Decentral and Incentivized Federated Learning Frameworks: A Systematic Literature Review

Leon Witt, Mathis Heyer, Kentaroh Toyoda, *Member, IEEE*, Wojciech Samek, *Member, IEEE*, and Dan Li, *Senior Member, IEEE*

***Abstract*—The advent of federated learning (FL) has sparked a new paradigm of parallel and confidential decentralized machine learning (ML) with the potential of utilizing the computational power of a vast number of Internet of Things (IoT), mobile, and edge devices without data leaving the respective device, thus ensuring privacy by design. Yet, simple FL frameworks (FLFs) naively assume an honest central server and altruistic client participation. In order to scale this new paradigm beyond small groups of already entrusted entities toward mass adoption, FLFs must be: 1) truly decentralized and 2) incentivized to participants. This systematic literature review is the first to analyze FLFs that holistically apply both, the blockchain technology to decentralize the process and reward mechanisms to incentivize participation. 422 publications were retrieved by querying 12 major scientific databases. After a systematic filtering process, 40 articles remained for an in-depth examination following our five research questions. To ensure the correctness of our findings, we verified the examination results with the respective authors. Although having the potential to direct the future of distributed and secure artificial intelligence, none of the analyzed FLFs is production ready. The approaches vary heavily in terms of use cases, system design, solved issues, and thoroughness. We provide a systematic approach to classify and quantify differences between FLFs, expose limitations of current works and derive future directions for research in this novel domain.**

***Index Terms*—Blockchain, federated learning (FL), incentive mechanism (IM), survey.**

Manuscript received 28 July 2022; revised 2 November 2022; accepted 9 December 2022. Date of publication 22 December 2022; date of current version 6 February 2023. This work was supported in part by the Key-Area Research and Development Program of Guangdong Province under Grant 2021B0101400001; in part by the Tsinghua University-China Mobile Communications Group Company, Ltd. Joint Institute, KAKENHI from MEXT/JSPS, Japan, under Grant 18K18162; in part by the German Ministry for Education and Research (BMBF) through BIFOLD under Grant 01IS18025A and Grant 01IS18037I; and in part by the European Union's Horizon 2020 Research and Innovation Programme through COPA EUROPE under Grant 957059. *(Corresponding authors: Wojciech Samek; Dan Li.)*

Leon Witt is with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China, and also with the Department of Artificial Intelligence, Fraunhofer Heinrich Hertz Institute, 10587 Berlin, Germany (e-mail: leonmaximilianwitt@gmail.com).

Mathis Heyer is with the Department of Industrial Engineering, Tsinghua University, Beijing 100084, China, and also with the Aachener Verfahrenstechnik (AVT), RWTH Aachen University, 52074 Aachen, Germany.

Kentaroh Toyoda is with the Institute of High Performance Computing, Agency for Science, Technology and Research, Singapore, and also with the Faculty of Science and Technology, Keio University, Yokohama 223-8522, Kanagawa, Japan.

Wojciech Samek is with the Department of Electrical Engineering and Computer Science, Technical University of Berlin, 10587 Berlin, Germany, also with the Department of Artificial Intelligence, Fraunhofer Heinrich Hertz Institute, 10587 Berlin, Germany, and also with the Berlin Institute for the Foundations of Learning and Data, 10587 Berlin, Germany (e-mail: wojciech.samek@hhi.fraunhofer.de).

Dan Li is with the Department of Computer Science, Tsinghua University, Beijing 100084, China.

This article has supplementary downloadable material available at https://doi.org/10.1109/JIOT.2022.3231363, provided by the authors.

Digital Object Identifier 10.1109/JIOT.2022.3231363

I. Introduction

CENTRALIZED platforms in the domains of search engines, mobile applications, social media, chat, music, and retail have been dominating the respective industries over the past decades. Business models where digital services are exchanged for user data have developed into high-revenue industries with a few single entities controlling the global market within the respective domains [1]. The resulting concentration of user data in a small number of entities, however, poses problems, such as the risk of private data leaks [2] or an increasing power imbalance in favor of market-dominating parties [1], [3], [4] which has caused policymakers to enhance data protection for individuals [5]. The need for confidential AI extends beyond B2C markets, such as when entities within the health sector or Internet of Things (IoT) companies are not allowed to collaborate on a common AI model due to sensitive data.

A promising solution that enables the training of machine learning (ML) models with improved data security is federated learning (FL). In FL, complex models such as deep neural networks (DNNs) are trained in a parallel and distributed fashion on multiple end devices with the training data remaining local at all times. Federated averaging (`FedAvg`) [6] is a widely applied algorithm for FL where a central authority aggregates a global model from the locally trained models in an iterative process. In theory, FL makes not only previously withheld sensitive data accessible to the ML process but also enables efficient training by taking advantage of the ever-increasing computational power of IoT and mobile devices. However, the majority of FL research focuses on advancing the efficiency of the technology, yet incentives and decentralization are necessary requirements for many real-world FL applications, and the prerequisite for FL to evolve from academic research to real-world products with the potential to disrupt the vigorous data and AI industry [7]. In particular, incentives and decentralization address the two major design problems in FL: 1) the star topology of FL that introduces the risk for a Single Point of Failure (SPoF) as well as for authority abuse and prohibits use-cases where equal power among participants is a mandatory requirement and 2) the lack of a practical reward system for contributions of participants that

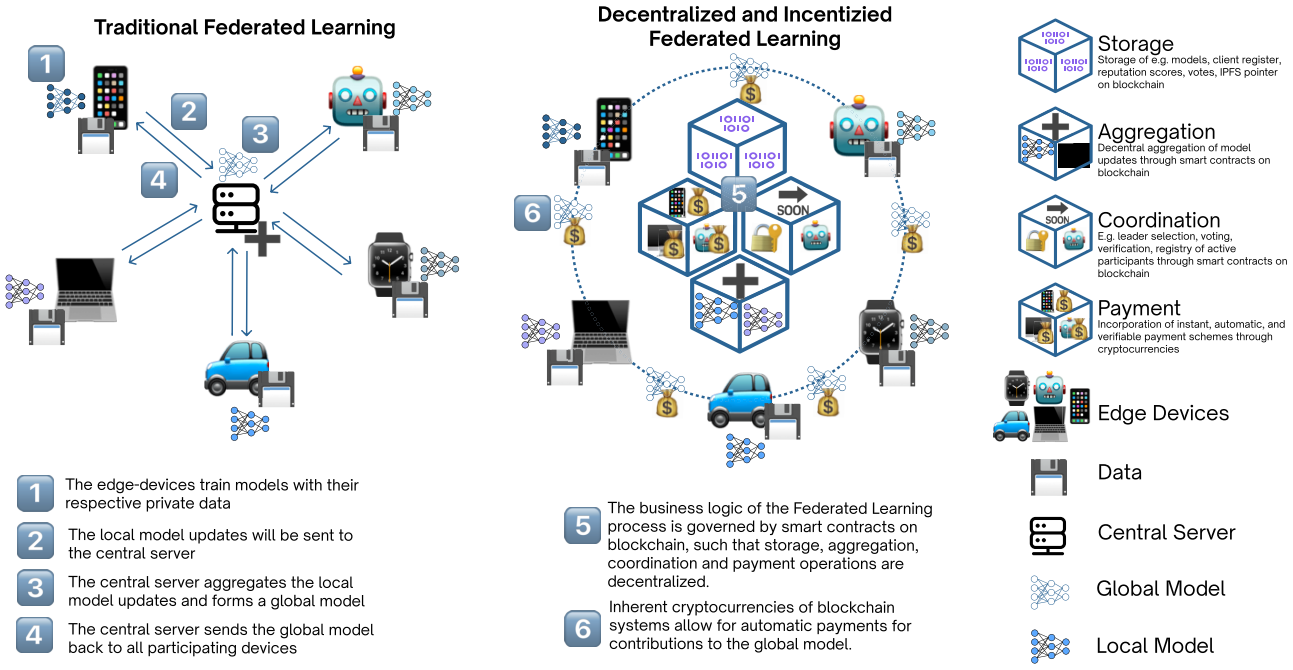

Fig. 1. FL versus decentralized and incentivized FLF.

hinders this technology from scaling beyond small groups of already entrusted entities toward mass adoption.

Although many proposals of incentivized and decentralized FL frameworks (FLFs) exist, we have not yet seen any full-fledged production-level FLF. To enhance the development toward production readiness, we compared state-of-the-art solutions despite their heterogeneity in terms of assumptions, use cases, design choices, special focus, and thoroughness by providing a general and holistic comparison framework.

Specifically, we undertake a systematic literature review (SLR) examining all relevant articles from 12 scientific databases in the domain of computer science. 422 publications were queried from these databases and filtered for relevant contributions, resulting in 40 papers remaining after three filtering steps. To the best of our knowledge, this is the first comprehensive survey on the design of both decentralized and incentivized federated artificial intelligence systems. The contribution of this article is threefold.

1) The first comprehensive systematic survey study on the combined topic of decentralized and incentivized FL based on the standardized preferred reporting items for systematic reviews and meta-analyses (PRISMA) process, ensuring transparency and reproducibility of the work.
2) A novel comparison framework for an in-depth analysis of incentivized and decentralized FLFs which goes beyond existing survey papers by: a) pointing out the limitations and assumptions of the chosen game-theoretic approaches; b) analyzing the existing solutions based on computational and storage overhead on the BC; and c) an in-depth analysis of the performed experiments.
3) Based on this comparison, we have identified limitations of recent work, derived tradeoffs in the design choices, and derived future research directions of decentralized and incentivized FLFs.

The remainder of this article is structured as follows. In Section II, we present the technical background of the distributed ledger technology and mechanism design (MD) in FL systems. In Section III, we provide an overview of existing surveys on this topic and their respective problem statements. Section IV summarizes the findings of the SLR and answers research questions concerning general applications of the FLFs, BC features, incentive mechanisms (IMs), and experiments. We derive limitations and further research directions in Section V. Finally, Section VI concludes this literature review.

In Appendix in the supplementary material, we outline the details of the methodology, the search process, the selection process, the search terms, and the number of papers retrieved from the respective databases to ensure reproducibility.

II. Preliminaries

The following sections discuss the fundamentals of FL, distributed ledger technology, and incentive design, and outline how these technologies are integrated in FLFs.

A. Federated Learning

FL is an ML technique where multiple actors collaboratively train a joint ML model locally and in parallel, such that the individual training data does not leave the device as depicted on the left side of Fig. 1. This decentralized approach to ML was first introduced by Google in 2016 [6] and addresses two key issues of ML: 1) the high computational effort for model training is relocated from a single central actor to a network of data-owning training devices and 2) as the training data remains on the edge devices, previously inaccessible data of

privacy-concerned actors can be integrated into the training process. Thus, "data islands" are prevented.

FL tasks can be classified in horizontal and vertical settings. In horizontal FL, the actors possess the same type of information on different entities, whereas, in vertical FL, the actors have available different information features on the same entity. The latter comes with the additional challenge of data alignment and information exchange [8]. Furthermore, the FL setting can range from a few collaborating entities, i.e., cross-silo (CS), to a federated system of millions of devices, i.e., cross-device (CD).

The `FedAvg` algorithm [6] is a widely adopted optimization algorithm for FL. Its objective is to minimize the empirical risk of the global model θ, such that

$$\arg\min_{\boldsymbol{\theta}} \sum_{i} \frac{|S_i|}{|S|} f_i(\boldsymbol{\theta}) \tag{1}$$

where for each agent i, f_i represents the loss function, S_i is the set of indexes of data points on each client, and $S \coloneqq \bigcup_i S_i$ is the combined set of indexes of data points of all participants. For that, the calculated gradients from the respective local model training get aggregated in three communication rounds.

1) A central server broadcasts a first model initialization θ_{init} to a subset of participating clients.[1]
2) These clients individually perform iterations of stochastic gradient descent over their local data to improve their respective local models θ_i.
3) In order to create a global model θ, all individual models θ_i are then sent back to the server, where they are aggregated (e.g., by an averaging operation). This global model is used as the initialization point for the next communication round.

