

# Towards Verifiable Federated Learning

Yanci Zhang and Han Yu

School of Computer Science and Engineering, Nanyang Technological University, Singapore  
{yanci001, han.yu}@ntu.edu.sg

## Abstract

Federated learning (FL) is an emerging paradigm of collaborative machine learning that preserves user privacy while building powerful models. Nevertheless, due to the nature of open participation by self-interested entities, it needs to guard against potential misbehaviours by legitimate FL participants. FL verification techniques are promising solutions for this problem. They have been shown to effectively enhance the reliability of FL networks and build trust among participants. Verifiable FL has become an emerging topic of research that has attracted significant interest from the academia and the industry alike. Currently, there is no comprehensive survey on the field of verifiable federated learning, which is interdisciplinary in nature and can be challenging for researchers to enter into. In this paper, we bridge this gap by reviewing works focusing on verifiable FL. We propose a novel taxonomy for verifiable FL covering both centralised and decentralised settings, summarise the commonly adopted performance evaluation approaches, and discuss promising directions towards a versatile verifiable FL framework.

## 1 Introduction

Machine learning (ML) has become a useful tool that has impacted wide-ranging applications in our daily lives. Various industries are increasingly relying on ML techniques to provide quality services based on their rich collections of data. However, traditional ML techniques require data to be located in a central facility in order to train models. In an increasingly privacy-aware world [GDPR, 2018], the privacy issues related to such a practice hinder the development of ML-empowered applications, especially in sensitive fields such as healthcare and finance.

To enable ML techniques to continue their development trajectory under data privacy protection requirements, the federated learning paradigm [Konečný *et al.*, 2016] has been proposed, enabling data owners (a.k.a. FL clients or FL trainers) to train machine learning models collaboratively without compromising privacy. The classical centralised FL framework consists of a central FL server and multiple FL clients.

The FL server first initialises the model parameters and securely passes them to the FL clients. Each client independently trains its local model using its local computing resources and local data. After this step, the client sends the model parameters back to the FL server. The server aggregates the received local models to obtain an updated global FL model, and distributes it among the clients for further training. This iterates until the model meets the convergence conditions. Apart from centralised FL, the decentralised FL architecture has also emerged. In decentralised FL, the server's role is taken up by a blockchain. Each data owner (a.k.a. trainer) performs local model training. These local model updates are shared with selected data owners (a.k.a. workers) who perform model aggregation and generate a new block containing the aggregated model on the blockchain. Information stored on the blockchain is immutable to modification and accessible by all.

Federated learning is a useful approach for enabling collaborative model training in a privacy-preserving manner. However, this open collaboration among potentially self-interested parties who may not know each other well has brought new challenges for FL. For example, the FL server which is responsible for selecting data owners to participate in FL, ascertaining their contributions to the model and distributing incentive payouts [Zhan *et al.*, 2021] might deviate from the agreed protocols in pursuit of its own interest. Similarly, FL clients might also deviate from the FL protocol to free-ride on others [Lyu *et al.*, 2020] or to slow down/impede the model training process [Park *et al.*, 2021]. In addition, FL clients who forge their identities, lie about the size or quality of their datasets and local computing resources can mislead the decisions and negatively impact the FL process. Therefore, it is crucial to verify the information provided and the behaviours undertaken by the FL participants in order to build a healthy FL ecosystem.

The research of introducing verification into FL has attracted significant attention. A verifiable FL system not only enhances the system security, but also helps build trust among legitimate participants. In centralised FL, to verify the correctness of the aggregated models produced by the FL server, the server needs to generate a proof, a signature or a ciphertext [Xu *et al.*, 2019; Guo *et al.*, 2020; Zhang *et al.*, 2020b; Fu *et al.*, 2020; Mou *et al.*, 2021; Han *et al.*, 2021; Jiang *et al.*, 2021; Madi *et al.*, 2021;

Hahn *et al.*, 2021] to prove that the aggregation result is not forged. FL clients need verification from the server in order to participate. Verification for clients’ local information, including but not limited to identity and local data, is the foundation of a secure FL system [Zhang *et al.*, 2020a; Hahn *et al.*, 2021; Mou *et al.*, 2021; Lo *et al.*, 2021; Li *et al.*, 2021; Zhang *et al.*, 2021a]. FL client training performance is another part of client verification [Kang *et al.*, 2019; Zhang *et al.*, 2020c; Kang *et al.*, 2020; Song *et al.*, 2021] as forged local model updates can lead to poor performance in the final model. Similarly, in decentralised FL, the workers need to verify the uploaded model parameters before performing model aggregation and update [Kim *et al.*, 2019; Bao *et al.*, 2019; Zhao *et al.*, 2020; Peng *et al.*, 2021].

The field of verifiable federated learning is interdisciplinary as it requires expertise from machine learning, cryptography and game theory, etc. This makes it challenging for researchers new to the field to grasp the latest development. Currently, there is no survey paper on this important and rapidly developing field. To bridge this gap, we provide a comprehensive survey of research works towards a verifiable FL framework in this paper. We propose a unique taxonomy of verifiable FL that organises existing works according to the FL system architecture, the stakeholders involved and the subjects of verification to provide readers with a multi-perspective view into this field. We analyse the limitations of current approaches, summarise the commonly adopted performance evaluation approaches, and offer promising future directions leading towards verifiable FL frameworks.

