# Artificial Identification: A Novel Privacy Framework for Federated Learning Based on Blockchain

Liwei Ouyang, Fei-Yue Wang, *Fellow, IEEE*, Yonglin Tian, Xiaofeng Jia, Hongwei Qi, and Ge Wang

***Abstract*— To provide off-chain federations with complete privacy services to realize on-chain federated learning (FL), this article proposes a novel privacy framework for FL based on blockchain and smart contracts, named Artificial Identification. It consists of two modules: private peer-to-peer identification and private FL, using two scalable smart contracts to manage the identification and learning process, respectively. Based on Ethereum and interplenary file systems (IPFS), we implement our framework and comprehensively analyze its performance. Experiments show that the proposed framework has acceptable collaboration costs and offers advantages in terms of privacy, security, and decentralization. Furthermore, combined with radio frequency identification (RFID) technology, the framework has the potential to realize automatic on-chain identification and autonomous FL of machine clusters composed of Internet of Things (IoT) devices or distributed participants.**

***Index Terms*— Blockchain, federated learning (FL), private identification, radio frequency identification (RFID), smart contracts.**

## I. Introduction

FEDERATED learning (FL) is a new distributed machine learning architecture proposed by Google in 2016 for data islands problem [1]. The standard FL aims to learn a united statistical model on data stored on remote devices under the constraint that device-generated data are stored and processed locally, with only intermediate updates being communicated periodically with a central cloud or sever [2]. As it can train data from multiple parties without data leakage, FL has a promising application in fields where data cannot be directly aggregated due to factors such as intellectual property rights, privacy protection, and data security [3]. The exchange and fusion of local parameters determine the performance of FL; however, the centralized processing center has the potential risk of single point of failures, which may still lead to data loss, leakage, and tampering. Therefore, the FL community begins to seek the help of blockchain, an emerging decentralized architecture with the characteristics of de-trust, traceability, and tamper-resistance [4].

In the general process of FL based on blockchain, federated members upload their local models on the blockchain, download updates from others and locally fuse global federated models after verifications, thereby replacing centralized processing centers with distributed ledgers, tracing on-chain interaction records, and avoiding single point of failures. According to this idea, there are two typical design patterns in recent research: building a dedicated FL blockchain and developing smart contracts to manage the collaborative learning process. Since the design of programmable smart contracts is applicable to different types of blockchain, the latter is usually more scalable than the former. BlockFL [5] is a dedicated FL blockchain that is assumed to be open only to federated members, and thus without identification and data encryption mechanisms. However, once attackers find BlockFL, all data on the blockchain will be leaked immediately, including models and participants. Learning markets (LM) [6] based on smart contracts improves BlockFL by encrypting all private parameters to be transmitted. But it needs to publicly record all member lists on the blockchain for permission management, thereby introducing a new risk of de-anonymous attack. In a word, both BlockFL and LM can only provide on-chain FL with partial privacy protection. There is an urgent need for a new privacy framework to provide off-chain federations with complete privacy protection in the whole process from on-chain identification to collaborative learning.

To fill this critical gap, we propose a novel privacy framework for FL based on blockchain and smart contracts, named Artificial Identification. Our framework aims to provide complete privacy services for off-chain federations who want to realize on-chain FL. Specifically, our service consists of two parts: 1) private peer-to-peer (P2P) identification that can transform an off-chain federation into an on-chain form and 2) private FL that only shares, verifies, and fuses federated models within the federation. In this article, we implement and comprehensively analyze our framework based on two prevalent blockchain projects, i.e., Ethereum and inter-plenary file systems (IPFS). Experiments show that the proposed

Manuscript received 26 May 2021; revised 10 August 2021; accepted 11 August 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFB1600400, in part by the Science and Technology Development Fund, Macau SAR (File no. 0050/2020/A1) and in part by the National Natural Science Foundation of China under Grant U1811463. *(Corresponding author: Fei-Yue Wang.)*

Liwei Ouyang is with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China, and also with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.

Fei-Yue Wang is with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the Institute of Systems Engineering, Macau University of Science and Technology, Macau 999078, China (e-mail: feiyue.wang@ia.ac.cn).

Yonglin Tian is with the Department of Automation, University of Science and Technology of China, Hefei, Anhui 230027, China.

Xiaofeng Jia is with the Department of Data Management, Beijing Big Data Centre, Beijing 101100, China.

Hongwei Qi is with the Datang (Beijing) Technology Company, Ltd., Beijing 100192, China.

Ge Wang is with the Institute of Systems Engineering, Macau University of Science and Technology, Macau 999078, China.

Digital Object Identifier 10.1109/TCSS.2022.3226861

framework with acceptable collaboration costs has advantages in privacy, security, and decentralization. Furthermore, combined with radio frequency identification (RFID) technology, the framework has the potential to realize automatic on-chain identification and autonomous FL of machine clusters composed of Internet of Things (IoT) devices or distributed participants. As a preliminary attempt to provide FL with complete privacy services, our framework can be extended with more identification or learning mechanisms based on scalable smart contracts.

The remainder of this article is organized as follows. Section II reviews the basic concepts and recent advances of FL, blockchain and smart contracts, blockchain-based FL and RFID technology; Section III presents our framework of Artificial Identification, explains its detailed design of operation mechanisms and smart contracts, and describes its combination with RFID; Section IV implements our framework based on Ethereum and IPFS and analyzes its performance; Section V discusses our future work; and Section VI concludes.

## II. Literature Review

In this section, we briefly review the basic concepts and recent advances of FL, blockchain and smart contracts, blockchain-based FL, and RFID technology.

### A. Federated Learning

FL proposes a privacy-preserving mechanism in distributed machine learning systems [7]. With data stored locally and models shared globally, FL can achieve collaboration between multiple nodes, organizations, and companies. According to the distribution of feature and sample spaces in different data parties, FL can be grouped into horizontal FL, vertical FL, and federated transfer learning [3]. Horizontal FL aims at the collaboration between different samples that share similar features [8], [9]. And vertical FL focuses on the scenarios where the samples are same while the features are different [10], [11]. For more general scenarios where both the samples and features are almost different between different parties, transfer learning is integrated to give an effective solution named federated transfer learning [12]. As the core of FL systems, collaboration mechanisms play great importance in both the efficiency and the security of the federation. Many works have been proposed to reduce the communication costs [1], [13], [14] and strengthen the aggregation security [15], [16], [17], [18], [19] in the collaborative group. However, the centralized manner and the lack of incentive mechanisms in FL systems can still make them vulnerable and inefficient [6]. In this article, we propose to build FL architectures on blockchain systems to keep the stability and activeness of the collaboration process.

### B. Blockchain and Smart Contracts

Blockchain is a decentralized system that can achieve coordination and collaboration among multiple mutually distrustful nodes [20]. Originated in the bitcoin system [21], blockchain technology has been widely used in finance [22], industry [23], healthcare [24], [25], communication, and transportation areas [26], [27], [28]. Generally, a blockchain system can be divided into six stacked layers, including data layer, network layer, consensus layer, incentive layer, contract layer, and application layer [20]. Smart contracts are a set of self-execute and self-verify protocols which provide the interfaces for flexible software-defined services and promote the wide application of blockchain systems in various scenarios [29]. Toward the issues of privacy [30], [31], security [32], [33], [34], [35], [36], design and performance [37], [38], and formal verification [39], [40], many works have been proposed to analyze and strengthen the existing smart contracts. Besides, lots of tools and frameworks [40], [41] have been developed to analyze their security, validity, and other characteristics.

In this article, we implement our framework based on two blockchain systems, i.e., Ethereum [42] and IPFS [43]. Proposed by Vitalik Buterin in 2013, Ethereum has become one of the most popular open-source blockchain projects. Similar to the bitcoin system, it also develops its own cryptocurrency (Ether) but is much more flexible, benefiting from the Turing complete smart contracts. IPFS is a peer-to-peer hypermedia protocol that can identify and share files in a global space with all the devices connected. Driven by the Protocol Labs, it has attracted thousands of contributors globally. The distributed and content-addressing manners used in IPFS make it a secure and robust system to avoid attacks and damage to files.

