology; Xin Liu and Lin Li are with Migu Culture Technology Co., Ltd; Bo Wu is with the 2012 Labs of Huawei Technologies.*

Digital Object Identifier: 10.1109/MCOM.001.1900130

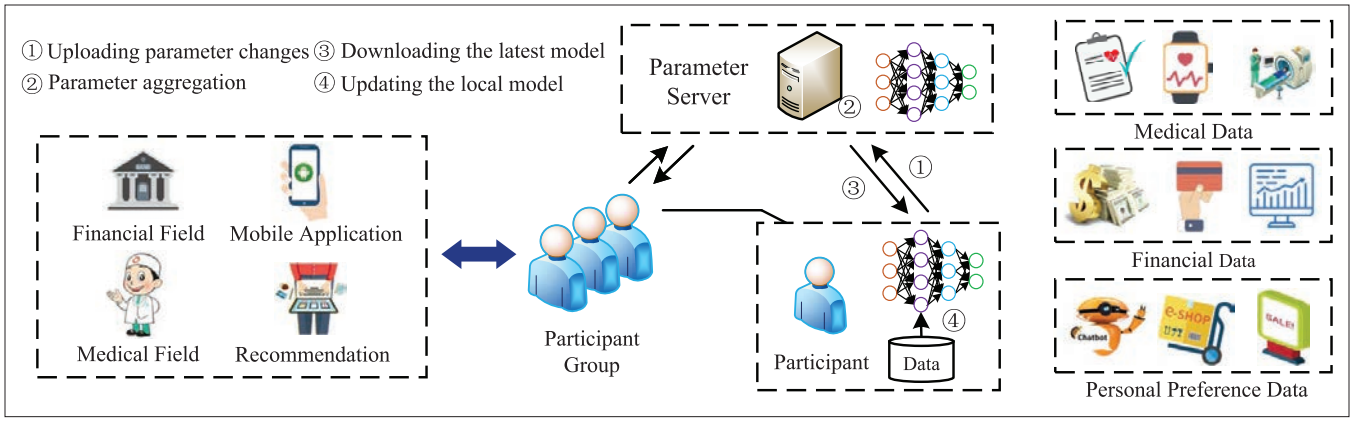


**Figure 1.** The workflow of the collaborative learning system.

to select the optimal parameter changes during each round of training. During the learning process, each participant evaluates the latest uploaded parameter changes with F-score and uploads the evaluation results to CSC. At the end of each round, CSC selects a new list of candidates based on the evaluations submitted by all participants (i.e., a subset of participants) to update the global model. And the parameters uploaded by candidates are optimal, which really contributes to the model training.

In what follows, we first review the development of collaborative learning and present the advantages of using blockchain technology. Then we present the system design of SecCL and its advantages. We implement a prototype using a typical model convolutional neural network (CNN) to verify the feasibility of SecCL, and the promising results show that SecCL ensures the security of the learning process while the model achieves high accuracy. Finally, we discuss the challenges that SecCL faces and conclude with future directions.

## Background

### Collaborative Learning

To improve efficiency when training large deep models with massive data, distributed deep learning methods [6] aim to find a global solution and assign training tasks to various central/graphics processing unit (CPU/GPU) clusters. Each cluster is called a worker, which trains the model with partial data. A parameter server is used to aggregate parameters from each worker and finally obtains a highly accurate global model. Inspired by distributed learning technology, Shokri and Shmatikov devised a system named PPDL [1], a privacy-friendly collaborative learning framework. PPDL treats each participant as a worker, in which the server collects each participant's parameters (i.e., gradients) instead of private data. Meanwhile, PPDL uses differential privacy to prevent parameter sharing from leaking sensitive information. Moreover, Google proposed federated learning [2] and secure aggregation protocol [7], which focus on training joint deep models on smartphones without collecting sensitive data. Each smartphone launches the training with its private local data (e.g., typing history), and a server collects the parameters submitted by each device to improve the global model. Yang *et al.* [8] combine federated learning with transfer learning, covering financial and medical scenarios among different enterprises. Furthermore, to enhance privacy, homomorphic encryption helps to protect messages interaction during the training process [3]. In a nutshell, Google implemented an instance of the collaborative learning system at scale (i.e., federated learning), based on which millions of smartphones train a global next word prediction model without collecting user data. The typical workflow of collaborative learning is shown in Fig. 1.