Optimization algorithms for the FL case are open research [7] and variations of `FedAvg` exist, e.g., `FedBoost` [9], `FedProx` [10], `FedNova` [11], `FedSTC` [12], and `FetchSGD` [13].

B. Blockchain: Distributed Ledger Technology

A distributed ledger kept by nodes in a peer-to-peer network is referred to as blockchain, first invented by Satoshi Nakamoto through Bitcoin [14]. Cryptographic connections of information enable resistance to alteration and immutability. A peer-to-peer consensus mechanism governs the network, obviating the requirement for central coordination [15]. The introduction of general-purpose BCs with smart contract capability supports Turing-completeness [16] and has allowed for the creation of decentralized, immutable, and transparent business logic on top of the BC. Here, smart contracts are computer programs that are decentrally stored on a distributed BC network and automatically executed when a predetermined condition is met.

1) BC Data Architecture: Even though the data structure varies across different BC, the structure can roughly be categorized into six layers.

1) *Data Layer:* The data layer defines how new information will get stored and how the respective blocks are designed. Blocks typically contain a block header and block body [14], [16]. The block header is a collection of metadata about the block and a summary of the transactions included in the execution payload. In Ethereum, the block body is a bundled unit of information that include an ordered list of transactions and consensus-related information [16].
2) *Network Layer:* BCs are peer-to-peer networks with nodes that require defined protocols for communication between them. The protocol comprises exchanging requests and answers between particular nodes (one-to-one communication) as well as "gossiping" information (one-to-many communication) through the network [17]. To make sure they are delivering and receiving the right information, each node must abide by a set of networking regulations.
3) *Consensus Layer:* The consensus mechanism refers to the entire stack of protocols, incentives, and ideas that allow a network of nodes to agree on the state of a BC. The two major types of consensus mechanisms for public BCs are Proof of Work (PoW) and Proof of Stake (PoS). PoW is used in Bitcoin [14] and the former version of Ethereum [16], where nodes in the network compete to find a specific hash value below a given number to prove that a certain amount of a specific computational effort has been expended in order to add blocks to the network. The fact that it would take 51% of the network's computer power to commit fraud on the chain ensures the network's security [14]. In PoS-based consensus mechanisms, blocks can be added by randomly chosen validators who have staked significant amounts of cryptocurrencies. The system is designed such that the staked cryptocurrencies get slashed when the validators act maliciously, securing the network crypto-economically [18].
4) *Incentive Layer:* To incentivize participation in the consensus mechanism, BC systems contain an inherent cryptocurrency reward for either contributing computational effort (i.e., PoW) or staking cryptocurrencies (i.e., PoS). FL-specific BC systems may reward operations beyond securing the BC, such as the storage of ML models [19], [20], [21], the aggregation of gradients [19], [22], [23], [24], [24], or contribution calculations [22], [23], [25].
5) *Contract Layer:* Smart contracts are simple programs that run on the respective virtual machine of the BC. The Ethereum virtual machine (EVM) is a Turing-complete environment for smart contracts most commonly used across other BCs, such as Polygon [26], BNB Chain [27], and Avalanche [28]. A smart contract is a collection of code (its functions) and data (its state) that resides at a specific address on the respective BC.
6) *Application Layer:* Decentralized applications (Dapps) on top of smart contracts cannot be censored, allow for anonymous participation, have zero downtime, and are compatible with other Dapps on the same BC.

[1] Clients, workers, and agents are used interchangeably.

2) BC-Based FL to Ensure Equal Power: Due to its intrinsic features, the distributed ledger technology is capable of mitigating open issues in the FL context.

1) *Decentralization:* Workers are subject to a power imbalance and an SPoF in server-worker topologies. A malicious server might refuse to pay reward payments or exclude employees at will. Furthermore, a server-worker architecture is incompatible with a situation in which numerous entities have a shared and equal stake in the advancement of their respective models. BC technologies' decentralization provides a federal system for entities of equal authority without the need for a central server.
2) *Transparency and Immutability:* Data on the BC can only be updated, not erased, as every peer in the system shares the same data. In an FL environment, a clear and immutable reward system ensures worker trust. On the other side, each client is audited, and as a result, can be held accountable for malevolent activity.
3) *Cryptocurrency:* Many general-purpose BC systems include cryptocurrency capabilities, such as the ability to incorporate payment schemes within the smart contract's business logic. Workers can be rewarded instantly, automatically, and deterministically based on the FL system's reward mechanism.

Therefore, BC systems [16], [29], [30] have the potential to mitigate the first issue of FL by ensuring trust through their inherent properties of immutability and transparency. They enable decentralized federations to mitigate dependencies on a central authority.

Fig. 1 (right) depicts the four major functions BC helps to facilitate in the FL process, namely, aggregation, coordination, storage, and payment.

1) *Aggregation:* In regular `FedAvg`-based FL, a central authority collects and aggregates the clients' gradients. BC can decentralize the process by performing the aggregation on-chain.
2) *Coordination:* Leader selection for the aggregation process, voting, verification, and onboarding of new clients are necessary for real-world FL but undefined steps in contemporary research. BC can provide a trustless and transparent infrastructure for those steps.
3) *Storage:* BC provides immutable and transparent storage for information where access is shared among clients.
4) *Payment:* BCs inherent functionality of cryptocurrencies allows for automated payment schemes to reward clients for the exerted effort. These four major operations of BC in FL are discussed in detail in Section IV-B.

C. Incentive Mechanism

FL use cases where pseudo-anonymous clients are expected to participate and invest their data and computational power cannot assume altruism but require compensation in any real-world scenario. MD, which is a field of economics, attempts to implement a protocol, system, or rule so that the desired situation (e.g., every participant contributes informed truthful model updates) is realized in a strategic setting, assuming that each participant acts rationally in a game theoretic sense [31].

The purpose of incorporating MD into FL is to incentivize clients to: 1) put actual effort into obtaining real and high-quality signals (i.e., training the model on local data) and 2) the submit model updates truthfully despite not being monitored directly. Such incentives can be distributed using BC infrastructure and their underlying cryptocurrencies. An appropriately designed mechanism ensures a desired equilibrium when every worker acts rationally and in their own best interest. Moreover, a mechanism ideally has low complexity and is self-organizing, avoiding the need for trusted execution environments (TEEs) [32] or secure multiparty computation (MPC), yet makes assumptions about the degree of information available.

The process of designing an FL protocol with MD consists of: 1) designing a mechanism and 2) a theoretical analysis. The former determines the whole procedure of FL including a reward policy. The reward policy defines: 1) how to measure clients' contribution to the overall model and 2) how to distribute rewards. Measuring contributions in FL is challenging since the aggregated gradients do not reveal explicit information about their effect on the overall performance. Additionally, a myriad of design choices for reward distribution exists, e.g., whether rewards should be given to the top contributor (i.e., winner takes all) or to multiple workers where rewards are equally distributed among all contributors or unequally distributed based on the workers' contribution.

1) Theories Behind Mechanism Design: We classify underlying theories for MD broadly into two categories, namely: 1) game theory and 2) auctions, based on [33].

The theory assumes that the clients' utility U is defined by expected profits Π (e.g., prizes) minus costs C (e.g., costs of model training and data collection)

$$U = \Pi - C. \tag{2}$$

Assuming individual rationality (IR), clients choose their actions to maximize their utilities. In this context, the interactions of choices that produce outcomes concerning utilities are referred to as games in the scientific literature. Games can be classified into cooperative and noncooperative. A noncooperative game is a game where each client individually determines their strategies so that their utilities are maximized, while a cooperative game maximizes the utility of the group. A game is called imperfect when a client does not know the others' information (e.g., utilities, strategies, etc.). When all the clients know others' information, such a game is called perfect. Popular game-theoretic methods applied in FL are Stackelberg games, contest theory, and contract theory.

Stackelberg Game: A leader (e.g., a task requester) determines their strategy, and followers (i.e., clients or workers) determine theirs according to the leader's action [34]. A task requester can be a leader who determines a reward first, and clients determine their effort and how much they should exert based on the condition. Seminal work on Stackelberg games includes, e.g., Khan et al. [35] who motivated the

modeling of FL as a Stackelberg game and propose an IM based on their best response algorithm. Another example of a Stackelberg game-based IM in centralized FL settings can be found with Zhang et al. [36]. The authors tackle the challenges of information secrecy and contribution measurement by training a deep reinforcement learning-based IM that allows optimal pricing for the central server and optimal training strategies for the participating clients.

Contest Theory: Contrary to Stackelberg games, clients need to exert efforts before joining the contest [37]. The process of FL can be seen as a contest as clients must train a model with their own local data sets while they are not guaranteed to receive prizes.

Contract Theory: In contract theory, an employer has to agree on a contract with employees given the situation that the employees may claim false capabilities [38]. This could be the case in FL as task requesters do not exactly know clients' capabilities (e.g., cost, computational resources, etc.).

Auctions: Auctions are applicable in designing the mechanisms of FLF as they optimally allocate resources, such as computational resources or amount of data, based on clients' reports. A task requester posts an FL task, potential clients bid with sealed information, such as computational cost and resource, and the requester assigns an FL task to winning clients. There are several auctions to determine winners (e.g., first-price sealed-bid (FPSB) auction, second-price sealed-bid (SPSB) auction, Vickrey–Clarke–groves (VCG) auction, and all-pay auctions) [39]. A detailed analysis of all auction types is beyond the scope of this article which is limited to the most popular auctions applied in FLFs: an FPSB auction is an auction where no bidders know others' bids and the highest bidder pays the price that they bid. An SPSB auction is similar to the FPSB, but the highest bidder only needs to pay the price that the second highest bid. A VCG auction is a sealed-bid auction for multiple resources. It is designed to achieve socially optimal resource allocation by charging winners of an auction the social loss they cause to others. This prevents clients from bidding their false valuations to win. An all-pay auction is an auction where all bidders need to pay regardless of whether or not they win. A Tullock contest is one of the most famous all-pay auctions [40].

2) Desirable Properties: Game theory and auctions provide a strong guarantee that a designed mechanism possesses desirable properties. Zeng et al. [8] summarized the main properties that a mechanism should possess in FL, namely, incentive compatibility (IC), IR, Pareto efficiency (PE), collusion resistance (CR) and fairness, and balanced budget (BB). IC is fulfilled when entities cannot be better off by deviating from their optimal strategies, and IR refers to the assumption that contributors would not participate if their respective utility was negative as in (2). A game is PE when it guarantees that the sum of profits is maximized. CR is achieved when no participant can be better off by colluding with others. A game is said to be fair when fairness, e.g., a payoff to the contribution, is preserved [41]. Finally, a game is BB when it is sustainable without external money inflows. Section IV-C discusses how these theories and properties are adopted in the FLFs' MD.

TABLE I
COMPARISON OF RELATED SURVEY PAPERS

Ref.	BC	FL	IM	Experiment Analysis
[43]	✓	✓		
[42]	Partially	✓		
[8]	Partially	✓	✓	
[44]		✓	✓	
[33]		✓	✓	
[45]	Partially	✓	Partially	
[46]	✓	✓	Partially	
This work	✓**(Detailed)**	✓	✓	✓

III. RELATED SURVEYS

To the best of our knowledge, this is the first analysis of holistic frameworks for fully decentralized FL with rewards for the participating clients. Yet, we have identified several survey papers in the context of either MD and FL [8], [33], [42] or BC and FL [8], [43], [44]. Table I shows the comparison of the related survey papers and our own.

Hou et al. [43] investigated the state-of-the-art BC-enabled FL methods. They focus on how BC technologies are leveraged for FL and summarized them based on the types of BC (public or private), consensus algorithms, solved issues, and target applications.