## 2 Preliminaries

### 2.1 Defining Verifiable Federated Learning

Before delving into our survey, we first need to define what counts as a verifiable FL approach. In [Kairouz *et al.*, 2021], verifiability is defined as the ability of an FL client or the FL server to prove to others who engage in an FL protocol that it has executed the desired behaviour faithfully, without revealing the potentially private data upon which they were acting. However, this definition is strictly restricted in the centralised FL setting. Therefore, in this paper, we extend the definition of verifiability to both centralised setting and decentralised setting as follows:

**Definition (Verifiable FL).** *Verifiability is the ability of one party to prove to other parties in an FL protocol that it has correctly performed the intended task without deviation.*

### 2.2 Stakeholder Analysis

In an FL ecosystem, there can be many model users looking for data owners to collaboratively build FL applications. To ensure trustworthy and sustainable interactions in such an open dynamic environment, there are two types of direct stakeholders for verifiable FL: 1) FL servers (only applicable in centralised FL settings), and 2) data owners. They are directly involved in the FL training processes. Indirect stakeholders include third-party auditors, researchers and developers, government agencies, policymakers, civil societies, and insurance companies, etc. Each stakeholder has unique needs for functional features from a verifiable FL system [Brundage

*et al.*, 2020]. The diverse stakeholders’ needs require different verifiable FL techniques.

FL servers need to verify that FL clients possess the resources they claim and have followed the training protocol faithfully. Data owners need to verify that the FL server is indeed protecting their data privacy. Third-party auditors need to leverage the auditable trails recorded during historical verification processes to ascertain the correctness of the verification operations. Researchers and developers need to verify whether the model performance (e.g., accuracy and running time) meets the expectation. Government agencies, policymakers and civil societies need to verify whether the FL frameworks comply with regulations. In mission-critical applications (e.g., healthcare, finance), insurance agencies need to verify if there exists any bias in the FL models involved. In order to support diverse stakeholders’ needs, different verifiable FL techniques are required.

## 3 The Proposed Verifiable FL Taxonomy

Based on the extended definition of FL verifiability given in Section 2.1, we propose a taxonomy of verifiable FL (as shown in Figure 1) according to the direct stakeholders who require the given form of verification. The taxonomy first separates verifiable FL into centralised and decentralised settings. Since any legitimate FL participant can misbehave, either through not performing its tasks faithfully or maliciously exposing private information, we further classify the verifiable FL based on the target of verification. This hierarchical taxonomy provides a clear overview of the current verifiable FL research landscape.

### 3.1 Verifiable Centralised FL Systems

In centralised FL, the server and clients are the two main types of participants. Therefore, the aggregated FL model distributed by the server and the local model updates submitted by the clients are important targets of verification.

#### Verifying the FL Server

The role played by the FL server during training is to aggregate local model updates submitted by FL clients and then distribute the aggregated model back to clients for further training. The reliability of the server is vital to the FL protocol. The server cannot be assumed to be always trustworthy in all scenarios [Kairouz *et al.*, 2021]. An untruthful FL

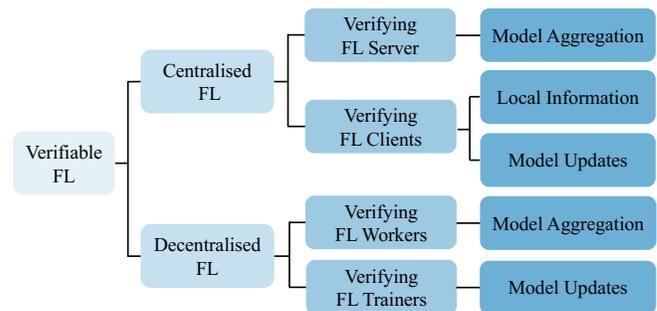


Figure 1: The proposed taxonomy of verifiable FL approaches.

server can compromise privacy and benefits for all clients involved. Therefore, it is crucial for the clients to verify the integrity of the received aggregated FL model.

When verifying the FL server, the server is the prover and each FL client is a verifier. In non-FL settings, SafetyNets [Ghodsai *et al.*, 2017] uses an interactive proof protocol to enable cloud service users to verify the correctness of the tasks that the cloud server performs. In each round of interaction, the user sends the cloud server a randomly selected challenge. The server proves its correctness by a short proof. Although SafetyNets is only applicable in non-FL settings and for deep neural network (DNN)-based models which can be represented as arithmetic circuits, this proof generation technique lays the foundation for FL server verification.

Existing FL server verification research commonly makes the following assumptions:

1. *Semi-honest FL Clients*: FL clients are assumed to truthfully upload their local model parameters, but are curious about other clients' private information. They might collude with the server to gain access to such information.
2. *Malicious FL Server*: The FL server might attempt to steal private information from clients through their local model updates. It might forge the aggregated global FL model to be distributed among the clients.