### C. Blockchain-Based FL

In complex scenarios with unknown security, FL can be vulnerable and unstable due to the accidents such as potential attacks on the message propagation process and unreliable nodes. Introducing blockchain into FL is promising to solve these problems [44], [45]. First, blockchain can be used as the secure bridge between federated nodes and the center for local model uploading and global model downloading. Second, the incentive mechanisms in blockchain can be used to improve the activeness of the federation. Many works have proposed ways in which the blockchain and FL are combined [46], [47], [48], [49]. The insecurity and privacy leakage in the FL communication process are discussed in [46], and blockchain is proposed as the media for information exchange. Considering possible malicious nodes in FL, BlockFL adds the consensus process to validate the effectiveness of the local models from different nodes [47]. To avoid the breakdown of FL caused by the failure of centralized servers, FLchain proposes to aggregate local models on a dedicated blockchain network which can greatly improve the robustness of federation [48]. Considering the reliability and viciousness of federated members, Kang et al. [49] propose the concept of reputation to measure the reliability of federated nodes. They build a new incentive mechanism to encourage high-quality nodes by combining reputation with smart contracts. Recently, LM builds an auditable and traceable AI market based on the blockchain system to support verified AI resource sharing and distributed machine learning like FL without mutual trust [6]. However, all above frameworks lack safe on-chain identification mechanisms and fail to protect off-chain federations' member lists. In this article, we propose a novel identification mechanism to transform off-chain federations into on-chain

FL collaborations, which protects the information of related members and promotes secure collaboration between valid nodes.

### D. Radio Frequency Identification

A typical radio frequency identification (RFID) system is usually comprised of three parts, i.e., electronic tags used for encoding object-level information, RFID readers used for decoding information from tags, and the information management system. RFID technology is widely used in industry, food area, retail business, and IoT systems due to the low cost, flexible usage, small size, and ease of use [50], [51], [52], [53]. For industrial applications, an emergency management system is built with RFID in [54]. An outdoor localization algorithm is designed with RFID and GPS data in [55]. To monitor the states of objects during the manufacturing process, RFID is used to develop a manufacturing execution system [56]. RAPShell proposes a sharing mechanism of production information with RFID to promote the cooperation of decision-makers in the planning and scheduling system [57]. Similarly, RFID-based monitoring systems are built in [58], [59], and [60] to monitor the production process in distributed scenarios. For IoT applications, RFID is regarded as the key component for automatic identifying, tracking, and monitoring of objects worldly [61], [62]. Driven by the advantages of RFID technology, we combine it to bridge the gap between physical organizations and on-chain collaborative federations.

## III. Artificial Identification

In this section, we explain our proposed framework from aspects of framework overview, operation mechanisms, and detailed design of smart contracts. Also, we discuss our potential combination with RFID.

### A. Framework Overview

Our framework aims to provide off-chain federations with complete privacy services to achieve on-chain FL. Therefore, we assume that before using our framework, the off-chain federations have been formed, and some necessary interaction information has been negotiated and shared within the federations. In this context, our proposed framework is shown in Fig. 1. The framework is running on an underlying blockchain network, and consists of two modules, namely, private P2P identification in (a) and private FL in (b). The former helps federated members with privacy protection requirements to transform their off-chain federation into an on-chain form, and the latter helps them to realize the subsequent collaborative FL. The main functions of the two modules are respectively implemented by two smart contracts, i.e., identification smart contract (ISC) and collaborative training smart contract (CTSC), and they are jointly programmed, reviewed, and deployed by all federated members before all operations start. By expanding the programmable ISC and CTSC, our framework is expected to support more customized identification workflows and collaborative learning modes, and this article only demonstrates a simple illustrative implementation. Some concepts are introduced as follows.

*Blockchain Account*: denoted by acc and associated with a pair of public key and private key $\{\mathrm{pk}_{\mathrm{acc}}, \mathrm{sk}_{\mathrm{acc}}\}$. A blockchain account represents a participant and its value is a blockchain address derived from $\mathrm{pk}_{\mathrm{acc}}$.

*Federated Members*: denoted by $F$. A member of an off-chain federation who has registered in the blockchain network with assigned $\{\mathrm{pk}_{F_i}, \mathrm{sk}_{F_i}\}$. The pre-shared interaction information includes the federal account $\mathrm{acc}_{\mathrm{FE}}$ with its corresponding $\{\mathrm{pk}_{\mathrm{FE}}, \mathrm{sk}_{\mathrm{FE}}\}$, the deployment address of ISC and CTSC, i.e., $\mathrm{add}_{\mathrm{ISC}}$ and $\mathrm{add}_{\mathrm{CTSC}}$, and the scale of off-chain federation $N$. To protect anonymity and privacy, members will not directly share their blockchain accounts off the blockchain, but have to invoke ISC to identify each other on the blockchain.

*Federal Account*: denoted by $\mathrm{acc}_{\mathrm{FE}}$. A blockchain account negotiated by a federation whose $\{\mathrm{pk}_{\mathrm{FE}}, \mathrm{sk}_{\mathrm{FE}}\}$ is shared among all federated members. $F$s broadcast messages within the federation by sending messages to the federal account and listening to messages from the federal account.

*TrustList*: If $F_i \in \mathrm{TrustList}_{F_j}$, it means $F_i$ has passed $F_j$'s P2P identification. $F_j$ admits that $F_i$ is a member of the federation. We stipulate that trust is mutual, namely, if $F_i \in \mathrm{TrustList}_{F_j}$, then $F_j \in \mathrm{TrustList}_{F_i}$. The final $\mathrm{TrustList}_{\mathrm{FE}}$ merged from the consensus of all federated members is our desired member list of the on-chain federation.

*ActiveList*: If $F_i \in \mathrm{ActiveList}_{F_j}$, it means $F_i$ has published the consensused $\mathrm{TrustList}_{\mathrm{FE}}$ merged by $F_j$, and $F_j$ believes that $F_i$ is an active participant of the subsequent collaborative FL. Every TrustList corresponds to a unique ActiveList, and $\mathrm{TrustList}_{\mathrm{FE}}$ corresponds to $\mathrm{ActiveList}_{\mathrm{FE}}$.