The other related survey papers focus on IM for FL [8], [33], [42], [44]. Zhan et al. [42] surveyed the IM design dedicated to FL. They summarize the state-of-the-art research efforts on the measures of clients' contribution, reputation, and resource allocation in FL. Zeng et al. [8] also surveyed the IM design for FL. However, in this publication, the authors focus on IM, such as Shapley values, the Stackelberg game, auction, context theory, and reinforcement learning. Besides [8] and [42], Ali et al. [44] also summarized involved actors (e.g., number of publishers and workers), evaluation data sets as well as advantages and disadvantages of the mechanisms and security considerations. Tu et al. [33] provided a comprehensive review of economic and game theoretic approaches to incentivize data owners to participate in FL. In particular, they cluster applications of Stackelberg games, noncooperative games, sealed-bid auction models, reverse action models as well as contract and matching theory for incentive MD in FL. Nguyen et al. [45] investigated opportunities and challenges of BC-based FL in edge computing. Finally, Wang et al. [46] surveyed BC-based FLF with a particular focus on FLF system compositions.

As revealed from our analysis and summarized in Table I, the existing survey papers lack a holistic analysis of FLFs that are both decentralized *and* incentivized. Such a review is urgently needed since the simultaneous implementation of these features comes with additional interdisciplinary challenges while being crucial to establishing a fair and trustworthy FLF to the benefit of the data owner. To fill this research gap, this article provides the first SLR on the topic of BC-enabled decentralized FL with IM.

IV. SYSTEMATIC LITERATURE REVIEW

The goal of the SLR is the identification of decentralized collaborative learning solutions where participation is

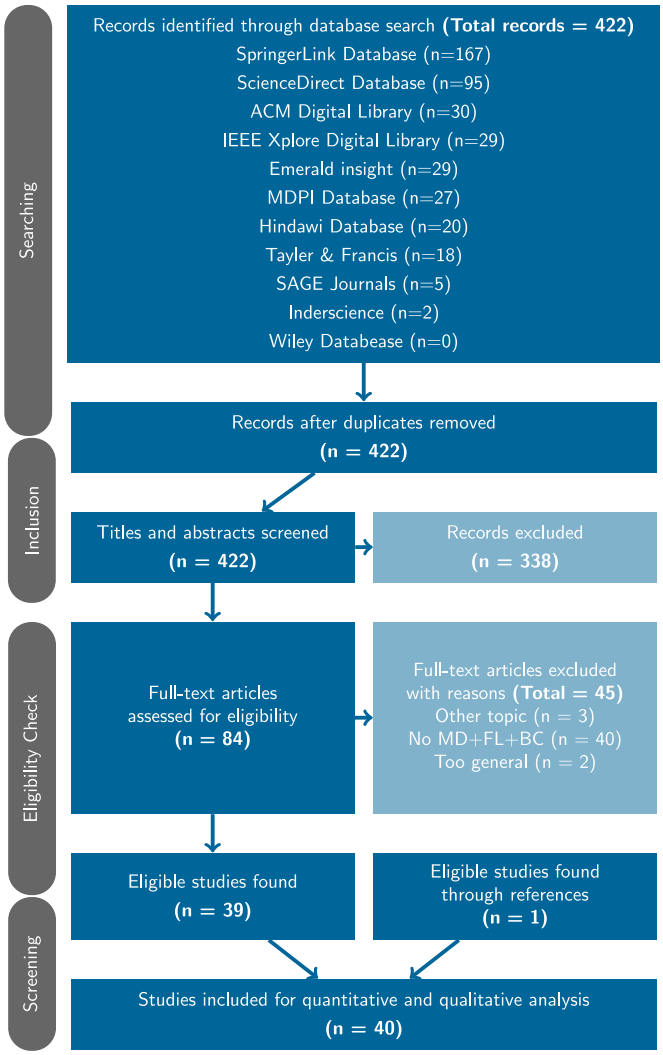

Fig. 2. Flow diagram of the search and screening step in the PRISMA methodology.

rewarded. For that, relevant publications are retrieved, filtered, and analyzed following a methodical, reproducible procedure. The procedure is inspired by the PRISMA methodology [47] and augmented with the guide for information systems proposed by Okoli and Schabram [48] and Kitchenham [49]. The five core steps of the systematic approach include: 1) defining research questions; 2) searching for literature; 3) screening; 4) reviewing; and 5) selecting and documenting relevant publications, and extracting relevant information.

A detailed visualization of the search and screening step can be found in Fig. 2. Details about the SLR are outlined in Appendix in the supplementary material.

Sections IV-A–IV-E systematically present the results of the literature review by answering our five research questions.

RQ1 *Overview:* a) What are the possible applications of FLF? and b) What problems were solved?

RQ2 *BC:* a) What is the underlying BC architecture? b) How is BC applied within the FLF and what operations are performed? and c) Is scalability considered?

RQ3 *IM:* a) How are IMs analyzed? and b) How are the contributions of workers measured?

RQ4 *FL:* a) Is the performance of the framework reported? b) How comprehensive are the experiments? c) Are non independent and identically distributed (non-IID) scenarios simulated? d) Are additional privacy methods applied? and e) Is the framework robust against malicious participants?

RQ5 *Summary:* What are the lessons learned from the review?

Each section is complemented by an explanatory table that classifies the considered papers according to categories defined in Table II. Hereafter, we use not applicable (n.a.) for the items that are n.a. Likewise, we use not stated (n.s.) for the items that should be stated but are not. We also use ✓ for the items that satisfy a condition while leaving cells empty when they do not.

A. RQ 1: Overview

1) RQ 1-1 (What Are Possible Applications of FLF?): Table III shows the summary of FLFs. Although most of the surveyed papers do not target specific applications (28 out of 40) due to the generalizability of neural networks, some are dedicated to specific applications, namely, IoT (6 out of 40), Internet of Vehicles (IoV) (5 out of 40), and Finance (1 out of 40). Applications of FLF in special domains may require additional constraints and characteristics. The heterogeneity of the required properties across those domains leads to vast differences in the design choices of function, operations, and storage of BC, contribution measurement, and privacy requirements.

One of the major application scenarios is IoT (e.g., [51], [52], and [63]). Sensor-equipped devices collect environmental information and execute model updates thanks to advances in neural engines while edge servers are often assumed to aggregate models that are trained by local sensor devices. For instance, power consumption measured at smart homes can be used for training an AI model of energy demand forecast [72]. Zhao et al. [52] proposed an FLF for smart home appliance manufacturers to obtain AI models trained with their customers' usage information.

Some solved issues pertaining to the IoT-based FL [51], [68]. Zhang et al. [51] proposed an FL-based failure device detection method that takes into account the fact that sensor readings are often imbalanced since sensors are, in general, not deployed uniformly in a sensing area. They propose a modified FedAvg algorithm called centroid distance weighted federated averaging (`CDW_FedAvg`) to obtain accurate models when local data sets are imbalanced at the devices. As sensor devices may not have enough resources to solely train neural networks, it is important to determine whether to delegate computationally intensive tasks to edge servers. Qu et al. [68] proposed an algorithm to determine whether to offload computation to edge servers when communications between IoT devices and edge computers are unreliable. Beyond, Liu et al. [82] reflected on FL in the context of 6G communication and how both technologies are expected to empower each other. The review pinpoints communication cost, security, privacy, and training interference as the key challenges of FL and 6G communication.

TABLE II
DEFINITION OF COLUMNS IN THE OVERVIEW TABLES

Table	Column	Definition	Examples
FL (TABLE 3)	Application	Fields of applications	Generic, IoT
	Setting	Whether a setting of FL is given	CS, CD
	Actors	Actors assumed in the FLF	Workers
	Setup	Whether how a system is set up was given (e.g., who deploys a BC)	✓
	Domains	To which domain each work contributes	
		SPoF: Single point of failure	✓
		BC: Blockchain	✓
		FL: Federated learning	✓
		IM: Incentive mechanisms	✓
		CM: Contribution measurement	✓
		SP: Security & privacy	✓
BC (TABLE 4)	Operations	Whether the following operations are executed on the BC	
		Agg.: Model aggregation	✓
		Cor.: Coordination	✓
		Pay.: Payment	✓
		Str: Storage	✓
	BC	BC used in the FLF	Ethereum
	Consensus	Consensus mechanism used	PoW
	On-chain	Items: Types of data stored on-chain	✓
		Eval.: Whether the on-chain storage amount was evaluated	✓
	Off-chain	Whether off-chain storage (e.g., IPFS) is used	✓
	Scalability	Whether scalability is considered	✓
IM (TABLE 5)	Sim.	Whether a game was evaluated via simulation	✓
	Theoretical analysis	Whether a game was theoretically analyzed	✓(Contract theory)
	Contribution measurement	Costs: Costs considered in analysis	Energy
		Metrics: Metrics to validate workers' contribution	Accuracy
		Abs.: Whether the metric is absolute	Accuracy
		Rel.: Whether the metric is relative	Accuracy
		Rep.: Whether reputation is considered	Accuracy
		Validator: Actors that validate workers' contribution	Task requesters
Experiments (TABLE 6)	Tasks	Types of ML tasks	
		Clf: Classification	✓
		Rgr: Regression	✓
	Datasets	Datasets used for evaluation	MNIST
	#Clients	Number of clients in the experiments	10
	Algorithms	FL algorithms used	FedAvg
	Privacy	Whether privacy protection methods were applied	✓
	Non-IID	Whether the non-IID condition was assumed	✓
	Adversaries	Whether security analysis was given against the following attacks	
		BT: Blockchain tampering	✓
		RP: Random model poisoning	✓
		RT: Reputation tampering	✓
		SP: Systematic model poisoning	✓
	Imp.	Whether the BC part was implemented for evaluation	✓
	Per.	Whether the performance of FL models was measured	✓

FL is beneficial to many scenarios in ITS or IoV, e.g., optimized routing, congestion control, and object detection for autonomous driving ([20], [59], [64], [74], [76]). Vehicles collect local information and train local models with collected data. Models are often aggregated by devices called road side units (RSUs) and mobile edge computers (MECs) which are often deployed on the road. In IoV, the CD setting is often preferred as mostly the same types of sensors are used to measure road conditions, and thus the common neural network model structure is shared by vehicles. As different locations have different road conditions, users need locally optimized models, and thus scalability is a key issue. Furthermore, we need extra protection for users' location privacy. Zou et al. [74] proposed an FLF for a knowledge trading marketplace where vehicles can buy and sell models that vary geographically. Chai et al. [59] proposed multiple BCs to deal with geographically dependent models. Kansra et al. [64] integrated data augmentation, a technique to synthetically generate data such as images, into FL to increase model accuracy for ITS, such as autonomous driving and road object identification. Wang et al. [76] proposed an FLF dedicated to the crowdsensing of unmanned aerial vehicles (UAVs). As UAVs are often

TABLE III
OVERVIEW OF DECENTRAL AND INCENTIVIZED FLFS

Ref.	Application	Setting	Actors	Setup	Domains					
					SPoF	BC	FL	IM	CM	SP
[22]	Generic	n.s.	Workers	✓	✓		✓	✓	✓	
[19]	Generic	CS	Workers	✓	✓			✓		✓
[50]	Generic	n.s.	Contributors, miners	✓	✓		✓			✓
[51]	IoT	n.s.	Clients, central organization		✓		✓	✓		✓
[52]	IoT	n.s.	Clients, miners	✓	✓					✓
[53]	Generic	n.s.	Aggregation servers, workers	✓				✓		
[54]	Generic	n.s.	Workers, task publishers, miners					✓		
[20]	IoV	n.s.	MEC, MBS, ADV					✓		✓
[25]	Generic	n.s.	Model requesters, FL servers, clients			✓		✓		
[55]	Generic	CD	Administrator, requesters, workers, validators	✓				✓		✓
[21]	Generic	n.s.	Central server, workers					✓		
[56]	Generic	CS	Workers, leaders, aggregation server					✓		✓
[57]	Generic	n.s.	Edge devices, fog nodes, cloud	✓	✓					
[58]	Generic	CS	Users, edge devices, cloud		✓					✓
[59]	IoV	CD	Vehicles, RSUs, BSs		✓					
[60]	Generic	CD	Task publishers, workers					✓		
[61]	Generic	n.s.	Trainers, buyers, reporters, data processors	✓	✓			✓		
[62]	Generic	CS	Data owners	✓				✓		
[63]	IoT	CD	Local devices, BSs, MEC nodes	✓	✓			✓		
[64]	IoV	CD	Vehicle edge nodes, BC nodes		✓		✓	✓		
[65]	Generic	n.s.	Trainer nodes	✓	✓			✓		✓
[66]	Generic	n.s.	Model initiators, computing partners, validators		✓					
[67]	Finance	CS	Follower candidates, leader nodes	✓	✓	✓		✓		
[68]	IoT	CD	Users, edge server, cloud server							
[69]	Generic	CD	Worker nodes	✓	✓					✓
[70]	Generic	n.s.	Task publishers, parties, miners, smart contract	✓	✓					✓
[71]	Generic	n.s.	Requesters, workers, crowdsourcing platform		✓			✓	✓	✓
[72]	IoT	n.s.	IoT devices, edge servers, central cloud server		✓					✓
[23], [24]	Generic	n.s.	Task requesters, workers	✓	✓			✓		
[73]	Generic	n.s.	Requesters, workers		✓				✓	
[74]	IoV	CD	RSUs (aggregators), MECs, vehicles (workers)		✓			✓	✓	
[75]	IoT	n.s.	Requesters, edge servers (workers), data collectors		✓			✓		✓
[76]	IoV	n.s.	UAVs (sensors, workers), requesters, MECs		✓	✓			✓	✓
[57]	Generic	n.s	Requesters, data-arbitrators, workers		✓					
[77]	Generic	n.s.	Requesters, miners		✓	✓		✓	✓	
[78]	Generic	n.s.	Miners, workers			✓				
[79]	Generic	n.s.	Requesters, workers, aggregation servers		✓			✓	✓	✓
[80]	Generic	n.s.	Administrator, requesters, workers, miners		✓			✓		
[81]	Generic	CS, CD	Workers, miners		✓		✓			

equipped with multiple sensors and can be easily deployed to sensing areas, an FLF with UAVs has a huge benefit for ITS applications, such as traffic monitoring and public surveillance.