Under these assumptions, a number of approaches have been proposed to verify the correctness of server aggregated FL models. In general, the server is required to return the aggregated model in a certain encrypted form or with computed proof for clients to verify. The server can obtain a valid proof only by correctly executing the aggregation steps. Since encryption is often involved for verifying the FL server, initialisation steps (e.g., key generation) are required before model training. We divide existing works based on the entity conducting the initialisation operations.

**Initialisation by a Trusted Third Party.** In this setting, a trusted third party takes charge of initial key generation and assignment before training begins. This entity is assumed to be trustworthy and does not collude with any FL participant.

VerifyNet [Xu *et al.*, 2019] is one of the first models to verify the correctness of the server returned results. Under VerifyNet, the server returns the aggregated FL model together with its proof. All clients can then verify the proof by utilising the homomorphic hash function integrated with the pseudorandom function to determine whether the aggregation is trustful or malicious. In [Mou *et al.*, 2021] and [Han *et al.*, 2021], the authors also adopt homomorphic hash functions and pseudorandom functions to verify server aggregated results via server-generated proof. The difference between these three works mainly lies in the underlying model gradient encryption and privacy protection techniques.

The homomorphic hash [Fiore *et al.*, 2014], however, is computationally costly as it heavily relies on the bilinear pairing [Brezing and Weng, 2005] which conducts the pairing operations on each entry in a vector, making the time cost linearly related to the dimension of the model gradient vectors. Therefore the communication cost is huge when the dimen-

sion of the model gradient vector is high. In view of this problem, VERIFL [Guo *et al.*, 2020] reduces the communication overhead by requiring each client to commit only the hash of its gradient vector rather than the vector itself. In [Zhang *et al.*, 2020b], the bilinear aggregate signature technique is introduced to verify the correctness of aggregated parameters. The authors leverage the Chinese Remainder Theorem to reduce communication costs. Similarly, VFL [Fu *et al.*, 2020] uses a pseudorandom generator and Lagrange interpolation together with the Chinese Remainder Theorem to encrypt the client's gradient. The server performs aggregation directly using the received ciphertext. Thus, clients can verify the correctness of the aggregated model by unpacking the received aggregated ciphertext from the server. In [Jiang *et al.*, 2021], the proof returned by the server is generated by cryptographic accumulators and stored in a public blockchain for verification. Clients can apply a variant of ElGamal encryption to verify the correctness of the aggregated models. Initialisation by a trusted third party is a strong assumption as in real-world applications as such an entity might not always be available.

**Initialisation by the Server or the Client(s).** To suit real-world application scenarios, researchers are incorporating the initialisation step into FL frameworks without relying on the availability of a trusted third party. This means the initialisation task is either performed by the FL server or FL client(s).

In [Madi *et al.*, 2021], the initial keys vital for homomorphic encryption are generated by one of the clients. This client is either randomly selected by the server, or elected by clients following a given voting protocol. However, this approach only works under the assumption that clients are honest and the server does not collude with any client. Otherwise, a malicious server can select a colluding client to compromise with the key generation process, which is essential for the subsequent training and verification steps. For the verification algorithm, the authors leverage the Paillier cryptosystem [Struck *et al.*, 2017], a linearly homomorphic authenticated encryption scheme, to check the correctness of aggregated model returned by the server. However, the computational overhead of Paillier encryption becomes very high when the number of participants involved is large.

Alternatively, VERSA [Hahn *et al.*, 2021] uses the public key infrastructure (PKI) [Maurer, 1996] to achieve safe initialisation without any trusted entity. PKI binds public keys with the respective participants when they advertise their keys to join the FL protocol. It reduces the computational cost by leveraging a lightweight pseudorandom generator to verify the aggregated FL model from the server.

### Verifying the FL Clients

The FL clients can also misbehave and deviate from the FL protocol. There are a few types of client misbehaviours. Firstly, clients can lie about the quantity and quality of their local data, which can mislead the FL model aggregation algorithms at the server. Besides, a client can forge its identity. These two misbehaviours are both about a client's local information. Secondly, clients who aim to free-ride on others and save their local computational resources can reuse local model updates from past iterations or even randomly generate model parameters to avoid actually performing local model

training. Malicious clients can even poison the local model parameters in an attempt to compromise the final FL model. Therefore, it is crucial to verify clients' local information and their model updates. In this case, FL clients are provers and the FL server is the verifier. Works in this domain commonly make the following assumptions:

1. *Semi-Honest FL Server*: The FL server is assumed to follow the FL protocols faithfully, but is curious about the clients' private information.
2. *Misbehaving FL Clients*: It is assumed that there exist FL clients who misbehave. The specific types of client misbehaviours vary among different works.

**Verifying Client Local Information.** A client's local information consists of its identity and local data. The verification of client identity mainly relies on signatures or proofs generated by the clients. VERSA [Hahn *et al.*, 2021] leverages digital signatures to verify the identities of clients, making it applicable for scenarios in which the clients are malicious rather than just being honest-but-curious. Alternatively, the server authenticates client identity utilising a Fiat-Shamir zero-knowledge proof [Shoup, 2000] before the start of the training process [Mou *et al.*, 2021].