**Advantages of Collaborative Learning:** First, in privacy-sensitive scenarios, different organizations can collaborate to train a model without exposing private data, such as medical data, online personal preference, and chat history. For example, Yang *et al.* [8] envisage the framework deployed in financial fields (e.g., multiparty borrowing detection). Second, enterprises can form an alliance and amplify the value of isolated data that exists in different organizations. Enterprises within the alliance can unify data standards and train a joint model, which avoids disappointing model performance due to insufficient data.

**Weaknesses of Current Solutions:** The desired goal of privacy preserving can be sabotaged when malicious participants do not follow the learning protocol. For example, Hitaj *et al.* [4] point out that adversaries feed fake labels to the model and train a generative adversarial network (GAN), which can lead to the exposure of a private training sample of the victim. Besides, it is impossible to know who uploaded malicious parameters when using secure aggregation [7]. Thus, collusive adversaries can hide behind benign participants and stealthily launch their attacks to poison the global model by training their delicately designed backdoor data [5]. Meanwhile, a curious server may maliciously modify the parameters from a benign participant and launch similar attacks. Current solutions lack auditing assessment of participants, which limits the practical deployment of collaborative learning.

### Blockchain and Smart Contracts

The blockchain is essentially a reliable distributed database for recording transactions. Miners generate blocks and link them one by one to construct an immutable ledger. Each block includes multiple transactions, which can record not only the money transferred from payer to payee but also

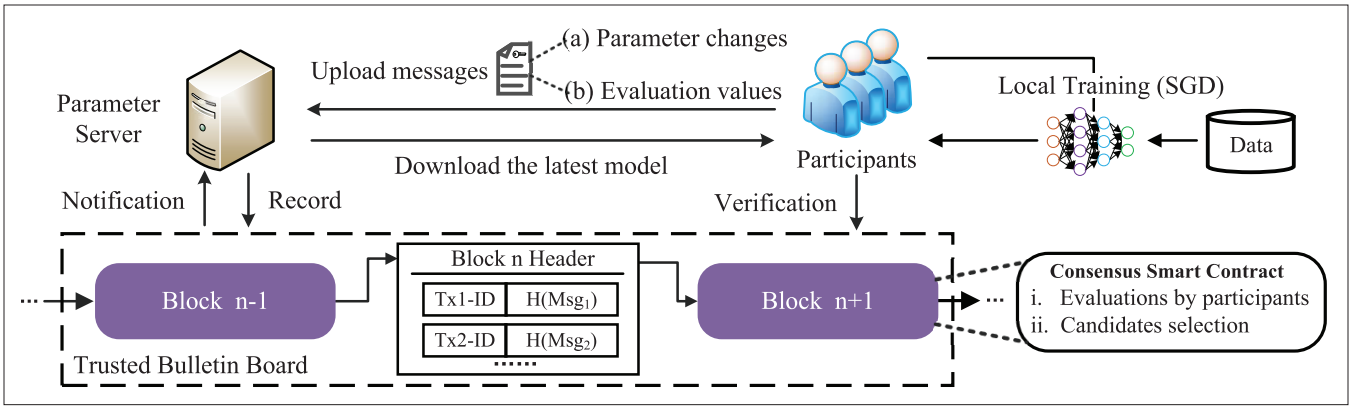


**Figure 2.** High-level overview of SecCL.

data for a specific application. For example, Bitcoin allows embedding at most 80 B statements to a transaction. In a public (or permissionless) blockchain such as Bitcoin, anyone can participate in it without a specific identity. Thus, sophisticated consensus mechanisms such as proof of work (PoW) are required to ensure the consistency of transactions in a fresh block. Different from the above, the permissioned blockchain can secure the interactions among a group of identified entities. These entities have a common goal (e.g., training a joint model) but may not fully trust each other. By relying on the identity of each entity, the blockchain nodes do not require costly mining (e.g., PoW) to reach consensus, which makes SecCL ready for deployment. However, both blockchains ensure that data cannot be tampered with by a few untrusted entities once the transaction is included in a block and confirmed by miners. We can utilize this feature of blockchain to track and verify the server's behaviors [9].