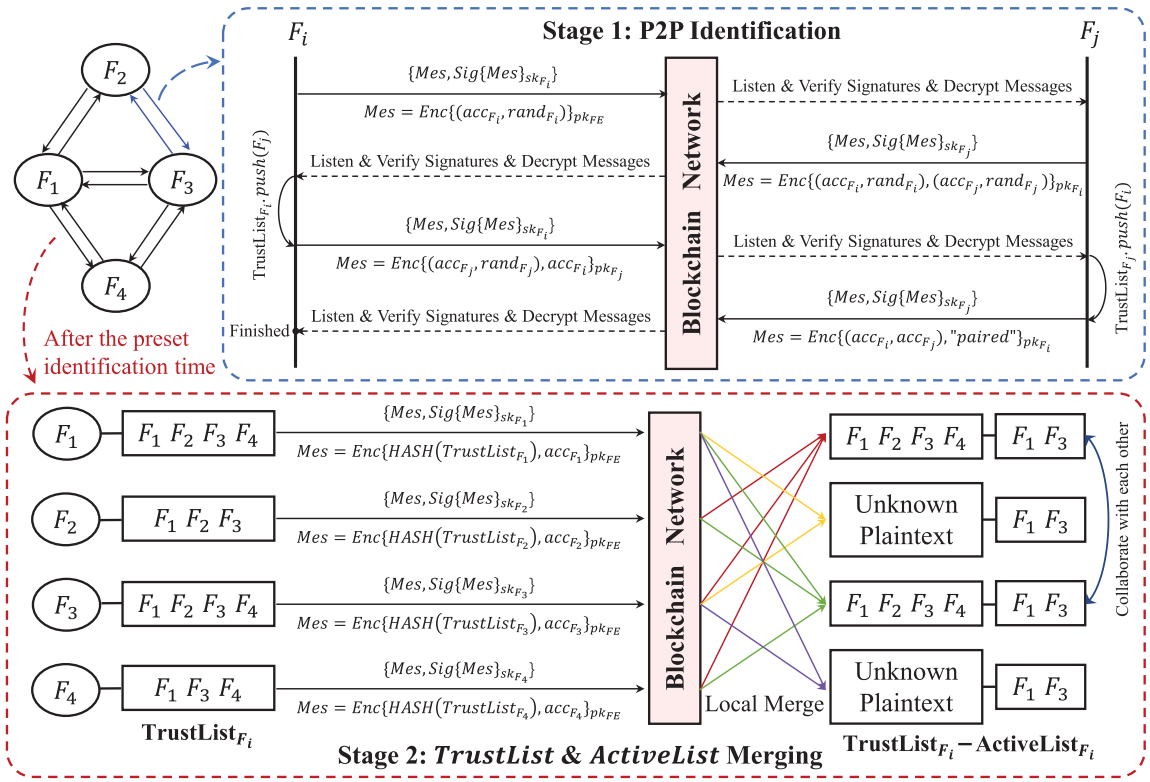
Additionally, to reduce the storage burden of the blockchain network and ensure communication security, private files such as datasets and models are encrypted with the owner's self-defined symmetric key $k_{\mathrm{owner}}$ and stored on IPFS. The access path Path(file) composed of the obtained IPFS hash and $k_{\mathrm{owner}}$ is recorded as formula (1). And federated members exchange these private files by sharing their access paths. Also, we require senders to broadcast and send messages with their digital signatures in the form of $\{\mathrm{Mes}, \mathrm{Sig}\{\mathrm{Mes}\}_{\mathrm{sk}_{\mathrm{sender}}}\}$, and this applies in all sections unless otherwise specified

$$\mathrm{Path(file)} = \{\mathrm{IPFS}\{\mathrm{Enc}\{\mathrm{file}\}_{k_{\mathrm{owner}}}\}, k_{\mathrm{owner}}\}. \tag{1}$$

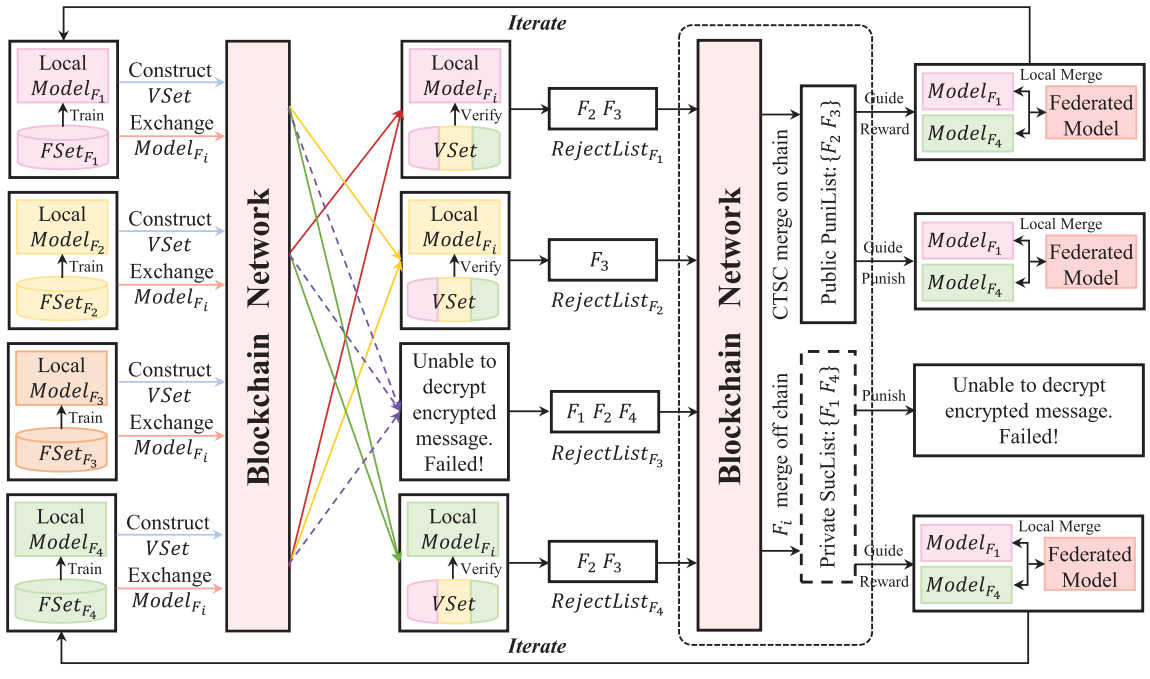
Both in Fig. 1(a) and (b), we take the interaction among four federated members as an example to show our framework, but their federated members do not correspond to each other. The detailed operation mechanisms and contract design are explained in Sections III-B and III-C.

### B. Private P2P Identification

This section aims to transform the off-chain federation into the on-chain form. Before the on-chain identification starts, the initial information held by a federated member $F_i$ is $\{\mathrm{pk}_{\mathrm{FE}}, \mathrm{sk}_{\mathrm{FE}}, \mathrm{pk}_{F_i}, \mathrm{sk}_{F_i}, \mathrm{add}_{\mathrm{ISC}}, \mathrm{add}_{\mathrm{CTSC}}, N\}$. Since the pre-shared interaction information is regarded as the only proof of identity, all participants who know the complete $\{\mathrm{pk}_{\mathrm{FE}}, \mathrm{sk}_{\mathrm{FE}}, \mathrm{add}_{\mathrm{ISC}}, \mathrm{add}_{\mathrm{CTSC}}, N\}$ will be identified as federated members in our design. The private P2P identification has two stages: P2P identification and TrustList & ActiveList merging.




(a)




(b)

Fig. 1. Framework of Artificial Identification. (a) Private P2P identification. (b) Private FL.

According to Fig. 1(a), their detailed operation mechanisms are as follows.

*1) P2P Identification:*

1) A federated member $F_i$ broadcasts his/her puzzle, i.e., rand number $\text{rand}_{F_i}$ and $\text{acc}_{F_i}$, within the federation by sending $\text{Enc}\{(\text{acc}_{F_i}, \text{rand}_{F_i})\}_{\text{pk}_{\text{FE}}}$ to $\text{acc}_{\text{FE}}$. At the same time, $F_i$ listens and responds to puzzles from other members.
2) Another federated member $F_j$ listens and responds to $F_i$: $F_j$ verifies and decrypts $F_i$'s puzzle, if the digital signature is correct and the sender is $\text{acc}_{F_i}$ encrypted in $F_i$'s puzzle, $F_j$ sends $\text{Enc}\{(\text{acc}_{F_i}, \text{rand}_{F_i}), (\text{acc}_{F_j}, \text{rand}_{F_j})\}_{\text{pk}_{\text{F}_i}}$ to $F_i$.