Finance is the other domain that we found in the surveyed papers. He et al. [67] proposed an FLF for commercial banks to better utilize customers' financial information. Financial information, such as credit level, risk appetite, solvency, movable, and real estate owned are crucial sources to understanding the characteristics of customers of financial services. However, it is too sensitive to directly use them for data mining. Hence, an FLF is a viable framework for financial information management.

2) RQ 1-2 (What Problems Were Solved?): The problems solved by the papers can be categorized into: 1) the SPoF issue in FL; 2) BC-related issues; 3) lack of clients' motivation; 4) how to fairly evaluate clients' contribution; and 5) security and privacy issues.

Most of the papers (29 out of 40) propose a system architecture of FLF to solve the problems of an SPoF in the current centralized server-clients architecture. More specifically, this issue is rooted in the structure of the original FL where an aggregation server is collecting local model updates from clients in a centralized manner. The idea to mitigate this issue is to decentralize the processes involved in FL using BC technologies. Each paper proposes operations, functions, and protocols processed in and outside the smart contract. Furthermore, some solve the issue of scalability in the FLF (e.g., [59] and [63]) and BC-related issues such as energy waste of consensus algorithms (e.g., [77] and [78]). We will go into the proposed system architectures and BC-related issues in Section IV-B.

An IM is integrated into FLFs to solve the problem of a lack of client motivation. The basic idea is to give monetary incentives to clients in return for their effort in training a local

model. The IM is also leveraged to solve the model poisoning attack which is an attack on a model update to deteriorate the quality of a global model by malicious clients' providing bogus local model updates. The idea for demotivating such attacks is to devise an IM that penalizes malicious activities. Furthermore, a reputation score based on contribution is also useful to screen potentially malicious clients. Here, we need a contribution measurement metric to fairly evaluate the quality of clients' model updates and detect the attacks. Details about the proposed IMs and contribution measurements will be covered in Section IV-C.

Twenty out of forty papers propose approaches to solve issues related to security and privacy. With few exceptions (i.e., attacks on reputation [60], [65], and [78]), both security and privacy issues are rooted in local model updates. The security issue is related to the model poisoning which we mentioned above, while the privacy issue is related to sensitive information that might be leaked from the local updates. We will further summarize the works that solve the security and privacy issues in Section IV-D.

B. RQ 2: Blockchain

1) RQ 2-1 (What Is the Underlying BC Architecture?): Table IV shows the overview of BC features. The BC system and its underlying consensus mechanism are an influential part of the FLFs infrastructure. FLFs are heterogeneous in terms of architecture, operation and storage requirements, contribution calculation, actors, and applied cryptography. Customized and tailored BC solutions may be required with respect to the underlying use case. Due to its restrictive scalability in terms of computation and storage, most of the analyzed FLFs apply BC as a complementary element in a more complex system, with a few exceptions [22], [23], [24], [55], [69]. BC systems themselves are complex distributed systems, heterogeneous across many dimensions, yet can roughly be categorized into public, private, and permissioned BCs.

1) *Public:* BCs are open access where participants can deploy contracts pseudo-anonymously.
2) *Private:* BCs do not allow access for clients outside the private network and require an entity that controls who is permitted to participate.
3) *Permissioned:* BCs are private BCs with a decentralized committee that controls the onboarding process.

Note that the FLFs that utilize open-source public BCs, such as Ethereum [50], [51], [53], [55], [56], [57], [58], [65], [69], [71], [72], Stellar [66], and EOS [73] were not deployed on the respective public BC in the experiments due to the enormous costs this would incur. Hyperledger Fabric [30] or Corda [83] are permissioned BCs running on private networks, allowing for faster throughput through a limited amount of potential nodes. This makes these frameworks more suitable for applications where BC replaces computationally expensive operations, such as aggregation or storage of neural network models.

The consensus protocol ensures the alignment and finality of a version across all distributed nodes without the need for a central coordination entity. While PoW is the most common mechanism applied in Bitcoin and Ethereum, it comes at the cost of wasting computational power on brute-forcing algorithmic hash calculations for the sole purpose of securing the network. Since many operations within the FLF frameworks are computationally expensive, these tasks can be integrated into the consensus mechanism which creates synergy and might be a better use of resources. Examples of consensus mechanisms can be found where the model accuracy is verified (Proof-of-Knowledge [59]), reputation scores are checked (Proof-of-Reputation [54]), the model parameters are securely verified (Proof-of-Federated-Learning [77]), the Shapley value is calculated for contribution measurement [25] or verification of capitalizing on efficient AI hardware (Proof-of-Model-Compression [78]).

2) RQ 2-2 (How Is BC Applied Within the FLF and What Operations Are Performed?): The BC technology is applied to mitigate the SPoF and power imbalance of the server-worker topology of traditional FL through a transparent, immutable, and predictable distributed ledger. Embedded cryptocurrencies suit the useful property of real-time reward payments for predefined actions at the same time. In general, Turing-complete smart-contract-enabled BCs allow for a variety of possible complementary features for the FL training process, namely, aggregation, payment, coordination, and storage.

1) *Aggregation:* The aggregation of model parameters, can be performed by a smart contract on top of BC [19], [21], [23], [24], [63], [67]. Since BC is assumed to be failure resistant, this strengthens the robustness against possible single-point of failure of an aggregation server. In addition, the deterministic and transparent rules of smart contracts ensure inherent trust with an equal power distribution among participants, while the transparency ensures auditability of contributions. Yet since every node in the BC has to compute and store all information, submitting a model to the smart contract for aggregation causes overhead in terms of both computation and storage on the BC. Assuming n FL-workers and m BC nodes over t rounds, the BC scales with $\mathcal{O}(t * n * m)$ which questions the feasibility of on-chain aggregation. There are two papers that try to reduce data size for on-chain aggregation. Witt et al. [22] proposed a system where 1-bit compressed soft-logits are stored and aggregated on the BC saving communication, storage, and computation costs by orders of magnitude. Feng et al. [63] employed a framework based on two BC layers where the aggregation process is outsourced to a mobile edge server.
2) *Coordination:* Applying BC to coordinate and navigate the FL process allows for decentralization without the heavy on-chain overhead. Instead of aggregating the model on-chain, letting the BC choose a leader randomly can ensure decentralization [52], [59], [69], [80]. Another way BC coordinates the FL process is by enabling the infrastructure for trustless voting atop the BC. Voting on the next leader (aggregator) [66], [67] or on each other's contributions [23], [24], [80] further democratizes the process. Beyond explicit coordination operations, such as voting or leader selection,

TABLE IV
OVERVIEW OF BC FEATURES

Ref.	Operations				BC	Consensus	On-chain		Off-chain	Scalability
	Agg.	Cor.	Pay.	Str.			Items	Eval.		
[22]	✓	✓	✓		Agnostic	n.a.	1-bit results of all participants	✓		✓
[19]	✓		✓		Corda V3.0	Algorand	Gradients			
[50]			✓		Ethereum	PoW	n.s.		✓	
[51]			✓		Ethereum	PoW	Contribution, Merkle tree			
[52]		✓	✓		n.s.	n.s.	Models		✓	
[53]			✓		Ethereum	PoW	n.s.	✓		
[54]				✓	TrustRE	PoR	Reputation scores			
[20]				✓	Custom	PoW	Model updates			
[25]			✓		Custom	PoSap	SV values, tasks			
[55]		✓	✓	✓	Ethereum	PoW	Tasks		✓	
[21]	✓		✓	✓	n.s.	PoW	Model updates, metadata			
[56]			✓		Ethereum, HF	PoW	PoT records	✓		
[57]		✓		✓	n.s.	n.s.	Reputation scores		✓	
[58]		✓	✓		Ethereum	PoW	Users' addresses			
[59]			✓	✓	n.s.	PoK	Local models, loss, signatures			✓
[60]				✓	Corda V4.0	PBFT	Reputation scores		✓	
[61]		✓	✓	✓	n.s.	Custom	Client info, model parameters			✓
[62]	✓				n.s.	n.s.	Masked gradients, global models			
[63]	✓		✓	✓	HF	Raft	Local updates			✓
[64]		✓	✓		n.s.	n.s.				
[65]		✓	✓		Ethereum	PoW	IPFS CIDs of models		✓	
[66]		✓	✓		Stellar	n.s.	IPFS CIDs of models		✓	
[67]	✓	✓		✓	Custom	Raft	Local and global models, loss			
[68]				✓	n.s.	Custom	Local and global models			
[69]	✓	✓			HF	Raft	Parameters			✓
[69]			✓	✓	Ethereum	PoW	Hashes of parameters			✓
[70]		✓	✓	✓	Ethereum	PoW	Aggregated models			
[71]		✓		✓	Ethereum	PoW	Encrypted models	✓		✓
[72]		✓			Ethereum	PoW	Aggregated local updates			
[23], [24]	✓	✓	✓	✓	Agnostic	n.a.	Tasks, voting results, model updates		✓	
[73]			✓	✓	EOS, HF	n.s.	Hashes of model updates, data size		✓	
[74]		✓	✓	✓	n.s.	n.s.	Models			
[75]				✓	Agnostic	PBFT	Model updates			
[76]				✓	n.s.	PoW	Tasks, model updates, aggregated models			
[57]				✓	Ethereum	PoW	Reputation scores		✓	
[77]			✓	✓	Custom	PoFL	Model updates			
[78]			✓	✓	Custom	PoMC	Model updates			
[79]				✓	n.s.	n.s.	Signatures, reputation scores, contributions			
[80]		✓	✓	✓	Agnostic	n.s.	Tasks, voting results, model updates			
[81]	✓		✓	✓	Custom	PoW	Model updates, computation time			

the implicit function of storing crucial information and data for the FL process, [22], [55], [71], [74], verifying the correctness of updates [52], [69] or keeping the registry of active members [19], [22], [23], [24], [25], [55] is crucial for the FL workflow and implies coordination through BC as an always accessible, verifiable, transparent, and immutable infrastructure.

3) *Payment:* Many general-purpose BC systems include cryptocurrency capabilities and therefore allow for the incorporation of instant, automatic, and deterministic payment schemes defined by the smart contract's business logic. This advantage was capitalized on by 26 of the 40 FLF we analyzed. Section IV-C discusses the details of applied payment schemes in the context of reward mechanisms and game theory.