Smart contracts are executable programs deployed on the blockchain and can get credible results by leveraging blockchain's security guarantee. Currently, many blockchain platforms support smart contracts (e.g., Ethereum, Hyperledger Fabric). Smart contracts will be executed automatically once invoked by users. Thus, participants can reach a consensus (e.g., selection of optimal parameters, incentives in crowdsourcing [10]) without depending on any centralized servers, which may be manipulated by powerful adversaries. In a nutshell, blockchain can offer the following properties to SecCL.

**Trusted Records:** The TBB ensures that participants cannot deny the parameters uploaded by themselves, and servers cannot manipulate the data recorded on the TBB.

**Reliable Contribution Assessment:** The CSC built on multi-winner election rules enables consensus among participants to select the optimal parameter changes, which does not rely on any centralized authorities while ensuring that no party can tamper with the rules of the contribution assessment.

## SecCL Design

### Basic Idea

In SecCL, participants leverage TBB to verify parameters from the parameter server, which maintains the global model and receives parameters to improve the learning performance. Meanwhile, by leveraging multi-winner election rules in the CSC, participants can reach consensus to select the best parameters. Figure 2 shows three entities in SecCL.

*Participants* ($P$) have the same training target and collaborate with each other by sharing a subset of parameter changes. There are two types of interaction messages used by $P$ to communicate with the server, and one *round* of training means that all participants have successfully uploaded the type (a) messages containing their selected parameter changes. Moreover, the participant who uploads a type (a) message will verify the hash recorded on TBB to ensure that the server cannot send fake parameters to other participants.

A *parameter server* interacts with $P$ and TBB. The server maintains parameters of the global model, and receives messages from $P$. Once the server receives a message from $p_i$ (one participant in SecCL), it will record the hash of the message on TBB and return the corresponding transaction-ID (Tx-ID) to $p_i$ for verification. The Tx-ID is a hash string (e.g., SHA-256) that uniquely identifies a transaction. At the end of each round of training, the server receives a list of candidates selected by CSC and updates the global model.

The TBB is built on the blockchain. The hash of a message is embedded in a transaction and cannot be tampered with once the transaction is included in a block. Besides, the CSC determines whose parameters are optimal to update the global model based on the evaluation values from $P$. Here, participants, whose uploaded parameter changes are finally adopted by the global model, are called *candidates* ($C$).

In the following subsections, we elaborate on the interactions between the entities and how the consensus among participants is achieved to securely train learning models.

### Message Interaction Protocol

The message interaction protocol begins with the boot of the collaborative learning phase. As shown in Fig. 2, the message interaction protocol details are as follows.

**Participant-Server Interaction:** There are two types of messages in the protocol. Figure 3 shows the workflow of a participant. First, at the beginning of each round of training, $P$ download the latest global model from the server and start their local training via stochastic gradient descent