Verification of a client's local data mainly relies on a safe data structure to store the data version information without exposing private information. In [Zhang *et al.*, 2020a], a blockchain is adopted to verify the integrity of client data. Each client periodically creates a Merkle tree in which each leaf node represents a client data record, and stores the tree root on the blockchain. When a dispute happens, the FL server can verify client data integrity by comparing it with the respective Merkle tree root stored on the blockchain. In [Lo *et al.*, 2021], the hashed global model parameters and local data version information (including timestamps and data size) are recorded in a blockchain, enabling verification and auditing of client identity and local model performance. These two works focus on making useful information available for verification purposes, rather than how to verify the information.

The quality of a client's local data can be verified by influence function during the training process. The influence-based hierarchical framework [Li *et al.*, 2021] enables the FL server to first identify the most negatively influential FL clients and find the most negatively influential data samples. By applying a participant selection strategy to remove these negative samples, the framework can fix bugs as well as accelerate FL model convergence. Similarly, in [Zhang *et al.*, 2021a], the data quality is verified using the loss on the validation set. If the difference between the loss of the global model formed with and without the suspicious local model exceeds a threshold, this model is discarded. Note that when verifying local data quality, clients are assumed to correctly follow the FL protocol and upload their models truthfully.

**Verifying Client Model Updates.** During FL training, clients are supposed to train local models with their local data and upload the local model parameters to the server for aggregation. Verification during this process is essential for ensuring that clients follow this agreed protocol.

Similar to verifying server aggregated models, the proof

generated by the client is also used to verify the correctness of client uploaded model updates. TrustFL [Zhang *et al.*, 2020c] leverages trusted execution environments to verify if local FL model training has been executed with high confidence. Each client performs local model training in an isolated environment and outputs the model updates with its signature. The server verifies the signatures of randomly selected clients in each round of training to check their correctness. To prevent clients from contributing only when requested to be verified, TrustFL adopts a "commit-and-prove" design, requiring the clients to send a commitment message indicating the training rounds that they have finished before they know about whether they will be verified. The verification can later be conducted on a random claimed round. To mitigate misbehaviours involving reusing previously obtained local models to update the server, TrustFL devises a "dynamic-yet-deterministic" data selection method by requiring data owners to ensure that training data used for local model training are different at each round. As a result, model updates from the same data owner in different iterations shall be different.

Reputation schemes have been applied to help verify FL clients' contributions to the aggregated model. There are multiple ways to calculate a client's reputation, including multi-weight subjective logic [Kang *et al.*, 2020], historical contribution evaluation based on Shapley values [Liu *et al.*, 2022b; Liu *et al.*, 2022a], gradient vector similarity [Xu and Lyu, 2020], interaction behaviours [Zou *et al.*, 2021], test performance [Wang and Kantarci, 2020; Wang and Kantarci, 2021], beta distribution [Song *et al.*, 2021] and subjective evaluation by FL task initiators [Zhang *et al.*, 2021a]. Besides evaluating client training contributions, reputation schemes have also been used to detect malicious users [Song *et al.*, 2021]. Reputable clients have a higher probability of joining FL model aggregation and the next training iteration. Reputation values can be stored on a blockchain for efficient and transparent management [Kang *et al.*, 2020; ur Rehman *et al.*, 2020; Zhang *et al.*, 2021a; Zhang *et al.*, 2021b].

Contract theory has also been applied to verify FL clients' contributions [Kang *et al.*, 2019; Ye *et al.*, 2020; Lim *et al.*, 2020; Ding *et al.*, 2020; Tian *et al.*, 2021]. The FL server designs specific contracts for different types of data owners characterised by different levels of data quality and the corresponding payoffs. Each data owner must select and sign one of these contracts in order to participate in FL. In some works [Kang *et al.*, 2019], if the data owner cannot fulfil the expected contractual obligation, the FL server can adjust its reputation accordingly and withhold payment.

### 3.2 Verifiable Decentralised FL Systems

Decentralised FL systems do not require an FL server. The tasks of model aggregation and distribution are mainly performed by participating data owners. Blockchains are often leveraged by such FL systems due to their desirable qualities, including:

1. *Security*: Once a block is accepted and added to the blockchain, no one can tamper with the records without being detected. This property makes blockchain secure storage for ensuring information integrity.

2. *Automation*: The smart contracts in blockchains ensure that once a pre-defined condition is met, specific operations can be triggered. This automatic contract checking scheme enhances the efficiency of blockchains.
3. *Transparency*: Information stored on the blockchain (e.g., FL model updates) is accessible by all, enabling them to perform verification and auditing conveniently.