4) *Storage:* Decentralized and publicly verifiable storage on the BC facilitates auditability and trust among participants. Even though expansive, since all BC nodes store the same information in a redundant fashion, it might make sense to capitalize on the immutability and transparency feature of BC and store information where either shared access among participants is required or where verifiability of the history is required to hold agents accountable for posterior reward calculations [54]. In particular, ML models [19], [20], [21], [23], [24], [59], [61], [62], [67], [68], [69], reputation scores [54], [57], User-information [19], [22], [23], [24], [25], [55], and Votes [22], [23], [24] are stored on-chain of the respective FLF.

3) RQ 2-3 (Is the Framework Scalable?): Especially if the FLF is intended to be used with hundreds to millions of devices, the scalability of the framework is an important characteristic. In particular: 1) storing large amounts of data such as model parameters and 2) running expansive computations on the BC, e.g., aggregating millions of parameters, calculating expansive contribution measurements, such as Shapley Value or privacy-preserving methods hinder the framework to scaling beyond a small group of entrusted entities toward mass

adoption. To overcome the scalability bottleneck of storage, some FLF applied an interplanetary file system (IPFS) [84], where data is stored off-chain in a distributed file system, using the content address as a unique pointer to each file in a global namespace over computing devices [50], [52], [55], [57], [60], [65], [66], [73]. Other FLFs are based on novel design choices to tackle the scalability issues: Witt et al. [22] applied compressed Federated Knowledge Distillation, storing only 1-bit compressed soft-logits on-chain. Chai et al. [59] designed a hierarchical FLF with two BC layers to reduce the computational overhead by outsourcing computation and storage to an application-specific subchain. Similarly, Feng et al. [63] proposed a two-layered design, where the transaction efficiency of the global chain is improved through sharding. Bao et al. [61] employed an adaption of counting bloom filters (CBFs) to speed up BC queries in the verification step of their FLF. Desai et al. [69] combined public and private BCs, with the former storing reputation scores for accountability and the latter used for heavy computation and storage. Furthermore, the authors apply parameter compression for further scalability improvements.

C. RQ 3: Incentive Mechanisms

1) RQ 3-1 (How Are Incentive Mechanisms Analyzed?): In general, the analysis comprises three steps: The first step is to determine what entities' behavior is examined. In FL, such entities could be workers or task requesters. The second step is to model the entities' utilities or profits. They can be obtained by taking the expectation of possible profits and costs into account. The last step is to analyze the defined utilities or profits. This could be done in a theoretical manner and/or via simulation. 30% (12 out of 40) of the surveyed papers analyzed the IM theoretically, while 45% (18 out of 40 papers) measured workers' rewards via simulation. However, we only focus on the papers with theoretical analysis in this section as we do not see much technical depth or differences in the simulation-based analysis. Our analysis of IMs is summarized in Table V.

In the first step, 28 papers assume that only workers exist, while 12 FLFs have additional entities that pay rewards (e.g., task requesters) [23], [24], [54], [57], [60], [61], [70], [73], [75], [77], [79], [80]. In such a case, the utilities of both entities have to be analyzed such that task requesters can be profitable even if they pay rewards to workers. For instance, workers and task requesters are assumed in [23], and it is vital to determine their behavioral assumptions (e.g., their goals, rationality, etc.).

The second step is to define the utilities or profits of entities. A utility is a 1-D measurable unit that quantifies an entity's value on an outcome and can have positive (e.g., rewards for workers and a value of AI models for task requesters) and negative values (e.g., a computation cost for workers and a total amount of payout for task requesters). Utilities and profits can be derived by subtracting costs from payouts (2). Although the elements of payouts Π are mostly straightforward (e.g., rewards for work contribution), the cost elements C are dependent on the assumed application scenarios. Typical costs in the surveyed papers are computation, electricity (e.g., [21], [23], [24], [59], [60], and [76]), data acquisition (e.g., pictures and sensor readings [74], [75], [76]), and privacy leakage due to model updates (e.g., [75], [76], and [77]). Even multiple cost factors can be considered (e.g., [74], [75], and [76]).

The last step is to analyze the defined utilities and profits to ensure the robustness of the IMs and derive the optimal reward allocation. A simple yet crucial analysis would be to prove that it is worthwhile for workers to join an FL task by showing that their profits are nonnegative. For instance, Bao et al. [61] modeled requesters' and/or workers' profits with given rewards and costs and proved that their profits are nonnegative. Utilities can be used to derive task requesters' and workers' optimal strategies by finding a point where utilities are maximized. By proving the existence of such a point, an equilibrium can be derived, which is a condition where entities (e.g., workers) cannot be better off deviating from their optimal strategies. An equilibrium state, if existent, is proof that a designed mechanism is stable. An equilibrium can be found by deriving the first- and second-order derivatives of the utility function with respect to the variable in question. For instance, Toyoda et al. [24] optimized the workers' data size.

Other works that determine optimal prices for tasks: Wang et al. [76] proposed a Q-learning-based approach to determine the optimal prices so that utilities are maximized via iterative learning processes. Similarly, Zou et al. [74] derived the optimal prices for workers with first- and second-order conditions when the value of data, transmission quality, and communication delay are the factors to determine their competitiveness and costs. Hu et al. propose a two-stage optimization method to determine the optimal values of data and their prices in order by solving an Euler–Lagrange equation of their utilities. These kinds of two-stage optimization games are often formalized as a Stackelberg game [21], [59]. Jian and Wu [21] proposed a Stackelberg-game-based IM for FL. They analyzed the equilibria of two reward policies where a contribution is measured by data size or accuracy by modeling an aggregation server as leader and workers as followers. The uniqueness of their analysis is that they incorporate training and uploading time into the analysis in terms of a constraint as each round has a deadline in the FL.

Chai et al. [59] proposed a multileader multifollower Stackelberg game to analyze their IM in IoV. Aggregation servers (RSUs) are leaders while workers (vehicles) are followers, and aggregation servers first suggest prices and workers determine how much data they should collect and use for training so that both entities' utilities are maximized in order. Due to the high dimensionality of each worker's strategy, it is difficult to employ the traditional backward induction method to derive an equilibrium. Hence, they leverage the alternating direction method of multipliers (ADMMs) algorithm [87] to iteratively reach the social optimum point.

Three papers model the IM in FL as a contract or contest: a task requester proposes a contract with a task description and its reward and workers can determine whether or not to sign such a contract and how many resources they will provide [60]. An FL process can be also seen as a contest as workers need to

TABLE V
OVERVIEW OF IMS AND CONTRIBUTION MEASUREMENT

Ref.	Sim.	Theoretical analysis	Costs	Contribution				Validator
				Metrics	Abs.	Rel.	Rep.	
[22]	✓	✓	Generic	Correlation of predictions (Peer truth serum)		✓		Smart contract
[19]		✓	n.s.	n.s.				Miners
[50]			n.a.	Accuracy (validation scores)	✓			Miners
[51]	✓		n.a.	Accuracy, data size	✓			Agg. server
[52]	✓		n.a.	Euclidean distance of model updates		✓		Miners
[53]	✓		n.a.	Data size	✓			Smart contract
[54]	✓		n.a.	Accuracy, energy consumption, data size		✓	✓	Task requesters
[20]			n.a.	Accuracy (loss)	✓			MEC servers
[25]	✓		n.a.	Accuracy (loss) (Shapley values)		✓		Miners
[55]			n.a.	Accuracy (loss, marginal) (rank)		✓		Validators
[21]	✓	✓(Stackelberg game)	Computation	Accuracy, data size	✓			Agg. servers
[56]			n.a.	n.s.				n.s.
[57]			n.a.	n.s.				n.s.
[58]			n.a.	Data size	✓			Task requesters
[59]	✓	✓(Stackelberg game)	Computation	Accuracy (loss)	✓			Agg. servers
[60]	✓	✓(Contract theory)	Energy	RONI [86], FoolsGold [87]			✓	Task requesters
[61]		✓	Generic	n.s.				Workers
[62]	✓		n.a.	Generic (Shapley values)		✓		Smart contract
[63]			n.a.	n.s.				n.s.
[64]			n.a.	Accuracy (generic, marginal)	✓			n.s.
[65]	✓		n.a.	Accuracy (generic)		✓		Workers
[66]			n.a.	n.s.				Validators
[67]	✓		n.a.	Similarity of model updates (Shapley values)		✓		n.s.
[68]	✓		n.a.	Communication delay, energy consumption	✓			Agg. servers
[69]			n.a.	Speed of model submission	✓			n.s.
[70]			n.a.	n.s.				n.s.
[71]			n.a.	Accuracy, data size	✓		✓	Task requesters
[72]			n.a.	Data size	✓			Agg. servers
[23], [24]		✓(Contest theory)	Computation	Accuracy (generic) (rank by voting)		✓		Workers
[73]			n.a.	Data size	✓			Workers
[74]	✓	✓	Data, communication	Accuracy (loss)	✓			Agg. servers
[75]	✓	✓	Sensing, privacy	n.s.				n.s.
[76]	✓	✓(Reinforcement learning)	Sensing, privacy, energy	Data size, sensing capacity	✓			Agg. servers
[57]			n.a.	n.s.				n.s.
[77]	✓	✓	Privacy	Accuracy	✓			BC nodes
[78]			n.a.	n.s.				n.s.
[79]	✓	✓	n.s.	Accuracy (loss)	✓		✓	Workers
[80]			n.a.	Accuracy (generic) (rank by voting)		✓		Workers
[81]			n.a.	Computation time	✓			Miners

work first, which incurs irreversible costs due to computation, whereas their rewards are not guaranteed at the time of model update submission. Toyoda et al. [23], [24] gave an incentive analysis based on the contest theory. Workers' utilities are used to derive how much effort workers' should exert on a task under the risk of not gaining prizes, while requesters' utility is used to determine how a prize should be split among workers.

2) RQ 3-2 (How Are the Contributions of Clients Measured?): IMs require: 1) the measurement of contribution by each client to 2) fairly distribute rewards. However, the clients' contributions in form of model updates or gradients do not imply direct information on the overall performance metric like the accuracy of the global model.

The metrics used in the literature can be categorized into absolute and relative ones. The absolute metrics are metrics that can be measured without others' local model updates. For instance, a loss function can be measured from a local model and a global model, and the difference between them can be used as a metric for contribution measurement. Although the majority of absolute metrics are based on the accuracy (e.g., [50] and [51]) and data size (e.g., [21] and [53]), other factors are also proposed, such as energy consumption (e.g., [54] and [68]) and computation time [81]. Some combine multiple metrics (e.g., [54] and [71]). Absolute metrics are generally straightforward but hard to validate, e.g., metrics that are based on the data size used depending on the client's honesty. In contrast, relative metrics can be measured by comparing submitted results (e.g., gradients and model updates) in terms of correlation or ranking. For instance, Zhao et al. [52] proposed a metric based on the Euclidean distance of workers' model updates. Likewise, Radanovic et al. [88] utilized peer-truth serum in the FL context, where contributions are measured based on the correlation of prediction on the labels of a public data set [22]. Voting is another approach to determine clients' contribution relatively. For instance, clients choose the best model updates from the previous `FedAvg` round by ranking the respective updates based on the accuracy using their local data sets [23], [24], [80].

A similar metric is clients' reputations. If the same clients are assumed to join different FL tasks, reputation scores calculated based on clients' past contributions can be used

to determine the reward distribution (e.g., [52], [54], [60], and [71]). For instance, Kang et al. [60] proposed to calculate workers' reputation based on a direct opinion by a task requester and indirect opinions by other task requesters. However, even if the clients' individual contribution is measured, the question regarding fair distribution remains an unsolved issue. The Shapley value is an approach to determine payouts to workers based on the marginal utility added, taking all possible combinations of contributors into consideration [89]. Three papers propose to use the Shapley value for fair reward distribution in the FL context [25], [62], [67]. Liu et al. [25] applied the Shapley value based on the accuracy of a test data set. He et al. compared their Shapley-value-based method with three approaches, namely: 1) equal distribution; 2) a method based on individual contribution; and 3) a method called the labor union game where only the order of submission is taken into account to contribution measurement, and found that the Shapley-value-based method outperforms the others in terms of workers' motivation and fairness [67]. Ma et al. [62] proposed a method to calculate Shapley values even if model updates are masked to preserve workers' privacy.