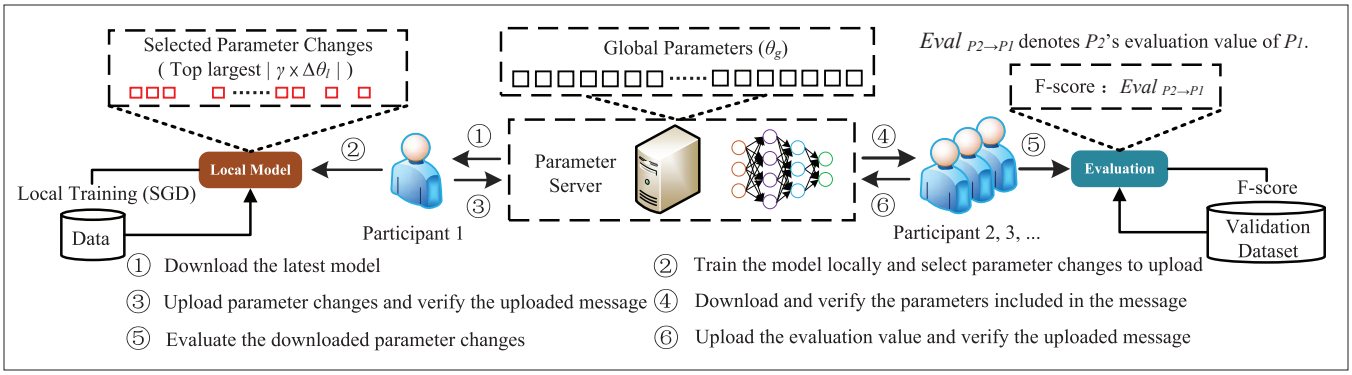


**Figure 3.** The steps taken by a participant to interact with the server.

(SGD). When the local training is completed, *P* asynchronously select partial parameters with the largest changes to upload (i.e., $\Delta\theta_l$ in Fig. 3) and send a type (a) message to the server. Second, when the server receives new parameters from $p_i$, all the other participants will be notified to download the message and the corresponding Tx-ID. Then they use the Tx-ID to verify the authenticity of the downloaded message and evaluate $p_i$'s uploaded parameter changes with their own private validation datasets. After that, they send a type (b) message containing the evaluation value (i.e., $Eval_{p2\rightarrow p1}$ in Fig.3) to the server.

**Server-TBB Interaction:** First, the server embeds the hash of both types of messages in a transaction and broadcasts it to the blockchain network, after which the corresponding Tx-ID will return to *P* for public verification. For example, *P* can use the Tx-ID to query the hash value to verify the authenticity of the message (e.g., H(Msg1) in Fig. 2). Note that due to the consensus for block generation, there is a delay in confirming the transaction. However, this delay normally takes several seconds, which is negligible compared to the training process. Second, the server transfers the evaluation values from *P* to CSC. At the end of each round, CSC selects new candidates based on evaluation values and then notifies the server of the list of *C* to update the global model. The invoking process of CSC is public on the TBB, so participants can trace the execution flow to ensure that the server correctly performs the proposed protocols and transfers the evaluation values to CSC.

### Consensus among Participants

The CSC is deployed on the TBB and implements the selection of *C*. The parameter changes from *C* should perform well on all participants' validation datasets. Thus, CSC stores evaluation values from *P* and converts the selection into a multi-winner problem, which has been extensively studied [11]. In SecCL, the multi-winner election considers the problem of selecting optimal parameters to update the global model. We use the *Broda* positional scoring function (PSF) as the basis for the selection. Considering that we choose *M* candidates, CSC first sorts all evaluation values from $p_i$ in descending order and selects a subset including *M* participants of top evaluation values. Then the scores of the other participants are calculated based on PSF and their positions in the subset selected by $p_i$. The total Broda score of a participant is the accumulative PSF score from all the other participants. Finally, *C* should be the subset of *P* with the greatest total *Broda* score. When CSC receives the evaluation values from all participants, it will select a new list of candidates based on the above and notify the server to update the global model. In a nutshell, the consensus among the participants ensures that only the optimal parameters can update the global model.

## Advantages of SecCL

### Obtaining Generalized Deep Models

Participants have the same training target in SecCL, for example, a specific task in the healthcare area. Therefore, the global model must perform well on all participants' validation datasets. Previous work lacks verification mechanisms, so the low-quality parameters cause the model to converge slowly or even fail to get highly accurate deep models. Moreover, participants may have non-IID data for local training. For example, each participant only has partial data when training a multi-class classifier. Thus, the parameter server cannot directly aggregate the parameters from all participants. Otherwise, the global model may deteriorate [2]. The consensus in SecCL ensures that the server only aggregates the optimal parameters, which facilitates the global model converging in a stable manner.

### Throttling Malicious Behaviors

**Security against Passive Adversaries:** "Passive" adversaries will loyally execute the message interaction protocol. However, some local training issues of each participant will delay the model convergence. For example, $p_i$ may have low-quality training data, and the obtained parameters do not contribute to improving accuracy. Based on the rules above, other participants will have a low evaluation value for $p_i$, and the CSC will not select $p_i$ as a candidate to update the global model, which effectively restricts the negative effects and ensures the accuracy of the global model.

**Security against Active Adversaries:** "Active" adversaries will actively attempt to threaten the data from other participants or sabotage the model. Active adversaries can be divided into two categories: malicious participants and curious servers.