3) $F_i$ verifies and decrypts the returned message, if the digital signature and $\text{rand}_{F_i}$ are correct and the sender is $\text{acc}_{F_j}$ encrypted in the message, $F_i$ adds $F_j$ to $\text{TrustList}_{F_i}$, and sends $\text{Enc}\{(\text{acc}_{F_j}, \text{rand}_{F_j}), \text{acc}_{F_i}\}_{\text{pk}_{F_j}}$ to $F_j$.
4) $F_j$ verifies and decrypts the returned message. If the digital signature and $\text{rand}_{F_j}$ are correct and the sender is $\text{acc}_{F_i}$ encrypted in the message, $F_j$ adds $F_i$ to $\text{TrustList}_{F_j}$, and sends $\text{Enc}\{(\text{acc}_{F_i}, \text{acc}_{F_j})$, "paired!"$\}_{\text{pk}_{F_i}}$ to $F_i$.
5) $F_i$ verifies and decrypts the returned message. If the digital signature and "paried" message are correct, P2P identification between $F_i$ and $F_j$ finishes, and $F_i$ and $F_j$ build mutual trust. Otherwise, $F_i$ deletes $F_j$ from $\text{TrustList}_{F_i}$, their P2P identification fails.

For federated members, whether they choose to actively broadcast puzzles or passively respond to puzzles, they will enter the above process and finally build mutual trust with others. Two members only need to complete P2P identification once, and the order of broadcast and response does not affect the identification result. Moreover, lazy members who neither broadcast nor respond cannot establish their own TrustList or be added to another TrustList, thereby being automatically excluded. After the end of this stage, every federated member locally maintains a TrustList. They add their own accounts to their TrustList, and only if $\text{len(TrustList)} \leq N$, they go to the next stage. Otherwise, they stop all collaboration.

*2) TrustList and ActiveList Merging:*

1) Every $F_i$ sorts the accounts in the $\text{TrustList}_{F_i}$ in dictionary order (or in other same orders), computes the hash digest of the sorted $\text{TrustList}_{F_i}$, and broadcasts within the federation by sending $\text{Enc}\{\text{Hash}(\text{TrustList}_{F_i}), \text{acc}_{F_i}\}_{\text{pk}_{\text{FE}}}$ to $\text{acc}_{\text{FE}}$. At the same time, $F_i$ listens and compares hash digests from other members in $\text{TrustList}_{F_i}$.
2) For every $F_j \in \text{TrustList}_{F_i}$, $F_i$ first listens and verifies their encrypted messages. If their digital signatures are correct and the senders are $\text{acc}_{F_j}$s encrypted in the messages, $F_i$ compares their hash digests to find the consensused hash digest, and takes its corresponding TrustList as $\text{TrustList}_{\text{FE}}$. Additionally, $F_i$ adds all the members who have published the merged $\text{TrustList}_{\text{FE}}$ to his/her locally maintained $\text{ActiveList}_{F_i}$. At last, $F_i$ derives the new shared key $k_{\text{Active}}$ for collaborative FL among $\text{ActiveList}_{F_i}$ by sequentially splicing the $\text{rand}_{F_j}$, $F_j \in \text{TrustList}_{\text{FE}}$.

Ideally, at the end of P2P identification, all federated members have identified with each other, and thus all $\text{TrustList}_{F_i} = \text{TrustList}_{\text{FE}}$ and all $\text{ActiveList}_{F_i} = \text{ActiveList}_{\text{FE}}$. Especially, in Fig. 1(a), we show how our framework works in a potentially nonideal situation where $F_2$ and $F_4$ do not mutually identify due to latency or other faults. Although they belong to the correct $\text{TrustList}_{\text{FE}}$ merged by $F_1$ and $F_3$, they do not know the full content (plaintext) of this $\text{TrustList}_{\text{FE}}$ and the corresponding $k_{\text{Active}}$, so they cannot collaborate with $F_1$ and $F_3$ later. On the contrary, $\text{ActiveList}_{F_i}$ and $\text{ActiveList}_{\text{FE}}$ are almost overt within the federation, because they can be easily derived by $F_2$ and $F_4$, and that is why $k_{\text{Active}}$ cannot be created based on $\text{ActiveList}_{F_i}$ or $\text{ActiveList}_{\text{FE}}$.

Finally, we discuss the design of ISC. To improve privacy, we do not store and compare any TrustList and ActiveList on the blockchain, and all the collection and consensus operations are performed locally by federated members. Therefore, ISC only has three main functions to broadcast puzzles, send messages to a specific receiver, and broadcast TrustLists. Both puzzles and TrustLists can only be broadcast once by one account in the preset time, so as to avoid the denial of service attack, exclude inactive members and realize automatic control of two stages. Since our framework is implemented based on Ethereum, these three functions transmit messages by triggering and monitoring "Event" to reduce the costs of on-chain storage. At the end of private P2P identification, all honest and active federated members additionally held the same $\text{TrustList}_{\text{FE}}$, $\text{ActiveList}_{\text{FE}}$, and $k_{\text{Active}}$.

### C. Private FL

This section aims to help federated members in $\text{ActiveList}_{\text{FE}}$ realize private FL with fair rewards and punishments. According to Fig. 1(b), the detailed operation mechanisms are as follows.

1) In the preset registration time, every $F_i \in \text{ActiveList}_{\text{FE}}$ reports $\text{len}(\text{ActiveList}_{\text{FE}})$ to CTSC with deposit $D_r$ for registration. CTSC computes the consensused $\text{len}(\text{ActiveList}_{\text{FE}})$ after the registration time, every $F_i \in \text{ActiveList}_{\text{FE}}$ adds accounts who have reported the correct $\text{len}(\text{ActiveList}_{\text{FE}})$ but not belong to $\text{ActiveList}_{\text{FE}}$ to his/her locally maintained $\text{RejectList}_{F_i}$.
2) In the preset VSet construction time, every $F_i \in \text{ActiveList}_{\text{FE}}$ exchanges validation set $\text{VSet}_{F_i}$ sampled from privately-held dataset in $\text{ActiveList}_{\text{FE}}$: $F_i$ sends $\text{Enc}\{\text{Enc}\{\text{Path}(\text{VSet}_{F_i}), \text{acc}_{F_i}\}_{k_{\text{Active}}}\}_{\text{pk}_{\text{FE}}}$ to $\text{acc}_{\text{FE}}$. At the same time, he/she listens, verifies, downloads and combines all other $\text{VSet}_{F_j}$s from $F_j \in \text{ActiveList}_{\text{FE}}$ to construct a unified validation set VSet locally.
3) In each iteration,
   a) in the preset verification time, every $F_i \in \text{ActiveList}_{\text{FE}}$ exchanges his/her local model $\text{Model}_{F_i}$ trained on privately-held dataset and its performance $E_{F_i}$ on VSet in $\text{ActiveList}_{\text{FE}}$: $F_i$ sends $\text{Enc}\{\text{Enc}\{\text{Path}(\text{Model}_{F_i}), E_{F_i}, \text{acc}_{F_i}\}_{k_{\text{Active}}}\}_{\text{pk}_{\text{FE}}}$ to $\text{acc}_{\text{FE}}$. At the same time, $F_i$ listens and verifies all other $\text{Model}_{F_j}$s from $F_j \in \text{ActiveList}_{\text{FE}}$, if $F_j$ uploads false models or does not upload models for two consecutive rounds, $F_i$ adds $F_j$ to $\text{RejectList}_{F_i}$, and reports $\text{RejectList}_{F_i}$ to CTSC after finishing all verifications.
   b) after the preset verification time, CTSC merges all $\text{RejectList}_{F_i}$s from $F_i$ who have reported consensused $\text{len}(\text{ActiveList}_{\text{FE}})$, and counts $\text{RJ}_{F_i}$, the number of times that $F_i$ is rejected. If $\text{RJ}_{F_i}$ is greater than the preset threshold, CTSC adds $F_i$ to the public PuniList, otherwise every $F_i \in \text{ActiveList}_{\text{FE}}$ adds $F_i$ to their local private SucList.
   c) every $F_i \in \text{ActiveList}_{\text{FE}}$ locally fuses the federated model $\text{Model}_{\text{FE}}$ of this round merely based on

$\text{Model}_{F_i}$, $F_i \in$ SucList, and goes to the next iteration.