Which entities validate the contribution is an open issue, complementary to the contribution measurement. The validators in FLF can be classified into: 1) aggregation servers FL (e.g., [20], [51], and [59]); 2) task requesters (e.g., [54], [60], and [71]); 3) validators whose task is only to measure contribution [55], [66]; 4) BC nodes (e.g., [50], [77], and [81]); 5) workers (e.g., [65], [79], and [80]); and 6) smart contracts (e.g., [22], [53], and [62]). Some of the works assume that aggregation servers, task requesters, or validators are expected to possess data sets to calculate the metrics discussed above. As reviewed in Section IV-B, others propose custom BC architectures for FL where the validation process is integrated into the consensus mechanism, making BC nodes validators. In some scenarios, aggregation servers, task requesters, and BC nodes take up the role of validators since they aggregate the model updates. However, data sets for validation may not be always available. Furthermore, metrics based on the correlation of predicted labels do not require any validation data set and can even be measured in a smart contract [22].

D. RQ 4: Experiments

Conducting experiments is a key element of FLF development for two reasons. First, the implementation of an example testifies to the feasibility of the approach and gives the authors the chance to identify weaknesses of their frameworks, e.g., poor scalability. Second, conducting experiments allows the comparison of the proposed approaches with each other, e.g., based on the accuracy of the models on standardized test sets. We screened the papers for experiments, and when present, examined them according to nine criteria (Table VI).

1) RQ 4-1 (Is the Performance of the Framework Reported?): The large majority of papers report results of their experiments expressed in either loss, accuracy, or F1 score (84.8%, 28 out of 33 papers with experiments). The remaining instead focus on the performance of their novel group-based Shapley value calculation for contribution measurement [62], the user interface [25], the computational effort and adversarial influence [69], or game-theoretic quantities, such as utility values and rewards [76]. We note that comparability of the approaches is not given through the conducted experiments, since even when using the same data sets and the same evaluation metrics, different experimental scenarios are investigated. In conclusion, to obtain insightful results, experiments should compare the performance (accuracy and computational effort) of an incentivized, decentralized FL system in a standardized challenging environment (non-IID, adversaries) with either the performance of a traditional centralized FL system or the performance of a locally trained model without FL. Ideally, the effectiveness of IM and decentralization efforts are reflected in the FLF performance through a holistic experimental design.

2) RQ 4-2 (How Comprehensive Are the Experiments?): First, it was found that the majority of publications do include experiments. Only seven of 40 papers did not conduct experiments [23], [24], [57], [64], [66], [67]. However, the analysis also shows that only 45% of the experiments implement the actual BC processes (15 out of 33 papers with experiments). Instead, the distributed functionality was simulated or its impact estimated. For instance, Mugunthan et al. [65] focused on the evaluation of the frameworks' contribution scoring procedure by simulating collusion attacks on the FL procedure. The effect of introducing BC to the FLF was accounted for by estimating the per-agent gas consumption. Similarly, Chai et al. [59] conducted experiments specifically designed to investigate the Stackelberg game-based IM. The authors accomplish this without implementing the BC processes.

To test the FL functionality of the framework, an ML problem and a data set must be selected. For the ML application and the data set used, we observe a high homogeneity. Almost all experiments realize classification problems and use publicly available benchmark data sets. The most common are MNIST (handwritten digits) [91] and its variations, as well as CIFAR-10 (objects and animals) [92]. Only Rathore et al. [72] and Li et al. [20] did not performed classification tasks. Rathore et al. [72] performed object detection on the PASCAL VOC 12 data set [93]. Object detection typically combines regression and classification by predicting bounding boxes and labeling them. Li et al. [20] applied their FLF to autonomous driving and minimized the deviations in steering-wheel rotation between a human-driven and simulation-driven vehicle. This corresponds to a regression task.

As to the number of training data holders, the experiments considered between one [20] and 900 [70] clients. In general, one would expect papers specifying CS settings to test with fewer (<100 [8]) and papers specifying CD settings to test with more (>100) clients. Of the frameworks clearly designed for a CD application, it is noticeable that only Kang et al. [60] and Desai et al. [69] conducted experiments with 100 participants or more. On the contrary, Rahmadika and Rhee [56] tested with as many as 100 participants, although only designing a CS framework.

Regarding the FL algorithm, the classic FedAvg [6] is mainly used. Furthermore, in some experiments algorithms

TABLE VI
OVERVIEW OF EXPERIMENTS. (BCWD = BREAST CANCER WISCONSIN DATA SET, BT = BLOCKCHAIN TAMPERING, DGHV = DIJK–GENTRY–HALEVI–VAIKUTANATHAN ALGORITHM, DP = DIFFERENTIAL PRIVACY, ECC = ELLIPTIC CURVE CRYPTOGRAPHY, HE = HOMOMORPHIC ENCRYPTION, HDD = HEART DISEASE DATA SET, KDD = KNOWLEDGE DISCOVERY AND DATA MINING TOOLS COMPETITION, RP = RANDOM MODEL POISONING, RSA = RIVEST–SHAMIR–ADLEMAN CRYPTOSYSTEM, RT = REPUTATION TAMPERING, SA = SECURE AGGREGATION [90], SP = SYSTEMATIC MODEL POISONING, ZKP = ZERO-KNOWLEDGE PROOF, AND 2PC = 2-PARTY COMPUTATION)

Ref.	Tasks		Datasets	#Clients	Algorithms	Privacy	Non-iid	Adversaries				Imp.	Per.
	Clf.	Rgr.						BT	RP	RT	SP		
[22]	✓		EMNIST	10	FD		✓		✓		✓		✓
[19]	✓		MNIST	4-10	FedAvg	HE (Paillier)						✓	✓
[50]	✓		MNIST	5	FedAvg, EWC	DP, HE	✓						✓
[51]	✓		Original	4	FedAvg, CDW	n.s.	✓					✓	✓
[52]	✓		MNIST	10	FedAvg	DP			✓				✓
[53]	✓		MNIST	25	FedAvg	n.s.						✓	✓
[54]	✓		MNIST, CIFAR10	n.s.	n.s.								✓
[20]		✓	Real-time AD video	1	n.s.	HE (DGHV), ZKP							✓
[25]	✓		MNIST	n.s.	FedAvg	SA						✓	
[55]	✓		MNIST	5	FedAvg, FedProx	Sym. cryptography			✓			✓	
[21]	✓		Reddit, Celeba	5-75	n.s.								✓
[56]	✓		MNIST	100	Original	Ring sig., HE (RSA), Rabin, ECC						✓	✓
[57]			n.a.	n.a.	n.a.								
[58]	✓		Mathworks handwritten	10	FedAvg	Pairing-based cryptography, ECC						✓	✓
[59]	✓		MNIST, CIFAR10	6	Original	Asym. cryptography, signatures			✓				✓
[60]	✓		MNIST	100	FedAvg		✓		✓	✓		✓	✓
[61]	✓		n.s.	10	n.s.	n.s.						✓	✓
[62]	✓		ORHD	9	FedAvg	SA	✓						
[63]	✓		MNIST	30	n.s.								✓
[64]			n.a.	n.a.	n.a.								
[65]	✓		Adult census, KDD	50	Custom FedAvg	DP	✓		✓	✓	✓		✓
[66]			n.a.	n.a.	n.a.								
[67]			n.a.	n.a.	n.a.								
[68]	✓		Original	3, 4	FedAvg	n.s.			✓				✓
[69]	✓		CIFAR10	100	FedAvg, signSGD	n.s.	✓				✓	✓	
[70]	✓		FMNIST	900	FedAvg	HE (Paillier)	✓		✓			✓	✓
[71]	✓		BCWD, HDD	3	FedAvg	DP						✓	✓
[72]	✓	✓	PASCAL VOC 2012	5-10	FedAvg	HE						✓	✓
[23], [24]			n.a.	n.a.	n.a.								
[73]	✓		MNIST	10	FedAvg		✓					✓	✓
[74]	✓		MNIST	50	n.s.								✓
[75]	✓		FEMNIST	35, 105, 175	n.s.		✓						✓
[76]	✓		MNIST	n.s.[2]	n.s.								
[57]			n.a.	n.a.	n.a.								
[77]	✓		CIFAR-10	20, 50, 100	n.s.	HE, 2PC							✓
[78]	✓		ImageNet	20	n.s.			✓	✓			✓	✓
[79]	✓		MNIST, CIFAR-10	10	FedAvg				✓		✓		✓
[80]	✓		MNIST	50	FedAvg								✓
[81]	✓		MNIST, CIFAR-10	2-6	FedAvg		✓						✓

are used that mitigate the problem of catastrophic forgetting (elastic weight consolidation (EWC) [50]), reduce the communication overhead (Federated Knowledge Distillation (FD) [22] and signSGD [69]), or show more robust convergence for non-IID and other heterogeneous scenarios (`FedProx` [55], `CDW_FedAvg` [51]). Chai et al. [59] and Mugunthan et al. [65] designed custom FL algorithms. Specifically, Chai et al. [59] proposed an FLF with two aggregation layers in order to promote scalability. In their FL algorithm, nodes in the middle layer aggregate the local model updates of associated nodes in the lowest layer. This semi-global model is then fine-tuned by the middle layer nodes based on data collected by the middle layer nodes themselves. Finally, nodes in the top layer aggregate the fine-tuned models from the middle layer nodes into a global model which is eventually passed back to the lowest layer nodes. All aggregations are weighted by the training data set size. Mugunthan et al. [65] proposed an FLF where all clients evaluate and score the differentially encrypted locally trained models of all other clients. These scores are reported to a smart contract which computes an overall score for each local model. Eventually, each client aggregates the global model from all local models, weighted by the overall score.

3) RQ 4-3 (Are Non-IID Scenarios Simulated?): In real-world applications of FL, the training data is often non-IID between the clients. This affects the performance of the global model and adds an additional layer of complexity with respect to contribution measurement. Hence, how we simulate non-IID scenarios with open data sets is crucial. In 11 publications and for various benchmark data sets, non-IID scenarios were considered. For the example of the MNIST data set, Witt et al. [22] simulated different levels of non-IID scenarios following the Dirichlet distribution as it can easily model the skewness of data distribution by varying a single parameter. Martinez et al. [73] splited the data set in overlapping fractions of various sizes, whereas Kumar et al. [50] divided the data set so that each trainer only possesses data from two of the ten classes. In a less skewed setting, Kumar et al. allocate data from at most four classes to each trainer, with each class being possessed by two devices.

4) RQ 4-4 (Are Additional Privacy Methods Applied?): Even though FL's core objective is to maintain confidentiality

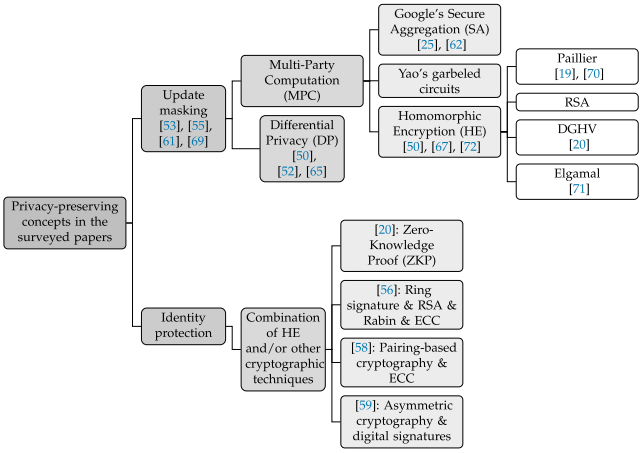

Fig. 3. Privacy-preserving concepts employed in survey papers. (DGHV = Dijk–Gentry–Halevi–Vaikutanathan Algorithm, ECC = Elliptic Curve Cryptography, and RSA = Rivest–Shamir–Adleman Cryptosystem. Papers with unspecified methods are added to next highest node.)

through a privacy-by-design approach where model parameters are aggregated instead of training data, there remain innumerable attack surfaces [94]. Therefore, the presented frameworks employ additional privacy-preserving mechanisms which can be divided into two groups: 1) mechanisms that encrypt or obfuscate gradients and prevent malicious parties to draw conclusions about the data set and 2) mechanisms that hide the identity of participating parties. A classification of the employed privacy-preserving methods can be seen in Figure 3.