First, previous works [4, 5] have shown that an active malicious participant can poison the global model or even steal a victim's local data through

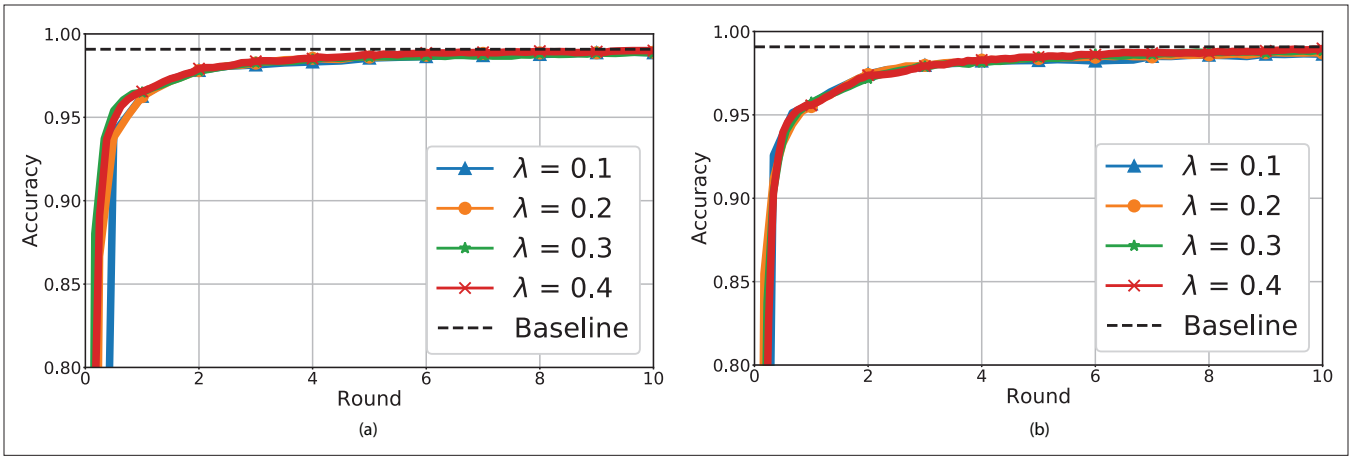


**Figure 4.** Feasibility verification of SecCL: a) 20 participants, CNN; b) 30 participants, CNN.

uploading parameter gradients derived from training their delicately designed data. These parameters lead to the model diverging from the latest state, and the accuracy will drop sharply, which can be detected by F-score. SecCL enables participants to observe and evaluate others' behaviors for the consensus that only the optimal gradients from benign participants can update the model. Thus, a single adversary will not succeed in sabotaging the model.

Second, the parameter server collects and distributes the parameter changes. It may modify or drop the parameters from a benign participant on purpose and launch similar attacks as in [4, 5]. By leveraging the records on the TBB, each participant is able to verify the authenticity of the parameters and check whether the parameters are modified by the server. Thus, any malicious behaviors of the server against the uploaded messages will be traced by all participants.

**Security against Collusions:** Note that the server is monitored by benign participants. Thus, we consider malicious participant collusion. Based on the algorithm of CSC, collusive adversaries may upload fake ranks and select one of them to affect the training process. However, the parameters uploaded by adversaries do not contribute to the model, that is, benign participants will have low evaluation values of them. Thus, SecCL can throttle adversaries' malicious behaviors as long as the number does not exceed a threshold ratio of all participants. Moreover, the aggregation method (e.g., parameter averaging) also eliminates occasional negative effects.

## Experimental Evaluation

In this section, we implement a prototype using Ethereum geth, based on which we built the experimental blockchain network to verify the feasibility. The prototype runs on Ubuntu 16.04 (Intel® Core™ i7-7700 CPU @ 3.60 GHz and 16 GB memory). We implement CSC using Solidity language for the selection of optimal parameters in each round. We use the JSON RPC application programming interface (API) and the Web3 library to implement the interaction with the blockchain. For the training process, we implement a server prototype with python to simulate interaction with the participants and the blockchain. Then participants use the *keras* toolkit to train the model and the *sklearn* toolkit to evaluate the parameters from participants, where the F-score is the average of each class to evaluate a multi-class classifier. The numbers of participants belong to {20, 30}. We use $\lambda$, $\varepsilon$ to represent the fraction of selected candidates and malicious adversaries, respectively. To unify experimental standards, each participant has the same local training strategy, that is, mini-batch size 32, learning rate 0.02, and the fraction of the sharing parameters 0.1.