4) Until the preset stop conditions are met, the training of $\text{Model}_{\text{FE}}$ is completed. CTSC confiscates deposits of $F_i \in$ PuniList and rewards to $F_i \in$ SucList.

Generally, the security of on-chain collaboration is ensured by the function call permissions preset in smart contracts based on member lists. However, in this article, member lists like $\text{ActiveList}_{\text{FE}}$ and $\text{TrustList}_{\text{FE}}$ cannot be publicly accessed and stored, so as an alternative, CTSC has to set strict cooperation time for each stage to manage collaboration process and avoid single point of failures. And this will bring some malicious behaviors we need to consider and solve.

First, although $\{\text{pk}_{\text{FE}}, \text{sk}_{\text{FE}}, \text{add}_{\text{ISC}}, \text{add}_{\text{CTSC}}, N\}$ as the only identity proof will not be actively disclosed, transparent and public CTSC may still be accidentally invoked by participants outside the federation (outsiders). To eliminate their influences, we require members to report $\text{len}(\text{ActiveList}_{\text{FE}})$ for registration and only those who know the correct $\text{len}(\text{ActiveList}_{\text{FE}})$ can report $\text{RejectList}_{F_i}$ to CTSC. Thus, these unexpected intruders without the correct $\text{len}(\text{ActiveList}_{\text{FE}})$ will not really affect $\text{Model}_{\text{FE}}$, and they can get back all their deposits in the end. Then, for members with the correct $\text{len}(\text{ActiveList}_{\text{FE}})$, there are two kinds of malicious behaviors. One is like $F_2$ and $F_4$ in Fig. 1(a) who can derive the correct $\text{len}(\text{ActiveList}_{\text{FE}})$ but not the $k_{\text{Active}}$, so they can report $\text{RejectList}_{F_i}$ to CTSC, but cannot decrypt the encrypted data and models, which corresponds to $F_3$ in Fig. 1(b). The other extreme case is that in small-scale collaboration, malicious members may decipher the correct $\text{TrustList}_{\text{FE}}$ and $k_{\text{Active}}$ in an exhaustive way, and then they can easily obtain the shared data and models. However, because they are not really identified by $\text{ActiveList}_{\text{FE}}$, their data and models will never be accepted in the collaborative training, and thus $\text{Model}_{\text{FE}}$ is not the best model for them.

To limit and punish the latter two kinds of malicious behaviors, we stipulate in CTSC that every $F_i$ can only submit a $\text{RejectList}_{F_i}$ once in each iteration, and all $F_i \in \text{ActiveList}_{\text{FE}}$ should actively monitor members who have reported the correct $\text{len}(\text{ActiveList}_{\text{FE}})$ but are not identified in $\text{ActiveList}_{\text{FE}}$, and add them to their $\text{RejectList}_{F_i}$. Since it is difficult to distinguish whether the members in the $\text{RejectList}_{F_i}$ and PuniList are malicious members or active members who have published false models [like $F_2$ in Fig. 1(b)], the privacy of $\text{ActiveList}_{\text{FE}}$ and the real honest members is still protected. Considering that members in the final PuniList may affect collaboration results, they will be forfeited all their deposits, and these forfeiture deposits $D_{\text{Puni}}$ will be equally rewarded to all honest members in $\text{ActiveList}_{\text{FE}}$.

Finally, we discuss the design of CTSC. CTSC consists of five main functions for registration, exchanging $\text{VSet}_{F_i}$, exchanging $\text{Model}_{F_i}$, reporting $\text{RejectList}_{F_i}$ and withdrawing deposits after collaboration. All of them can only be invoked in the preset callable time, and the functions of reporting $\text{RejectList}_{F_i}$ and withdrawing deposits can only be invoked once by one account, so as to protect the security of $\text{Model}_{\text{FE}}$ and funds. Similar to ISC, the functions of exchanging $\text{VSet}_{F_i}$ and exchanging $\text{Model}_{F_i}$ transmit messages by triggering and monitoring "Event." At the end of private FL, members in $\text{ActiveList}_{\text{FE}}$ can obtain ideal $\text{Model}_{\text{FE}}$, and be fairly rewarded or punished in the form of cryptocurrency.

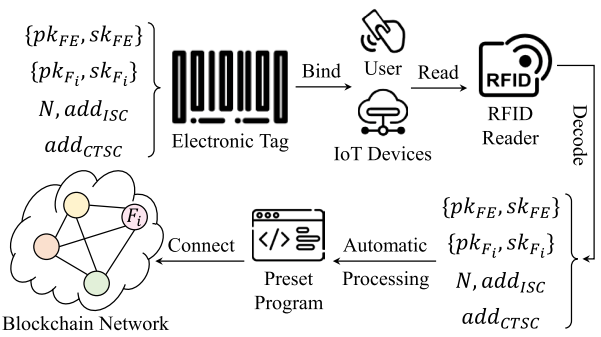



Fig. 2. Automatic on-chain identification process based on RFID and our proposed framework.

### D. RFID-Based Automatic On-Chain Identification and Collaboration

As aforementioned, our framework is enabled based on programmable smart contracts, which can serve as software-defined agents to encapsulate and process scalable identification mechanisms and collaboration relationships. Therefore, by encapsulating these standard smart contracts and their corresponding interaction programs on trusted hardware or machines, we can combine RFID technology to realize automatic on-chain identification and autonomous FL of machine clusters composed of IoT devices or distributed participants.

Fig. 2 shows the automatic on-chain identification process based on RFID and our framework. All initial information, i.e., $\{\text{pk}_{\text{FE}}, \text{sk}_{\text{FE}}, \text{pk}_{F_i}, \text{sk}_{F_i}, \text{add}_{\text{ISC}}, \text{add}_{\text{CTSC}}, N\}$, is encoded in electronic tags and bound with off-chain machine clusters composed of IoT devices or distributed participants in the form of RFID stickers or RFID cards. These off-chain trusted hardware or machines are preset with standard processing programs. They first read and decode their tag information with RFID readers, and then automatically interact with smart contracts to complete the private P2P identification described in Section III-B and realize the on-chain autonomous FL described in Section III-C.

## IV. Experiments

In this section, we develop the smart contracts designed in Section III to implement the proposed framework and examine its advantages from four aspects, including collaboration costs, privacy, security, and decentralization.

### A. Platform and Setup

Based on the development architecture Ganache + Truffle, we create a virtual Ethereum blockchain locally and develop the designed smart contracts using Solidity programming language to implement the proposed framework. Web3.js is used to interact with smart contracts and verify their performance, JavaScript libraries eth-ecies and crypto-js are used to support various cryptography algorithms, and Python + Keras is used to implement training, prediction, and fusion algorithms of AI models. Additionally, files of models or model weights are saved using the Keras.model library, and then stored or

TABLE I
DEPLOYMENT COSTS OF SMART CONTRACTS

| **Smart Contracts** | Helper | ISC | CTSC |
|---|---|---|---|
| **Deployment Costs (gas)** | 347 439 | 799 855 | 1 646 441 |

shared on IPFS. To reduce the execution costs, we extract the common functions in ISC and CTSC to form a *Helper* contract for others to call, including querying a specific member in a specific list and querying the consensused len(ActiveList$_{\mathrm{FE}}$). Therefore, three contracts are deployed, i.e., Helper, ISC, and CTSC.

The experimental scenario is assumed corresponding to Fig. 1(b). Five federated members with privacy protection requirements transform their off-chain federation into an on-chain form for the on-chain FL. Among them, only $F_3$ and $F_5$ do not mutually identify, so they do not belong to ActiveList$_{\mathrm{FE}}$ but can derive ActiveList$_{\mathrm{FE}}$. Since $F_3$ and $F_5$ have the same situation, we omit $F_5$ in Fig. 1(b). Also, we assume that $F_2$ submits a false model in the iteration, and $F_1$ and $F_4$ reject $F_2$. Additionally, due to the inherently limited computing ability of Ethereum smart contracts, such as only supporting 8–256 signed or unsigned integer variables, and lacking complete mathematical libraries, we inherit a verified contracts library named SafeMath for preventing overflow. Considering that there is no list of honest members to query, all functions in ISC and CTSC are preset with strict callable time.