The methods of the first group can be further divided into: 1) approaches that are based on cryptographic secure MPC and 2) approaches that are based on differential privacy (DP).

MPC refers to cryptographic methods by which multiple participants can jointly compute a function without having to reveal their respective input values to the other participants. MPC approaches include three groups of methods [90].

1) Google's secure aggregation (SA) [90] is specifically designed to achieve low communication and computation overhead and to be robust toward device dropout. It has been employed by Liu et al. [25] and Ma et al. [62]. Beyond implementing SA, the latter develops a group-based Shapley-value method for contribution measurement, since the native Shapley-value method cannot be applied to masked gradients.
2) Yao's garbled circuits have not been applied to any of the analyzed frameworks, but are mentioned here for completeness.
3) While Yao's garbled circuits were developed for 2-party secure computing, homomorphic encryption (HE) allows for higher numbers of participants [90]. As shown in Figure 3, HE has been employed in several works [19], [20], [67], [70], [71], [72]. For that, different implementations of the homomorphic idea have been chosen, such as the Pallier cryptosystem, the Elgamal cryptosystem, or the Dijk–Gentry–Halevu–Vaikutanathan Algorithm (DGHV). Li et al. [71] chose the Elgamal cryptosystem that is less computationally expensive than other HE approaches.

DP refers to a method where noise is drawn from a probability density function $p_{\text{noise}}(x)$ with expected value $\mathbb{E}(p_{\text{noise}}(x)) = 0$ obfuscates the individual contribution with minimal distortion of the aggregation. DP has been employed by Mugunthan et al. [65], Zhao et al. [52], and Kumar et al. [50], with the latter combining the use of HE and DP. However, the fewer clients participate in the DP process, the heavier the distortion of the aggregated model, introducing a tradeoff between privacy and model accuracy. Zhao et al. [52] mitigated the loss in accuracy by incorporating a novel normalization technique into their neural networks instead of using traditional batch normalization (e.g., [50] and [65]). Besides MPC and DP, another technique for data set protection is chosen by Qu et al. [77]. Instead of the clients sharing masked gradients, the FLF relies on requesters sharing masked data sets in the model verification step. This prevents other workers from copying the models while testing and evaluating them. HE and 2-party computation (2PC) are used. Zhang et al. [55], Desai et al. [69], Bao et al. [61], and Rahmadika and Rhee [53] also relied on the masking of gradients but do not specify the privacy-preserving mechanisms.

The second group of frameworks targets the protection of participants' identities through cryptographic mechanisms. For that, Rahmadika and Rhee [56] combined ring signatures, HE (RSA), Rabin algorithm, and elliptic curve cryptography (ECC), while Chai et al. [59] incorporated digital signatures and asymmetric cryptography approaches, and Rahmadika et al. [58] performed authentication tasks through pairing-based cryptography and ECC. Only one framework implements measures for both masking gradients as well as hiding identities. Li et al. [20] used DGHV for masking gradients and zero-knowledge proof (ZKP) for identity protection.

Finally, He et al. [67] specifically addressed the problem of aligning entities. This problem occurs in vertical FL where different parties hold complementary information about the same user. The parties have to find a way of matching this information without disclosing the identity of their users. To solve this problem, He et al. [67] employed Encrypted Entity Alignment which is a protocol for privacy-preserving interdatabase operations.

5) RQ 4-5 (Is the Framework Robust Against Malicious Participants?): The experiments consider and simulate different types of adversaries, whereas some publications consider multiple types of attacks. Four groups of attack patterns were identified in the publications: 1) random model poisoning; 2) systematic model poisoning; 3) reputation tampering (RT); and 4) BC tampering. The most common attack considered in the experiments is random model poisoning. This includes attacks, where local models are trained on a randomly manipulated data set [55], [60], [65], [79] or where random parameter updates are reported [22], [52], [59], [60], [68], [70], [78]. For instance, malicious agents in [55] use a training data set with intentionally shuffled labels, whereas in [70] the parameter updates are randomly perturbed with the Gaussian noise. Kang et al. [60] analyzed the effects of a bad or manipulated data set by providing 8% of the workers with training data where only a few classes are present, and another 2%

of the workers with mislabeled data. Kang et al. quantify the insufficiency of the data set using the Earth mover's distance.

The second most commonly simulated type of attack is systematic model poisoning where the attackers manipulate the model through well-planned misbehavior. In [69], a fraction of workers collude and manipulates their image classification data sets by introducing a so-called trojan pattern: the malicious agent introduces a white cross to a certain fraction of a class, e.g., to 50% of all dog pictures in an animal classification task and relabels these data points as horse pictures. This creates a backdoor in the model that cannot be detected by subjecting the model to dog or horse pictures which will be correctly classified. However, pictures with the trojan pattern will be misclassified. Other forms of systematic model poisoning can be found with Witt et al. [22], Mugunthan et al. [65], and Gao et al. [79].

The third type of attack that was simulated is RT. Here, malicious agents intentionally provide colluding agents with perfect reputation or voting scores [60], [65]. The fourth type of attack is BC tampering [78]. Here, malicious miners intentionally fork the BC and prevail by building a longer branch faster than the honest miners.

E. RQ 5: Summary: What Are Lessons Learned?

The inherent complexity of FLF leads to heterogeneity of the scientific research across the dimensions: 1) application; 2) overall design; 3) special focus on open issues; and 4) details and thoroughness.

Application: Although the majority of analyzed works offer application independent frameworks (classified as "generic" in Table III) other FLFs are applied across IoT, Industrial-IoT (IIoT), IoV, and Finance. The heterogeneity of the required properties across those domains causes differences in the design choices of function, operations, storage of BC, contribution measurement, and privacy requirements.

Variety of Possible Design Choices: In addition to the domain-specific influence on the system architecture, design choices about the FL algorithms, communication protocol, applications of BC within the ecosystem, BC technology (existing or novel), storage and operation on BC, security tradeoffs, MD, contribution measurement, etc., add to the complexity and overall variety of such systems. For example, some works apply BC as the outer complementary layer [54] while BC is the core infrastructure for coordination, storage, aggregation, and payment in other FLFs [22], [55]. Furthermore, some works developed application-specific BC systems, while others tried to embed an FLF on top of existing BC frameworks, such as Ethereum for cheaper and pragmatic deployment. Our survey exposes a similar variety in the choice of the contribution measurement. The spectrum reaches from the computationally lightweight correlation of answers on a public data set [22] as a proxy for contribution as opposed to the Shapley value, a measurement with strong theoretical properties but massive computational overhead [25], [67].

Special Focus: The aforementioned complexity as well as its novelty results in many open issues across a broad spectrum. Many works, therefore, focus on solving specific issues, such as enhanced privacy [19], [20], [56], [66], novel BC systems [25], [51], bandwidth reduction [68], novel contribution measurements [22], [25], [68], or game theory (e.g., [21] and [60]), as the major contribution which further complicates a holistic comparison of FLF.

Thoroughness: The analyzed papers also vary heavily in provided detail and thoroughness, ranging from first concepts, lacking details in terms of important specifications, such as performance, specific function, operation and storage on BC, contribution measurement, robustness, experiments, and privacy to theoretically detailed and experimentally tested solutions. None of the analyzed papers are production ready.

Standards for Better Comparability: For better reproducibility, implementability, and comparability we suggest considering and defining the following elements when designing an FLF.

System Model and Architecture: With respect to the system model and system architecture, the authors should specify the following aspects.

1) Assumed application.
2) Type of FL (i.e., CD versus CS and horizontal versus vertical).
3) Entities (including attackers).
4) Setup (e.g., who manages a system and who deploys it).
5) Role of BC within the FLF (e.g., what part does BC replace and what functions/operations).
6) BC design (e.g., consensus algorithms, BCs, and smart contracts).
7) Non-BC design (e.g., off-chain storage, privacy protection, and authentication).
8) Procedures (e.g., flowcharts and diagrams).
9) Theoretical analysis of IMs.
10) Specification of clients' contribution measurement.
11) Possible attacks (e.g., system security and data privacy).

Performance Analysis: The performance analysis should include the following subanalyses.

1) Quantitative performance analysis.
2) Scalability analysis with respect to blockchain and contribution measurement.

Cost Analysis: The cost analysis should consider the following items.

1) Overhead and cost analysis of BC infrastructure.
2) Overhead and cost analysis of the contribution measurement.
3) Performance-cost tradeoff discussion.

V. Future Research Directions

The multitude of possible applications of FLF come with different requirements in terms of accuracy, latency, cost, and privacy. To account for this, we classify future research into two main directions, namely: 1) increase in framework performance and 2) expansion of framework functionalities.

A. Performance

Most state-of-the-art publications only consider BC and IMs on a conceptual or theoretical level, however, they lack a performance analysis. Yet, low operational costs and latency,

as required by real-time applications, such as autonomous driving and demand high-performance systems. To develop such frameworks, we have identified four performance bottlenecks as future research directions: 1) framework scalability; 2) communications and network; 3) framework implementation; and 4) framework evaluation and comparison.

1) Framework Scalability: One of the major factors for the applicability of an FLF is its ability to scale beyond small groups toward mass adoption. Out of the 40 papers, only six mentioned and considered scalability within the design of their respective FLF. In particular, our reviews show that the integration of the distributed ledger technology frequently leads to scalability problems. In FLFs, the BC technology becomes a scalability bottleneck if:

1) it is part of the operating core infrastructure of the FLF (e.g., [22]) and not only a complementary outer layer technology (e.g., [54]);
2) heavy operations, such as aggregation or reward calculation are performed on-chain [25];
3) a large amount of information is stored on the BC, such as model updates;
4) the BC framework is public and used outside the realm of the FLF;
5) the consensus mechanism is resource-intense (e.g., PoW).

There are multiple promising future strategies to improve the scalability of the framework. First, FLF-specific BC systems have been proposed that replace the computational overhead of the PoW-based systems with computational heavy tasks in FLF, such as model parameter verification [77], reputation verification [54], or contribution measurement calculations [25]. Second, Zhang et al. [78] have investigated the use of efficient AI hardware to increase BC scalability. Wang et al. [95] explored the domain of resource optimization in BC-based FLFs to further improve the scalability. Moreover, Weng et al. [19] aimed to improve scalability by enhancing the privacy procedures for the FLF processes. Another promising research direction is the application of zero-knowledge succinct noninteractive argument of knowledge (ZK-SNARKs) [96] in the FLF context. ZK-SNARKs is a promising cryptographic technology that allows a *prover* to prove to a *verifier* that computation has been executed without revealing the program itself. This verification is faster than actually computing the original code and can be implemented easily on the smart contract. Hence, this will improve the scalability of BC-enabled FLF dramatically. However, due to its generality, which processes leverage ZK-SNRAKs is an open question. Finally, the performance of the BC itself can be improved, e.g., by increasing the number of transactions per second.

2) Communication and Network: Another major remaining challenge is the communication bottleneck. Decentralized wireless gadgets, as employed in decentralized FL, operate on lower communication rates than traditional intra- or inter-datacenter links. This leads to a tradeoff between accuracy and communication cost. Although there exists first theoretical research on the nature of this tradeoff, its findings have not yet been incorporated in the proposed FLFs [7]. In terms of communication rates, new developments are also expected once the 6G technology is introduced, which is predicted to be mutually empowering with FL [82].

Furthermore, researchers face the communication-related problem of scheduling and resource allocation under dynamic channel condition and heterogeneous computing capacity of devices in IoT [97]. For instance, Yang et al. [97] proposed a device selection strategy in UAV to keep the low-quality devices from affecting the learning efficiency and accuracy.