### Feasibility Verification

In this experiment, we focus on verifying the feasibility of SecCL without considering the existence of adversaries. We randomly divide the MNIST dataset equally for each participant, that is, each participant may have different numbers of data for each class. Besides, we utilize the same settings (e.g., mini-batch size, learning rate) to train the model on the entire dataset and use the accuracy as the baseline (0.9908).

Figure 4 shows the accuracy of the global model. Training models can achieve similar accuracy to the baseline for different settings of participants and $\lambda$. As the number of participants increases, the convergence time will increase slightly, but it will not affect the final accuracy of the global model. Moreover, without the existence of adversaries, we found that the CSC selecting more candidates will accelerate the accuracy improvement. For example, when 30 participants collaborate for 10 rounds, $\lambda = 0.4$ enables the model to achieve a higher accuracy that is closer to the baseline than $\lambda = 0.1$.

### Effectiveness of Throttling Adversaries

In this experiment, we consider the case of 20 participants to verify the effectiveness of throttling malicious behaviors of adversaries. They collude to feed fake labels to the training model (similar to [4]) when *Round* = 5.

As shown in Fig. 5, the collaborative learning process is not affected when $\varepsilon < 0.25$. The accuracy of the global model sharply decreases when $\varepsilon \geq 0.25$, which can be detected by benign participants on their validation datasets. Thus, SecCL can restrict the negative effect of low-quality parameters as long as the number of adversaries does not exceed a threshold.

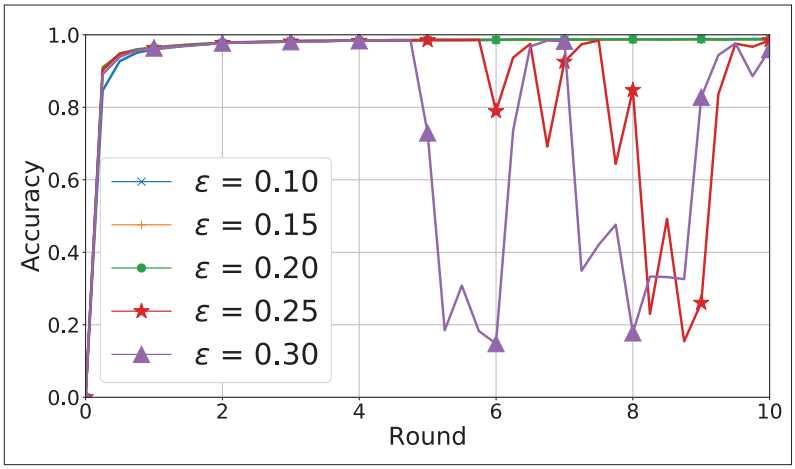


**Figure 5.** Throttling adversaries: 20 participants, λ = 0.2, CNN.

Moreover, we found that there is a trade-off between security and efficiency. As shown in the first experiment, without the existence of adversaries, selecting more candidates will improve the efficiency of training as all the parameters contribute to improving the global model. However, it also means that the system is more vulnerable to collusive adversaries, because the more candidates, the more easily collusive adversaries can sneak into the candidate group. Thus, it is essential to select a reasonable number of candidates when considering different factors (e.g., preferred security or efficiency).

## Challenges and Opportunities

SecCL is a secure collaborative learning system, in which entities utilize the designed protocol for behavioral auditing. However, it still has some open issues.

**Training Verification:** In SecCL, participants share parameters to train a global model. Although SecCL ensures that forged parameters cannot affect the global model, current frameworks lack a mechanism for verifying whether the parameters are derived from local training. For example, a malicious participant may not execute local training and steal others' contributions by uploading modified parameters. Thus, we need to design more sophisticated proof-of-training mechanisms to prevent such behaviors. Naively, we can use smart contracts to execute verification algorithms, which provide a more equitable assessment without any centralized authority.