In blockchain-based decentralised FL systems, the trainers (i.e., data owners) refer to the smart contract to train their local models and upload their local parameter updates to the blockchain network. Selected participants or organisations can register as workers (a.k.a. miners) in the blockchain, who take charge of verifying the information uploaded by trainers. Each worker mines according to the blockchain consensus protocol until it can generate a new block or a block is received from another worker. When a worker generates a new block, the other workers verify the contents of the block (e.g., random numbers, the status of smart contract changes, aggregated models). If a block is agreed upon by the majority of the workers, the block is added to the blockchain and accepted by the network. After the new block containing the updated FL model is added to the blockchain, all participants can update their models according to the information stored and begin the next round of FL model training.

### Verifying the FL Trainers

In decentralised FL settings, trainers upload model updates to the blockchain. Verification of FL model updates occurs before workers can aggregate them. In this scenario, trainers are the provers and each worker is a verifier. The common assumption made by existing works is that:

1. *Dishonest Trainers*: A trainer might upload forged model parameters to the blockchain.

In BlockFL [Kim *et al.*, 2019], a blockchain is designed for exchanging local model updates, verifying the submitted information, and allocating rewards. Specifically, workers in BlockFL are either randomly selected devices or separate nodes such as network edges. Each device engaging in the FL protocol is assigned to a random worker. Workers verify the received model updates before aggregation. They compare the sample size of a device with the time taken for it to produce the model updates to identify misbehaving devices which attempt to avoid actually training the local model.

Proofs are also used to verify trainers' behaviours. In [Bao *et al.*, 2019; Zhao *et al.*, 2020], model gradients generated by the trainers are masked and uploaded with their correctness proofs. The workers, who are registered trainers, check the masked gradients based on the proofs to determine which of them shall be selected for model aggregation. The selected gradients are then aggregated by the workers following the blockchain consensus protocol. A worker is selected as the leader based on verifiable random functions [Zhao *et al.*, 2020] or its reliability ranking [Bao *et al.*, 2019] to document its aggregation results into the blockchain. The reliability reflects each trainer's performance. It jointly considers the proof verification results, the model buyer's evaluation and whether it has successfully identified misbehaviours (if

the trainer has been selected as a worker). A trainer's reliability value will also impact its chance to participate in future FL model training. Besides, the systems support third-party auditors to investigate misbehaviours based on the training records in the blockchain.

### Verifying the FL Workers

Workers in a blockchain are responsible for FL model aggregation. The verification of FL model aggregation takes place after a worker has successfully aggregated the received local model gradients. The system needs to verify whether the worker has performed the aggregation correctly. In this scenario, the worker who adds the new block to the blockchain is the prover and other workers are the verifiers. The following common assumption is made by existing works:

1. *Unreliable Workers*: It is assumed that there exist compromised workers who might not perform FL model aggregation following the correct protocol, while the majority of the workers remain trustworthy.

In VFChain [Peng *et al.*, 2021], each trainer generates a signature and upload it to the blockchain together with its model updates. The workers compete with each other to create a valid block containing the aggregated FL model and signatures from the trainers. A trusted committee is formed by randomly selected workers to verify the aggregated model according to verifying contract. The committee is continuously updated over training iterations. Historical aggregated models, together with verification signatures, are stored in the blockchain which is immutable to modification and accessible to all users. To improve the auditing efficiency, an authenticated data structure has been proposed based on a dual skip chain to store the training trails. Nevertheless, VFChain can only work when the trainers are semi-honest and do not collude with each other.

## 4 Verifiable FL Evaluation Methods

To evaluate the effectiveness of a verifiable FL approach, a combination of theoretical analysis and experimental evaluation is commonly adopted.

### 4.1 Theoretical Analysis of Verifiability

Theoretical analysis of verifiability is most relevant for FL verification approaches designed to detect misbehaviours by FL participants [Xu *et al.*, 2019; Kang *et al.*, 2019; Bao *et al.*, 2019; Guo *et al.*, 2020; Zhang *et al.*, 2020b; Fu *et al.*, 2020; Zhang *et al.*, 2020c; Jiang *et al.*, 2021; Madi *et al.*, 2021; Hahn *et al.*, 2021; Song *et al.*, 2021; Mou *et al.*, 2021; Han *et al.*, 2021]. These works generally require the prover to generate a proof, a signature or a ciphertext for the verifier to check. The goal of theoretical analysis is to show that, under the given verifiable FL scheme, once the generated proof is verified to be correct by the verifier, it can be concluded that the activity being verified is correct.

The specific approach to proving the verifiability of a given scheme depends on the proof generation technique used. Taking the works based on homomorphic hash functions and pseudorandom functions to verify the correctness of aggregation from the server for example [Xu *et al.*, 2019;

Mou *et al.*, 2021; Han *et al.*, 2021], these works perform analysis based on the 1-BDHI assumption [Fiore *et al.*, 2014] and the DDH assumption [Boneh and Franklin, 2001] to prove that the proposed scheme can conclusively verify the prover’s activities. Verifiability analysis is a useful approach to show the theoretical foundation and expected performance bounds of given verifiable FL approaches.

## 4.2 Experimental Evaluation Metrics

Experimental evaluations are useful to test given verifiable FL approaches in environments with simulated or actual misbehaviours in order to study their effectiveness in complex settings. The following evaluation metrics are commonly used to quantify the effectiveness of these approaches.