### B. Evaluation and Discussion

*1) Collaboration Costs:* According to the preset rules in Ethereum Yellow Paper [42], any execution of programming segments in Ethereum will trigger a payment for computing resources calculated using the unit called gas. The actual cryptocurrency payment of a transaction is $\mathrm{Cost}_{\mathrm{Eth}} = \mathrm{gas} * \mathrm{gasPrice}$, where gasPrice is the exchange rate between Ethereum's cryptocurrency Eth and gas, and can be set arbitrarily by senders. The higher the gasPrice, the faster the transaction will be packaged into the blockchain. The gas consumption of our smart contract deployments and executions are shown in Tables I and II, respectively. Since the order of invoking, the complexity of calculations, and the size of parameters will all affect the gas consumption, the data in Tables I and II are the maximum values we observe in the experiment. For instance, the function of registration consumes the most gas (102 765) on the first call to create a new list and consumes much less gas (87 765) on subsequent calls, similar to broadcasting puzzles and TrustList$_{F_i}$. Also, $F_1$, $F_2$, $F_3$ and $F_4$ may consume different amounts of gas when calling the same function for different computational complexity, such as publishing RejectList$_{F_i}$ (157 459 / 85 392 / 171 030 / 216 678) and withdrawing deposits (93 024 / 61 257 / 61 873 / 79 457). And the costs of functions that only transmit messages mainly depend on the size of messages, such as sending a message, exchanging VSet$_{F_i}$ and Model$_{F_i}$. Hence, cryptography algorithms with shorter ciphertexts are preferred to reduce gas consumption. In a word, the execution costs of smart contracts are not fixed values, and we choose the observed maximums as a representation.

At the maximum estimation, the gas of $N$ federated members to complete P2P identification can be computed as formula (2). Taking gasPrice $= 2 \times 10^{10}$ wei as an example, three federated members should cost 1 100 712 gas for identification, i.e., 0.022 Eth. According to Eth's historical highest and lowest prices from April 2016 to August 2021, 1 Eth = \$4366.10 and 1 Eth = \$5.86 [63], the corresponding payments are \$96.05 and \$0.13. The former is almost unacceptable, while the latter can be considered in comparison to centralized identification. The actual costs will be far less than this maximum estimation and can be further reduced with the future upgrade of the Ethereum platform. Also, to deal with the concerns that the P2P identification costs increase rapidly with the expansion of the collaboration scale, more efficient identification workflows are expected to be introduced

$$94\,699 \times N + 57\,804 \times 3 \times C_N^2 + 98\,793 \times N = 86\,706N^2 + 10\,6786N. \quad (2)$$

Another collaboration costs we consider is the computational time. Besides the original model training time, the computational time includes the contract execution time to realize on-chain interactions. Since the contract executions take effect only when the new block is added to the main chain, transactions sent asynchronously by distributed participants but packaged into the same block can be regarded as having the same computational time, i.e., block time [64]. Therefore, taking the private P2P identification as an example, assuming that the time differences among all $F$s to broadcast puzzles, send messages, and broadcast TrustList$_{F_i}$ are less than one block time, and their transactions are immediately packaged into the same added block without pending. Then, the additional computational time is five block times. Similarly, that of one iteration is two block times. Referring to the existing Ethereum, one block time is about 13 s, and the five block times are about 65 s. Also, the average block time can be flexibly customized in the personal blockchain.

*2) Privacy:* Our framework improves the privacy of FL based on blockchain in two aspects: identification and collaborative training. For identification, methods of sharing blockchain accounts off the chain and recording honest member lists in smart contracts are conducive to reducing collaboration costs and managing interactive permissions. However, malicious participants can easily launch a de-anonymous attack to correspond the blockchain accounts with their real identities and monitor federated members' transactions in the long term. To solve this problem, honest member lists in this article are maintained locally and compared in the form of hash digests. Thus, it is difficult for attackers to decipher the correct plaintexts unless they actually finish the private P2P identification. As for the training, although the interaction records can be traced publicly, all messages are encrypted before being transmitted, and the function names in Section III can be independent of their functions. Therefore, in the open blockchain network, it is difficult for outsiders to monitor the behaviors of specific accounts purposefully. In addition, for the members in the PuniList, their additional punishments include the potential privacy leaks, even if outsiders can

TABLE II
EXECUTION COSTS OF SMART CONTRACTS

| | | | |
|---|---|---|---|
| **ISC** | Broadcast Puzzle | Send Message | Broadcast $TrustList_{F_i}$ |
| | 94 699 | 57 804 | 98 793 |
| **CTSC** | Register | Exchange $VSet_{F_i}$ | Exchange $Model_{F_i}$ |
| | 102 765 | 63 038 | 63 516 |
| | Report $RejectList_{F_i}$ | Withdraw Deposit After Collaboration | |
| | 216 678 | 93 024 | |

hardly distinguish whether they are real federated members. It should be noted that the privacy of this part is relative to the federation, which only guarantees that outsiders cannot know the real $\text{TrustList}_{\text{FE}}$ and $\text{ActiveList}_{\text{FE}}$. Members in the federation with identity proof can still infer the $\text{ActiveList}_{\text{FE}}$.

*3) Security:* First, we consider the security of $\text{Model}_{\text{FE}}$. We already discuss three malicious behaviors and their corresponding solutions in Section III-C. Based on our design, the unexpected intruders will not affect the final $\text{Model}_{\text{FE}}$, and the members who can submit $\text{RejectList}_{F_i}$ and affect $\text{Model}_{\text{FE}}$ will be added to PuniList and forfeited all their deposits. Moreover, their data and models will not be accepted, and $\text{Model}_{\text{FE}}$ is not ideal for them. For better security, we can also detect and eliminate malicious behavior in advance by extending smart contracts to bind an evaluable and traceable credit score for each participant. This mechanism has been discussed and implemented in our previous works [6], [64].

Another security issue that needs to be considered is that some members may still need to be mutually identified due to network latency or other faults after the end of the preset time. We discuss one case of this problem in Fig. 1(a). Here we examine a more extreme case. Because we set that every $F_i$ can only maintain and broadcast one $\text{TrustList}_{F_i}$, $F_i$ cannot join two different TrustLists or ActiveLists at the same time, nor belong to two different sub federations. However, it is possible to have multiple completely disjoint subfederations at the same time, such as $\{F_1, F_2, F_3\}$ and $\{F_4, F_5, F_6\}$. At this time, they can complete FL in the sub-training clusters, and then fuse the obtained federated models after extending the identification in the subsequent collaboration.

*4) Decentralization:* The traditional centralized identification mode usually relies on a certification authority to issue and manage digital certificates, while the traditional FL usually relies on a centralized cloud or server to fuse federated models. Once these centralized devices or organizations are attacked, there will be the risks of single point of failures and privacy leakage. To solve this problem, both identification and FL modes proposed in this article are decentralized. Specifically, all identifications are completed peer-to-peer, all honest member lists are maintained locally, and all federated models are fused distributedly.

## V. FUTURE WORK

Our work can be extended from three aspects. First, support the flexibility of federations. Currently, for better collaboration security and efficiency, our framework encourages federated members to remain active rather than join or quit halfway. Therefore, once off-chain federations change, they need to restart private P2P identification immediately. However, the dynamic change of members may be a potential demand in some application scenarios, so efficient re-identification mechanisms are required. Second, avoid the disclosure of pre-shared information. Since identity proofs will not be actively disclosed is a strong assumption, our framework adopts two strategies to reduce the probability of identity theft, including checking the collaboration scale and punishing inactive members. Thus, malicious participants can hardly infiltrate and have to contribute. In the future, more incentive and behavior-tracking mechanisms can be utilized to reduce the motivation for evil. Finally, optimize collaboration costs and apply the framework in practical scenarios. Our framework has broad application scenarios, and we can further expand it in the fields of IoT, finance, and healthcare.