**Data Alliance and Incentives:** In privacy-sensitive scenarios, enterprises can form a data alliance for collaborative learning. The first essential step to put SecCL into practical use is to unify data standards. For example, data for lung disease may vary from different hospitals. Yang *et al.* have raised this issue [8] and attempt to solve it by transfer learning. Then the data owners can obtain incentives based on the contribution assessment rules in the smart contract, which will attract enterprises to join the alliance (e.g., Smartretro [12]). However, related works indicate that bugs in smart contracts will affect the execution results [13, 14], and adversaries may manipulate the invocation of smart contracts [15]. We can leverage the existing mechanisms (e.g., cross-graph analysis [15]) to detect vulnerabilities before deploying the contracts to avoid economic losses. Based on the above, AI companies can pay for the data without violating laws such as GDPR.

**Deployments for More Scenarios:** SecCL should be agile and compatible with more application scenarios. For example, the required data is cross-industry in some scenarios. Take the recommendation system as a case. Vendors analyze personal preferences and want to recommend products that match users' purchasing power, but the information usually exists in banks, which cannot be collected to train a model. As mentioned above, we can leverage the permissioned blockchain to form a data alliance. Each enterprise maintains blockchain nodes to record information during the collaboration. Meanwhile, the permissioned chain offers the identity of each enterprise, based on which SecCL can utilize smart contracts to allocate the profit equitably during the training process.

**Performance Improvement:** First, we should optimize the training algorithm. For deep learning models, uploading parameters may cause more overhead than collecting user data. This issue has been proposed by Google [2] when they deploy the system to train a next-word-prediction model on millions of Android phones. As a countermeasure, we can improve the performance to assign tasks based on the communication capability of participants and encode the parameters to upload. Second, in the experiments, we implemented the prototype based on geth, which aims to verify the feasibility of SecCL. However, in practical deployment, we should build the infrastructure of SecCL on the permissioned blockchain such as Hyperledger Fabric, in which identified blockchain nodes use traditional consensus mechanisms without costly mining and provide sufficient performance for SecCL.

## Conclusion

In this article, we propose a blockchain-based secure collaborative learning system that can restrict the malicious behaviors of the parameter server and participants while still ensuring the accuracy of the training model. Without exposing data to train deep learning models, collaborative learning will create a new business model and increase the enthusiasm of people for participating in collaboration. Combined with blockchain technology, we can use smart contracts to establish equitable mechanisms to assess the contribution of participants, improving the model and providing incentives for them based on the assessment, which enables personal control of private training samples and amplifies the value of precious but unshareable data. Moreover, we summarize open issues in the current collaborative learning systems and point out potential solutions using blockchain.


### Acknowledgments

This work was in part supported by the National Key R&D Program of China with No. 2018YFB0803405, National Science Foundation for Distinguished Young Scholars of China (Grant No. 61825204), National Natural Science Foundation of China under Grant 61932016, 61572278, and U1736209, Beijing Outstanding Young Scientist Program with

No. BJJWZYJH01201910003011, and Beijing National Research Center for Information Science and Technology (BNRist) with No. BNR-2019RC01011.