**Model Performance.** Model performance is an important evaluation metric for verifiable FL systems as the goal of FL participants is to obtain an improved model. In FL settings, model performance is closely related to the number of participating data owners and the size of the local gradient updates from them. Experiments often fix the number of participating data owners while varying the size of their local gradient updates (and vice versa) to study the FL model performance. Test accuracy is commonly used to evaluate model performance. Precision, recall and F1-score have also been used to evaluate verifiable FL approaches designed for classification models [Zhang *et al.*, 2020a].

**Verification Overhead.** Verification incurs computation and communication overhead, which is related to the number of data owners and the size of the local gradient updates. Experiments designed to measure computation and communication overhead are often conducted in settings where these two factors are varied over a range of values. The computation cost is quantified by the time taken for verification. It can be separated into the time taken for proof generation and validation [Han *et al.*, 2021], or in finer granularities (e.g., time taken for key generation, encryption, verification, decryption and evaluation [Madi *et al.*, 2021]). One-time computation costs (e.g., time for secure setup) are often ignored. Communication overhead is measured by the total amount of data transmitted during the verification process. This includes the amount of transmission required to successfully upload the verification information onto the blockchain in blockchain-based verifiable FL [Jiang *et al.*, 2021; Lo *et al.*, 2021].

**Resistance to Attacks.** To evaluate the robustness of verifiable FL, resistance to attacks has been used as an indicator. These attacks are conducted in simulated environments. In works designed to verify the training performance of FL clients [Kang *et al.*, 2020; Kang *et al.*, 2019], the quality of local training data, the number of attackers and the intensity of attacks are varied to simulate different conditions in which the verifiable FL approach needs to operate. The performance of the FL model produced in such conditions is used to measure how resistant the approach is in the face of attacks.

## 5 Promising Future Research Directions

Through our survey, it can be observed that research in verifiable FL today mostly focuses on making key information

available to facilitate verification, and performing verification on specific activities and roles in various types of FL systems. To advocate verification to become an integral part of future FL systems to allow the community of participants to self-police and self-organise, we envision the following future research directions.

**Relaxation of Simplifying Assumption.** Current works commonly assume at least one party in FL is semi-honest. In centralised FL, if clients want to verify the correctness of the aggregated model returned by the FL server, all clients are assumed to be semi-honest. Similarly, the FL server also needs to be semi-honest when it verifies the clients. This makes existing approaches vulnerable in situations in which the server and clients are malicious or collude. This limiting assumption needs to be relaxed to enable future verifiable FL approaches to handle more challenging real-world conditions.

**Verification without Trial FL Model Training.** In FL settings, participants often need to self-declare certain information in order to be selected to join a federation. Such self-declared information includes the quality and quantity of local data, and the computational resource of a data owner. Due to privacy concerns, the verifier cannot directly inspect such information to check if it is truthfully declared. Thus, existing verifiable FL approaches generally require candidate data owners to participate in FL model training first, before they can verify their claims on local resource commitment by observing their performance during training. To reduce the waste on computation and communication resources as a result of this trial-and-error approach, research on privacy-preserving verification of self-declared information without requiring trial participation in FL model training is desired.

**Automating Auditing Report Generation.** Decentralised verifiable FL approaches often keep traceable trails of verification for auditing purposes. These records usually contain model updates, aggregated models and verification proofs, etc. However, raw records are not convenient for third-party auditors to make sense of. This source of information can be leveraged in combination with explainable AI (XAI) research [Tjoa and Guan, 2021] to automate the generation of auditing reports to enhance the transparency of the FL verification processes. By dynamically incorporating XAI techniques [Kim *et al.*, 2021] such as explicit explanation, alternative explanation, knowledge of the explainees and interaction designs with FL verification, auditing reports for diverse FL stakeholders with different needs for insight can be generated, thereby enabling wider participation in upholding the integrity of verifiable FL.

**Verification-based Community Sanctioning.** Identifying individual instances of misbehaviour by FL participants is important. Nevertheless, in order to build long-term insight into participants’ behaviour patterns, it is necessary to investigate how to leverage the verification results to form the basis of community sanctioning. Preliminary works such as [Li *et al.*, 2021] are starting to emerge in this area which attempts to build trust among participants based on verification results. Research in verification-based social norm formation [Shulman *et al.*, 2017] and interaction strategies can enable FL systems to self-police their members and deter misbehaviours.