## VI. CONCLUSION

In this article, Artificial Identification, a novel privacy framework for FL based on blockchain and smart contracts, is proposed to provide complete privacy services for off-chain federations that want to realize on-chain FL. Artificial Identification manages identification and learning process based on scalable smart contracts, and consists of two modules: private P2P identification and private FL. We implement and comprehensively analyze the proposed framework based on Ethereum and IPFS. The experiments show that our framework with acceptable collaboration costs has advantages in privacy, security and decentralization. Furthermore, as a preliminary attempt, the framework has the potential to be extended with more identification or learning mechanisms, and combined with RFID technology to realize automatic on-chain identification and autonomous FL of machine clusters composed of IoT devices or distributed participants.

## REFERENCES

[1] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2016, *arXiv:1610.05492*.
[2] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.
[3] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol. (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.
[4] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Comput. Ind. Eng.*, vol. 149, Nov. 2020, Art. no. 106854.
[5] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Blockchained on-device federated learning," *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1279–1283, Jun. 2020.
[6] L. Ouyang, Y. Yuan, and F.-Y. Wang, "Learning markets: An AI collaboration framework based on blockchain and smart contracts," *IEEE Internet Things J.*, vol. 9, no. 16, pp. 14273–14286, Aug. 2022.

[7] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[8] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proc. 53rd Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2015, pp. 1310–1321.

[9] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," 2016, *arXiv:1602.05629*.

[10] S. Hardy et al., "Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption," 2017, *arXiv:1711.10677*.

[11] R. Nock et al., "Entity resolution and federated learning get a federated resolution," 2018, *arXiv:1803.04035*.

[12] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Jan. 2021.

[13] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," 2017, *arXiv:1712.01887*.

[14] S. Caldas, J. Konečny, H. B. McMahan, and A. Talwalkar, "Expanding the reach of federated learning by reducing client resource requirements," 2018, *arXiv:1812.07210*.

[15] K. Bonawitz et al., "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 1175–1191.

[16] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 5, pp. 1333–1345, May 2018.

[17] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," 2017, *arXiv:1712.07557*.

[18] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," 2017, *arXiv:1710.06963*.

[19] A. Bhowmick, J. Duchi, J. Freudiger, G. Kapoor, and R. Rogers, "Protection against reconstruction and its applications in private federated learning," 2018, *arXiv:1812.00984*.

[20] Y. Yuan and F.-Y. Wang, "Blockchain: The state of the art and future trends," *Acta Autom. Sinica*, vol. 42, no. 4, pp. 481–494, 2016.

[21] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," White Paper. Accessed: Aug. 9, 2021. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[22] W. Viriyasitavat, L. D. Xu, Z. Bi, and V. Pungpapong, "Blockchain and Internet of Things for modern business process in digital economy—The state of the art," *IEEE Trans. Computat. Social Syst.*, vol. 6, no. 6, pp. 1420–1432, Dec. 2019.

[23] S. Zhao, S. Li, and Y. Yao, "Blockchain enabled industrial Internet of Things technology," *IEEE Trans. Computat. Social Syst.*, vol. 6, no. 6, pp. 1442–1453, Dec. 2019.

[24] S. Wang et al., "Blockchain-powered parallel healthcare systems based on the ACP approach," *IEEE Trans. Computat. Social Syst.*, vol. 5, no. 4, pp. 942–950, Dec. 2018.

[25] B. B. Gupta, K.-C. Li, V. C. M. Leung, K. E. Psannis, and S. Yamaguchi, "Blockchain-assisted secure fine-grained searchable encryption for a cloud-based healthcare cyber-physical system," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 12, pp. 1877–1890, Dec. 2021.

[26] Y. Yuan and F.-Y. Wang, "Towards blockchain-based intelligent transportation systems," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 2663–2668.

[27] Y. Yuan and F.-Y. Wang, "Blockchain and cryptocurrencies: Model, techniques, and applications," *IEEE Trans. Syst. Man, Cybern., Syst.*, vol. 48, no. 9, pp. 1421–1428, Sep. 2018.

[28] X. Huang, D. Ye, R. Yu, and L. Shu, "Securing parked vehicle assisted fog computing with blockchain and optimal smart contract design," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 2, pp. 426–441, Mar. 2020.

[29] L. Ouyang, S. Wang, Y. Yuan, X. Ni, and F. Y. Wang, "Smart contracts: Architecture and research progresses," *Acta Automatica Sinica*, vol. 45, no. 3, pp. 445–457, 2019.

[30] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2016, pp. 839–858.

[31] F. Zhang, E. Cecchetti, K. Croman, A. Juels, and E. Shi, "Town crier: An authenticated data feed for smart contracts," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 270–282.

[32] S. Badruddoja, R. Dantu, Y. He, K. Upadhayay, and M. Thompson, "Making smart contracts smarter," in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency (ICBC)*, May 2021, pp. 254–269.

[33] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on Ethereum smart contracts (SoK)," in *Proc. Int. Conf. Princ. Secur. Trust*. Cham, Switzerland: Springer, 2017, pp. 164–186.

[34] T. Chen, X. Li, X. Luo, and X. Zhang, "Under-optimized smart contracts devour your money," in *Proc. IEEE 24th Int. Conf. Softw. Anal., Evol. Reeng. (SANER)*, Feb. 2017, pp. 442–446.

[35] S. Meiklejohn et al., "A fistful of bitcoins: Characterizing payments among men with no names," in *Proc. Conf. Internet Meas. Conf.*, Oct. 2013, pp. 127–140.

[36] D. Ron and A. Shamir, "Quantitative analysis of the full bitcoin transaction graph," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Cham, Switzerland: Springer, 2013, pp. 6–24.

[37] Z. Shuai, Y. Yong, N. Xiao-Chun, and W. Fei-Yue, "Scaling blockchain towards bitcoin: Key technologies, constraints and related issues," *Acta Automatica Sinica*, vol. 45, no. 6, pp. 1015–1030, 2019.

[38] T. Dickerson, P. Gazzillo, M. Herlihy, and E. Koskinen, "Adding concurrency to smart contracts," in *Proc. ACM Symp. Princ. Distrib. Comput. (PODC)*. New York, NY, USA: Association for Computing Machinery, 2017, pp. 303–312, doi: 10.1145/3087801.3087835.

[39] K. Bhargavan et al., "Formal verification of smart contracts: Short paper," in *Proc. ACM Workshop Program. Lang. Anal. Secur.*, 2016, pp. 91–96.

[40] P. Tsankov, A. Dan, D. Drachsler-Cohen, A. Gervais, F. Bünzli, and M. Vechev, "Securify: Practical security analysis of smart contracts," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2018, pp. 67–82.

[41] S. Kalra, S. Goel, M. Dhawan, and S. Sharma, "ZEUS: Analyzing safety of smart contracts," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2018, pp. 1–12.

[42] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, pp. 1–32, Apr. 2014.

[43] J. Benet, "IPFS–content addressed, versioned, P2P file system," 2014, *arXiv:1407.3561*.

[44] F.-Y. Wang, R. Qin, Y. Chen, Y. Tian, X. Wang, and B. Hu, "Federated ecology: Steps toward confederated intelligence," *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 2, pp. 271–278, Mar. 2021.

[45] F.-Y. Wang, W. Zhang, Y. Tian, R. Qin, X. Wang, and B. Hu, "Federated data: Toward new generation of credible and trustable artificial intelligence," *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 3, pp. 538–545, Jun. 2021.