## References

[1] R. Shokri and V. Shmatikov, "Privacy-Preserving Deep Learning," *Proc. 22nd ACM SIGSAC Conf. Computer and Commun. Security*, ser. CCS '15, 2015, pp. 1310–21.
[2] K. Bonawitz *et al.*, "Towards Federated Learning at Scale: System Design," arXiv preprint arXiv:1902.01046, 2019.
[3] M. Shen *et al.*, "Secure SVM Training over Vertically-Partitioned Datasets Using Consortium Blockchain for Vehicular Social Networks," *IEEE Trans. Vehic. Tech.*, 2019.
[4] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep Models under the GAN: Information Leakage from Collaborative Deep Learning," *Proc. 2017 ACM SIGSAC Conf. Computer and Commun. Security*, ser. CCS '17, 2017, pp. 603–18.
[5] E. Bagdasaryan *et al.*, "How to Backdoor Federated Learning," arXiv preprint arXiv:1807.00459, 2018.
[6] J. Dean *et al.*, "Large Scale Distributed Deep Networks," *Advances in Neural Information Processing Systems 25*, 2012, pp. 1223–31.
[7] K. Bonawitz *et al.*, "Practical Secure Aggregation for Privacy-Preserving Machine Learning," *Proc. 2017 ACM SIGSAC Conf. Computer and Commun. Security*, 2017, pp. 1175–91.
[8] Q. Yang *et al.*, "Federated Machine Learning: Concept and Applications," *ACM Trans. Intelligent Systems and Technology*, vol. 10, no. 2, 2019, p. 12.
[9] Y. Kucuk *et al.*, "Bigbing: Privacy-Preserving Cloud-Based Malware Classification Service," *Proc. 2018 IEEE Symp. Privacy-Aware Computing*, 2018, pp. 43–54.
[10] B. Wu *et al.*, "Smartcrowd: Decentralized and Automated Incentives for Distributed IoT System Detection," *Proc. IEEE 39th Int'l. Conf. Distributed Computing Systems*, 2019.
[11] Y. Wu *et al.*, "A Scalable Collusionresistant Multi-Winner Cognitive Spectrum Auction Game," *IEEE Trans. Commun.*, vol. 57, no. 12, 2009.
[12] B. Wu *et al.*, "Smartretro: Blockchainbased Incentives for Distributed IoT Retrospective Detection," *Proc. 2018 IEEE 15th Int'l. Conf. Mobile Ad Hoc and Sensor Systems*, Oct. 2018, pp. 308–16.
[13] T. Chen *et al.*, "An Adaptive Gas Cost Mechanism for Ethereum to Defend Against Underpriced DoS Attacks," *Info. Security Practice and Experience*, J. K. Liu and P. Samarati, Eds., Springer, 2017, pp. 3–24.
[14] L. Luu *et al.*, "Making Smart Contracts Smarter," *Proc. 2016 ACM SIGSAC Conf. Computer and Commun. Security*, ser. CCS '16, 2016, pp. 254–69; http://doi.acm.org/10.1145/2976749.2978309
[15] T. Chen *et al.*, "Understanding Ethereum via Graph Analysis," *Proc. IEEE INFOCOM 2018*, Apr. 2018, pp. 1484–92.

## Biographies

ZHICHAO ZHANG received his Bachelor's degree from Beijing University of Posts and Telecommunications, China, in 2017. He is working toward his Master's degree supervised by Prof. Ke Xu in the Department of Computer Science and Technology at Tsinghua University, Beijing, China. His research interests include federated learning, network security, and blockchain.

KE XU [M'02, SM'09] received his Ph.D. from the Department of Computer Science and Technology at Tsinghua University, where he serves as a full professor. He has published more than 100 technical papers and holds 20 patents in the research areas of next generation Internet, P2P systems, the Internet of Things, and network virtualization and optimization. He is a member of ACM and has guest edited several special issues in IEEE and Springer journals.

QI LI [M'12] received his B.Sc. and Ph.D. degrees in computer science and Technology from Tsinghua University in 2003 and 2012, respectively. His research interests include network architecture, protocol design, and system and network security.

XIN LIU is the chairman of Migu Culture Technology Co., Ltd. He has more than 20 years of experience in the mobile communications and Internet industries. He directs the business of Migu Culture Technology Co., Ltd, including platform development, big data, product design, and 5G.

LIN LI set up and managed four teams in Migu, located in Beijing, Nanjing, Chengdu, and Shanghai, respectively. The business he directed includes Internet platform development, product design, big data development, system operation, and troubleshooting. Meanwhile, he leads an international standardization team for W3C, 3GPP, GSMA, tech for 5G, and web applications.

BO WU received his Bachelor's degree from the School of Software at Shandong University, China, in 2014, and his Ph.D. degree from the Department of Computer Science and Technology at Tsinghua University in 2019. He is working in the 2012 Labs of Huawei Technologies. His research interests include network architecture, NetAI, network security, next generation Internet, and blockchain.

YUNZHE GUO received his Bachelor's degree from Sichuan University, China, in 2017. He is working toward his Master's degree at Tsinghua University. His research interests include deep learning and machine learning security.

Combined with blockchain technology, we can use smart contracts to establish equitable mechanisms to assess the contribution of participants, improving the model and providing incentives for them based on the assessment, which enables personal control of private training samples and amplifies the value of precious-but-unshareable data.