## Acknowledgements

This research is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-RP-2020-019); Alibaba Group through Alibaba Innovative Research (AIR) Program and Alibaba-NTU Singapore Joint Research Institute (JRI); the Nanyang Assistant Professorship (NAP); the RIE 2020 Advanced Manufacturing and Engineering (AME) Programmatic Fund (No. A20G8b0102), Singapore; and Future Communications Research & Development Programme (FCP-NTU-RG-2021-014). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

## References

- [Bao *et al.*, 2019] Xianglin Bao, Cheng Su, et al. FLChain: A blockchain for auditable federated learning with trust and incentive. In *BIGCOM*, pages 151–159, 2019.
- [Boneh and Franklin, 2001] Dan Boneh and Matt Franklin. Identity-based encryption from the weil pairing. In *CRYPTO*, pages 213–229, 2001.
- [Brezing and Weng, 2005] Friederike Brezing and Annegret Weng. Elliptic curves suitable for pairing based cryptography. *Designs, Codes and Cryptography*, 37(1):133–141, 2005.
- [Brundage *et al.*, 2020] Miles Brundage, Shahar Avin, et al. Toward trustworthy ai development: mechanisms for supporting verifiable claims. *arXiv preprint arXiv:2004.07213*, 2020.
- [Ding *et al.*, 2020] Ningning Ding, Zhixuan Fang, and Jianwei Huang. Optimal contract design for efficient federated learning with multi-dimensional private information. *IEEE Journal on Selected Areas in Communications*, 39(1):186–200, 2020.
- [Fiore *et al.*, 2014] Dario Fiore, Rosario Gennaro, and Valerio Pastro. Efficiently verifiable computation on encrypted data. In *CCS*, pages 844–855, 2014.
- [Fu *et al.*, 2020] Anmin Fu, Xianglong Zhang, et al. Vfl: A verifiable federated learning with privacy-preserving for big data in industrial iot. *IEEE Transactions on Industrial Informatics*, 2020.
- [GDPR, 2018] GDPR. General data protection regulation. <https://gdpr-info.eu/>, 2018. Accessed: 2021-12-08.
- [Ghodsizadeh *et al.*, 2017] Zahra Ghodsizadeh, Tianyu Gu, and Siddharth Garg. Safetynets: Verifiable execution of deep neural networks on an untrusted cloud. *arXiv preprint arXiv:1706.10268*, 2017.
- [Guo *et al.*, 2020] Xiaojie Guo, Zheli Liu, et al. VeriFL: Communication-efficient and fast verifiable aggregation for federated learning. *IEEE Transactions on Information Forensics and Security*, 16:1736–1751, 2020.
- [Hahn *et al.*, 2021] Changhee Hahn, Hodong Kim, et al. Versa: Verifiable secure aggregation for cross-device federated learning. *IEEE Transactions on Dependable and Secure Computing*, 2021.
- [Han *et al.*, 2021] Gang Han, Tiantian Zhang, et al. Verifiable and privacy preserving federated learning without fully trusted centers. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–11, 2021.
- [Jiang *et al.*, 2021] Changsong Jiang, Chunxiang Xu, and Yuan Zhang. PFLM: Privacy-preserving federated learning with membership proof. *Information Sciences*, 576:288–311, 2021.
- [Kairouz *et al.*, 2021] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G.L. D’Oliveira, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badri Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrede Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1-2):1–210, 2021.
- [Kang *et al.*, 2019] Jiawen Kang, Zehui Xiong, et al. Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory. *IEEE Internet of Things Journal*, 6(6):10700–10714, 2019.
- [Kang *et al.*, 2020] Jiawen Kang, Zehui Xiong, et al. Reliable federated learning for mobile networks. *IEEE Wireless Communications*, 27(2):72–80, 2020.
- [Kim *et al.*, 2019] Hyesung Kim, Jihong Park, et al. Blockchained on-device federated learning. *IEEE Communications Letters*, 24(6):1279–1283, 2019.
- [Kim *et al.*, 2021] Mi-Young Kim, Shahin Atakishiyev, et al. A multi-component framework for the analysis and design of explainable artificial intelligence. *Machine Learning & Knowledge Extraction*, 3:900–921, 2021.
- [Konečný *et al.*, 2016] Jakub Konečný, H Brendan McMahan, et al. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.
- [Li *et al.*, 2021] Anran Li, Lan Zhang, et al. Efficient federated-learning model debugging. In *ICDE*, pages 372–383, 2021.
- [Lim *et al.*, 2020] Wei Yang Bryan Lim, Sahil Garg, et al. Dynamic contract design for federated learning in smart healthcare applications. *IEEE Internet of Things Journal*, 8(23):16853–16862, 2020.