[46] D. N. Dillenberger et al., "Blockchain analytics and artificial intelligence," *IBM J. Res. Develop.*, vol. 63, nos. 2–3, pp. 1–5, 2019.

[47] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Blockchained on-device federated learning," 2018, *arXiv:1808.03949*.

[48] U. Majeed and C. S. Hong, "FLchain: Federated learning via MEC-enabled blockchain network," in *Proc. 20th Asia–Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Sep. 2019, pp. 1–4.

[49] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10700–10714, Dec. 2019.

[50] E. Ilie-Zudor, Z. Kemény, F. van Blommestein, L. Monostori, and A. van der Meulen, "A survey of applications and requirements of unique identification systems and RFID techniques," *Comput. Ind.*, vol. 62, no. 3, pp. 227–252, Apr. 2011.

[51] K. Domdouzis, B. Kumar, and C. Anumba, "Radio-frequency identification (RFID) applications: A brief introduction," *Adv. Eng. Inform.*, vol. 21, no. 4, pp. 350–355, 2007.

[52] J. Curtin, R. J. Kauffman, and F. J. Riggins, "Making the 'MOST' out of RFID technology: A research agenda for the study of the adoption, usage and impact of RFID," *Inf. Technol. Manage.*, vol. 8, no. 2, pp. 87–110, 2007, doi: 10.1007/s10799-007-0010-1.

[53] E. Elbasani, P. Siriporn, and J. S. Choi, "A survey on RFID in industry 4.0," in *Internet of Things for Industry 4.0*. Cham, Switzerland: Springer, 2020, pp. 1–16.

[54] E. E. Ozguven and K. Ozbay, "An RFID-based inventory management framework for emergency relief operations," *Transp. Res. C, Emerg. Technol.*, vol. 57, pp. 166–187, Aug. 2015.

[55] H. Cai, A. R. Andoh, X. Su, and S. Li, "A boundary condition based algorithm for locating construction site objects using RFID and GPS," *Adv. Eng. Informat.*, vol. 28, no. 4, pp. 455–468, Oct. 2014.

[56] R. Y. Zhong, Q. Dai, T. Qu, G. Hu, and G. Q. Huang, "RFID-enabled real-time manufacturing execution system for mass-customization production," *Robot. Comput.-Integr. Manuf.*, vol. 29, no. 2, pp. 283–292, 2013.

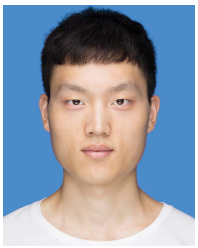
[57] R. Y. Zhong, Z. Li, L. Y. Pang, Y. Pan, T. Qu, and G. Q. Huang, "RFID-enabled real-time advanced planning and scheduling shell for production decision making," *Int. J. Comput. Integr. Manuf.*, vol. 26, no. 7, pp. 649–662, 2013.
[58] Z. X. Guo, E. W. T. Ngai, C. Yang, and X. Liang, "An RFID-based intelligent decision support system architecture for production monitoring and scheduling in a distributed manufacturing environment," *Int. J. Prod. Econ.*, vol. 159, pp. 16–28, Jan. 2015.
[59] T. Qu, H. D. Yang, G. Q. Huang, Y. F. Zhang, H. Luo, and W. Qin, "A case of implementing RFID-based real-time shop-floor material management for household electrical appliance manufacturers," *J. Intell. Manuf.*, vol. 23, no. 6, pp. 2343–2356, Dec. 2012.
[60] F. Zhang, P. Jiang, M. Zheng, and W. Cao, "A performance evaluation method for radio frequency identification–based tracking network of job-shop-type work-in-process material flows," *Proc. Inst. Mech. Eng., B, J. Eng. Manuf.*, vol. 227, no. 10, pp. 1541–1557, Oct. 2013.
[61] X. Jia, Q. Feng, T. Fan, and Q. Lei, "RFID technology and its applications in Internet of Things (IoT)," in *Proc. 2nd Int. Conf. Consum. Electron., Commun. Netw. (CECNet)*, Apr. 2012, pp. 1282–1285.
[62] M. Zhang, F. Sun, and X. Cheng, "Architecture of Internet of Things and its key technology integration based-on RFID," in *Proc. 5th Int. Symp. Comput. Intell. Design*, vol. 1, Oct. 2012, pp. 294–297.
[63] Investing. *Historical Data for Ethereum*. Accessed: Aug. 9, 2021. [Online]. Available: https://www.investing.com/crypto/ethereum/historical-data
[64] L. Ouyang, Y. Yuan, Y. Cao, and F.-Y. Wang, "A novel framework of collaborative early warning for COVID-19 based on blockchain and smart contracts," *Inf. Sci.*, vol. 570, pp. 124–143, Sep. 2021.

**Liwei Ouyang** received the B.S. degree in automation from Xi'an Jiaotong University, Xi'an, China, in 2018. She is currently pursuing the Ph.D. degree in social computing with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China.

Her current research interests include social computing, blockchain, and smart contracts.

**Fei-Yue Wang** (Fellow, IEEE) is a Professor at the Institute of Automation, Chinese Academy of Sciences, the Director at the State Key Laboratory for Management and Control of Systems, and also the Director at the China Economic and Social Security Research Center, University of Chinese Academy of Sciences. His research interest covers theories, methods, and applications for robotics, AI, intelligent control, parallel systems, social computing, parallel intelligence, and knowledge automation.

Dr. Wang is a fellow of INCOSE, IFAC, ASME, and AAAS. He received the National Prize in Natural Sciences of China and became an Outstanding Scientist of ACM for his work in intelligent control and social computing in 2007. He received the IEEE ITS Outstanding Application and Research Awards in 2009 and 2011, respectively. He received the IEEE SMC Society Norbert Wiener Award in 2014. Since 1997, he has been serving as the General or Program Chair for over 30 IEEE, INFORMS, IFAC, ACM, and ASME Conferences. He was the President of the IEEE ITS Society from 2005 to 2007, the Chinese Association for Science and Technology, USA, in 2005, the American Zhu Kezhen Education Foundation from 2007 to 2008, the Vice President of the ACM China Council from 2010 to 2011, the Vice President and the Secretary General of the Chinese Association of Automation from 2008 to 2018. He was the Founding Editor-in-Chief (EiC) of the *International Journal of Intelligent Control and Systems* from 1995 to 2000, the *IEEE ITS Magazine* from 2006 to 2007, the IEEE/CAA JOURNAL OF AUTOMATICA SINICA from 2014 to 2017, and the *China's Journal of Command and Control* from 2015 to 2020. He was the EiC of the *IEEE Intelligent Systems* from 2009 to 2012, the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS from 2009 to 2016, and has been the EiC of the IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS since 2017, and the Founding EiC of *China's Journal of Intelligent Science and Technology* since 2019. He is currently the President of CAA's Supervision Council, the IEEE Council on RFID, and Vice President of IEEE Systems, Man, and Cybernetics Society.



**Yonglin Tian** received the bachelor's degree from the University of Science and Technology of China, Hefei, China, in 2017, where he is currently pursuing the Ph.D. degree with the Department of Automation.

His research interests include computer vision and intelligent transportation systems.

**Xiaofeng Jia** is currently a Senior Engineer and the Director of the Data Management Department, Beijing Big Data Center. He has been engaged in the research and practice of big data, blockchain, and privacy computing.

**Hongwei Qi** received the Ph.D. degree from the Institute of automation, Chinese Academy of Sciences, Beijing, China, in 2004.

He is the CEO of the Datatang (Beijing) Technology Company, Ltd., Beijing. He is also a Senior Engineer. He is mainly engaged in artificial intelligence data acquisition, processing, and federal data.

**Ge Wang** is currently pursuing the Ph.D. degree with the Institute of Systems Engineering, Macau University of Science and Technology, Macau, China.

His research interests include federated governance, Smart contracts, DAO, and parallel intelligence.