- [Liu *et al.*, 2022a] Zelei Liu, Yuanyuan Chen, et al. Contribution-aware federated learning for smart health-care. In *IAAI*, 2022.
- [Liu *et al.*, 2022b] Zelei Liu, Yuanyuan Chen, et al. GTG-Shapley: Efficient and accurate participant contribution evaluation in federated learning. *ACM Transactions on Intelligent Systems and Technology*, 2022.
- [Lo *et al.*, 2021] Sin Kit Lo, Yue Liu, et al. Blockchain-based trustworthy federated learning architecture. *arXiv preprint arXiv:2108.06912*, 2021.
- [Lyu *et al.*, 2020] Lingjuan Lyu, Jiangshan Yu, et al. Towards fair and privacy-preserving federated deep models. *IEEE Transactions on Parallel and Distributed Systems*, 31(11):2524–2541, 2020.
- [Madi *et al.*, 2021] Abbass Madi, Oana Stan, et al. A secure federated learning framework using homomorphic encryption and verifiable computing. In *RDAAPS*, pages 1–8, 2021.
- [Maurer, 1996] Ueli Maurer. Modelling a public-key infrastructure. In *ESORICS*, pages 325–350, 1996.
- [Mou *et al.*, 2021] Wenhao Mou, Chunlei Fu, et al. A verifiable federated learning scheme based on secure multi-party computation. In *WASA*, pages 198–209, 2021.
- [Park *et al.*, 2021] Jung Wuk Park, Dong-Jun Han, et al. Sageflow: Robust federated learning against both stragglers and adversaries. In *NeurIPS*, 2021.
- [Peng *et al.*, 2021] Zhe Peng, Jianliang Xu, et al. VFChain: Enabling verifiable and auditable federated learning via blockchain systems. *IEEE Transactions on Network Science and Engineering*, 9(1):173–186, 2021.
- [Shoup, 2000] Victor Shoup. Practical threshold signatures. In *EUROCRYPT*, pages 207–220, 2000.
- [Shulman *et al.*, 2017] Hillary C. Shulman, Nancy Rhodes, et al. The state of the field of social norms research. *International Journal of Communication*, 11:1192–1213, 2017.
- [Song *et al.*, 2021] Zhendong Song, Hongguang Sun, et al. Reputation-based federated learning for secure wireless networks. *IEEE Internet of Things Journal*, 9(2):1212–1226, 2021.
- [Struck *et al.*, 2017] Patrick Struck, Lucas Schabhüser, et al. Linearly homomorphic authenticated encryption with provable correctness and public verifiability. In *C2SI*, pages 142–160, 2017.
- [Tian *et al.*, 2021] Mengmeng Tian, Yuxin Chen, et al. A contract theory based incentive mechanism for federated learning. *arXiv preprint arXiv:2108.05568*, 2021.
- [Tjoa and Guan, 2021] Erico Tjoa and Cuntai Guan. A survey on explainable artificial intelligence (XAI): Toward medical XAI. *IEEE Transactions on Neural Networks and Learning Systems*, 32(11):4793–4813, 2021.
- [ur Rehman *et al.*, 2020] Muhammad Habib ur Rehman, Khaled Salah, et al. Towards blockchain-based reputation-aware federated learning. In *INFOCOM-WKSHPS*, pages 183–188, 2020.
- [Wang and Kantarci, 2020] Yuwei Wang and Burak Kantarci. A novel reputation-aware client selection scheme for federated learning within mobile environments. In *CAMAD*, pages 1–6, 2020.
- [Wang and Kantarci, 2021] Yuwei Wang and Burak Kantarci. Reputation-enabled federated learning model aggregation in mobile platforms. In *ICC*, pages 1–6, 2021.
- [Xu and Lyu, 2020] Xinyi Xu and Lingjuan Lyu. A reputation mechanism is all you need: Collaborative fairness and adversarial robustness in federated learning. *arXiv preprint arXiv:2011.10464*, 2020.
- [Xu *et al.*, 2019] Guowen Xu, Hongwei Li, et al. Verifynet: Secure and verifiable federated learning. *IEEE Transactions on Information Forensics and Security*, 15:911–926, 2019.
- [Ye *et al.*, 2020] Dongdong Ye, Rong Yu, et al. Federated learning in vehicular edge computing: A selective model aggregation approach. *IEEE Access*, 8:23920–23935, 2020.
- [Zhan *et al.*, 2021] Yufeng Zhan, Jie Zhang, et al. A survey of incentive mechanism design for federated learning. *IEEE Transactions on Emerging Topics in Computing*, pages 1–1, 2021.
- [Zhang *et al.*, 2020a] Weishan Zhang, Qinghua Lu, et al. Blockchain-based federated learning for device failure detection in industrial iot. *IEEE Internet of Things Journal*, 8(7):5926–5937, 2020.
- [Zhang *et al.*, 2020b] Xianglong Zhang, Anmin Fu, et al. A privacy-preserving and verifiable federated learning scheme. In *ICC*, pages 1–6, 2020.
- [Zhang *et al.*, 2020c] Xiaoli Zhang, Fengting Li, et al. Enabling execution assurance of federated learning at untrusted participants. In *INFOCOM*, pages 1877–1886, 2020.
- [Zhang *et al.*, 2021a] Jingwen Zhang, Yuezhou Wu, and Rong Pan. Incentive mechanism for horizontal federated learning based on reputation and reverse auction. In *WWW*, pages 947–956, 2021.
- [Zhang *et al.*, 2021b] Qinnan Zhang, Qingyang Ding, et al. Blockchain empowered reliable federated learning by worker selection: A trustworthy reputation evaluation method. In *WCNCW*, pages 1–6, 2021.
- [Zhao *et al.*, 2020] Yang Zhao, Jun Zhao, et al. Mobile edge computing, blockchain and reputation-based crowdsourcing iot federated learning: A secure, decentralized and privacy-preserving system. *arXiv preprint arXiv:1906.10893v2*, 2020.
- [Zou *et al.*, 2021] Yue Zou, Fei Shen, et al. Reputation-based regional federated learning for knowledge trading in blockchain-enhanced iot. In *WCNC*, pages 1–6, 2021.