Another challenge is related to key collisions during update communication: to avoid throughput issues, data is typically uploaded iteratively in multiple smaller batches, causing latency, and collision effects to become more dominant. For instance, Desai et al. [69] pointed out that Hyperledger cannot deal with the multiversion concurrency control (MVCC) of its underlying database so many transactions fail and need to be repeated. Accordingly, future research should be directed to the three compression objectives identified by Kairouz et al. [7]: 1) gradient compression (client-to-aggregator communication); 2) model compression (aggregator-to-client communication); and 3) local computation reduction. In consequence, security and privacy mechanisms need to be adapted to operate on the compressed data (Section IV-D4). Starting points for this upcoming research include sparsification and quantization approaches [98], 1-bit compression [22], or the parallelization on multiple contracts [59], [63], [69]. For the latter, Desai et al. [69] analyzed the tradeoff between communication speed and the number of employed parallelized contracts.

In general, future framework proposals should consider communication cost and time in their simulative methods, e.g., building up on Kang et al. [60].

3) Framework Implementation: Most of the papers we reviewed are focused on the algorithm side. However, in order to go beyond theory toward real production-ready deployments, implementation details have to be taken into consideration. For instance, incurred deployment and maintenance costs of unproven novel BC systems are often ignored. Introducing a new, custom-made, and highly complex infrastructure introduces security risks and it requires a large team of experts to run and maintain such a system in practice. Therefore, software/hardware co-design is another vital topic in FL (e.g., [78], [99], [100], and [101]). For instance, Wang et al. [102] pointed out that cipher-text operation and encryption parts are major bottlenecks on the FL and proposed a novel field programmable gate array (FPGA) design for it. We believe that there are potential research topics in the software/hardware co-design for FL. Interested readers may refer to the survey papers of Khan et al. [99], [100].

4) Framework Evaluation and Comparison: While many papers have conducted performance evaluation, few showed a comparison with other FLFs. This hinders the scientific advancement toward high-performing frameworks as the different design choices of the papers remain uncompared. Furthermore, the frameworks have often not been evaluated in realistic scenarios: 1) relatively well-known benchmark data sets, such as MNIST and CIFAR-10 are chosen (29 out of 34 papers that conducted experiments on classification)

and 2) the non-IID setting is only applied in 11 out of 34 papers. Furthermore, inconsistencies between the targeted FL setting (i.e., CS and CD) and the number of clients in the experiments are observed. In particular, FLFs that assume CD should simulate a large number of clients, however, only Kang et al. [60] and Desai et al. [69] conducted experiments with 100 participants or more (Table VI).

To better evaluate FLFs, we suggest using common data sets dedicated to FL (e.g., LEAF [103]) as well as simulating different levels of non-IID data among clients (e.g., the Dirichlet distribution [22]). We also suggest deploying an FLF on the clusters of inexpensive computers such as Raspberry Pi [104] to realistically simulate large-scale FL scenarios under the CD assumption.

Moreover, it is difficult to simulate the effect of decentralization and incentivization (e.g., Shapley value and game-theoretic mechanisms) in a comparable way since each paper uses different assumptions. Therefore, to fairly compare FLFs, holistic experiments should be designed, where the effects of decentralization and incentivization are captured by metrics, such as overall accuracy, cost, or latency.

B. Functionalities

Future research should also focus on integrating further functionalities into the FLFs. First, most of the proposed FL systems are limited to supervised classification, however, other types of ML problems should be considered as well. Second, lightweight privacy-preserving techniques are necessary for some applications that use sensitive information (e.g., medical logs and personal financial information). Third, a fair, nonmanipulable, and lightweight mechanism for contribution measurement has yet to be developed.

1) Beyond Supervised FL and Federated Averaging: To expand the applicability of FLFs, ML tasks beyond supervised learning should be enabled, such as anomaly detection, reinforcement learning, natural language processing, user behavior analysis, and unsupervised learning tasks (e.g., [105] and [106]). This will require new or adapted model aggregation algorithms and a new contribution measurement to integrate such tasks with IM and BC.

So far, `FedAvg` requires the same neural network architecture on all devices to participate. This may lead to issues in real-world environments where clients might have different hardware and bandwidth capabilities. Federated Knowledge Distillation [22] is an interesting novel FL approach in this context, allowing for a flexible neural network architecture and a dramatic reduction in bandwidth [107]. However, a Federated Knowledge Distillation requires a public data set to distill the knowledge.

Traditional deep learning algorithms, such as DNNs and convolutional neural networks (CNNs), are generally power-hungry, which is problematic in the IoT environment. To address this challenge, biological neurons-inspired DNNs called spiking neural networks (SNNs) have been actively studied for edge AI (e.g., [108] and [109]). SNNs will enable edge devices to exploit brain-like biophysiological structure to collaboratively train a global model while helping preserve privacy. For instance, lead federated neuromorphic learning (LFNL) is a method to enable SNNs in a federated manner [109]. Furthermore, a leader election scheme is proposed to elect one device with high capability (e.g., computation and communication capabilities) as a leader to manage model aggregation, eliminating a fixed central coordinator and avoiding model poisoning attacks.

2) Toward Lightweight Privacy-Preserving FL: Despite FL being a data privacy-preserving technology by design, research has shown that certain characteristics of the underlying training data sets can be inferred from the global model and that additional privacy-preserving measures are recommended. Our review shows that two classes of security concerns are targeted by the publications, namely: 1) leakage of data set characteristics and 2) disclosure of participant identities. Although a substantial number of papers (20 out of 40 publications) address one of these concerns, only a single paper addresses both [20]. Moreover, preventing data set leakage through DP or MPC inflicts tradeoffs. Specifically, DP comes with a tradeoff between data security and model accuracy, while MPC comes with a tradeoff between data security and computation complexity, and it might thus not be applicable with a large number of participants [62]. It is worth noting that the model accuracy of DP cannot be inherently improved due to intentionally added noise. Hence, it would be important to explore lightweight MPC algorithms [90] to accommodate a large number of clients for privacy-preserving FL.

3) Toward Fair, Nonmanipulable, and Lightweight Contribution Measurement: Although multiple approaches for contribution measurements have been explored in the literature (Section IV-C), a fair, nonmanipulable, and lightweight mechanism has yet to be developed as the following overview shows.

First, a contribution can be measured based on the clients' honest reports of the amount of data, local accuracy, or local loss. Yet reward systems based on such simplified assumptions may not be applicable in any real-world scenario as the dominant strategy for an individual-rational agent is dishonest behavior (e.g., reporting the best possible outcome without costly model training). Recent technologies, such as TEE and ZK-SNARKs are promising for trusted computation on mobile, edge, and IoT devices [110]. However, how to leverage them to achieve honest reports without incurring additional costs (e.g., computational costs) is an open question.

Second, relative contribution measurement based on the client's reputation or majority voting is an interesting research avenue, promising to relax heavy verification, and control mechanics for high-reputation clients. However, how to quantify the reputation fairly and robustly remains open research. Similarly, the majority voting methods may not reflect actual contribution due to its nature.

Third, absolute or direct contribution measurement refers to assessing each client's model update on a public data set. However, this approach 1) requires a trusted central authority performing tests and 2) limits scalability due to the computational overhead. For instance, the Shapley value is a common method for measuring an agent's contribution, but

still comes at the cost of heavy computational overhead even when optimized (e.g., [111] and [112]).

Lately, correlation-based reward mechanisms, such as correlated agreement (CA) [113], [114] and peer-truth serum [22], have been proposed as promising approaches for contribution measurements for FL. Without having access to the ground truth, the reward is calculated based on the correlation of the reported signals of peers. This implicit approach does not require an explicit contribution measurement and therefore avoids computational overhead.

In addition, when theoretically developing and analyzing IMs, as performed by 12 out of 40 papers, more sophisticated assumptions concerning: 1) information availability; 2) uniformity in utility functions; or 3) IR should be made to guarantee the robustness of the mechanisms in a real-world scenario. Specifically, as clients are humans, they may not follow their optimal strategies derived from the analysis. For instance, not all clients would take the cost of energy consumption into account when determining their strategies. We suggest taking humans' behavioral bias (e.g., prospect theory [115], [116]) as well as nonquantifiable measures (e.g., the utility of privacy) into the theoretical analysis of IMs.

VI. Conclusion and Outlook

FL is a promising new AI paradigm focused on confidential and parallel model training on the edge. To apply FL beyond small groups of entrusted entities, a decentralization of power, as well as compensation for participating clients, has to be incorporated into the FLF. This work traversed and analyzed 12 leading scientific databases for incentivized and decentralized FLFs based on the PRISMA methodology, ensuring transparency, and reproducibility. We found 422 papers and studied 40 works in-depth after three filtering rounds. To ensure correctness, the results were verified by the respective authors. We overcame the challenge of heterogeneity of FLFs in terms of use cases, applied focus, design choice, and thoroughness by offering a comprehensive and holistic comparison framework. By exposing the limitations of existing FLFs and providing directions for future research, this work aims to enhance the proliferation of incentivized and decentralized FL in practice.

References

[1] J. Clement, *Google, Amazon, Facebook, Apple, and Microsoft (GAFAM)—Statistics & Facts*, Statista, New York, NY, USA, 2021.

[2] S. Wheatley, T. Maillart, and D. Sornette, "The extreme risk of personal data breaches and the erosion of privacy," *Eur. Phys. J. B*, vol. 89, no. 1, p. 7, 2016.

[3] T. Mirrlees. "Getting at GAFAM's power: A structural and relational framework." Feb. 2021. Accessed: Oct. 18, 2022. [Online]. Available: https://www.researchgate.net/publication/350442622_Getting_at_GAFAM's_Power_A_Structural_and_Relational_Framework/citations

[4] C. Santesteban and S. Longpre, "How big data confers market power to big tech: Leveraging the perspective of data science," *Antitrust Bull.*, vol. 65, no. 3, pp. 459–485, 2020.

[5] "2018 reform of EU data protection rules." European Commission. 2018. [Online]. Available: https://ec.europa.eu/commission/sites/beta-political/files/data-protection-factsheet-changes_en.pdf

[6] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. AISTATS*, 2017, pp. 1–10.

[7] P. Kairouz et al., "Advances and open problems in federated learning," 2019, *arXiv:1912.04977*.

[8] R. Zeng, C. Zeng, X. Wang, B. Li, and X. Chu, "A comprehensive survey of incentive mechanism for federated learning," 2021, *arXiv:2106.15406*.

[9] J. Hamer, M. Mohri, and A. T. Suresh, "FedBoost: A communication-efficient algorithm for federated learning," in *Proc. ICML*, vol. 119, 2020, pp. 3973–3983.

[10] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Int. Conf. Mach. Learn. Syst.*, vol. 2, 2020, pp. 429–450.

[11] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," in *Proc. NeurIPS*, vol. 33, 2020, pp. 7611–7623.

[12] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-i.i.d. data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020.

[13] D. Rothchild et al., "FetchSGD: Communication-efficient federated learning with sketching," 2020, *arXiv:2007.07682*.

[14] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, Decentralized Bus. Rev., Seoul, South Korea, 2008, Art. no. 21260.

[15] S. S. Shetty, C. A. Kamhoua, and L. L. Njilla, "Distributed consensus protocols and algorithms," in *Blockchain for Distributed Systems Security*. Hoboken, NJ, USA: Wiley, 2019, pp. 25–50.

[16] G. Wood et al., "Ethereum: A secure decentralised generalised transaction ledger," Ethereum Project, Zug, Switzerland, Yellow Paper, pp. 1–32, 2014.

[17] "Ethereum networking layer." Accessed: Oct. 18, 2022. [Online]. Available: https://ethereum.org/en/developers/docs/networking-layer/

[18] "Ethereum proof-of-stake." Accessed: Oct. 18, 2022. [Online]. Available: https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/

[19] J. Weng, J. Weng, J. Zhang, M. Li, Y. Zhang, and W. Luo, "DeepChain: Auditable and privacy-preserving deep learning with blockchain-based incentive," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 5, pp. 2438–2455, Sep./Oct. 2021.

[20] Y. Li, X. Tao, X. Zhang, J. Liu, and J. Xu, "Privacy-preserved federated learning for autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 8423–8434, Jul. 2022.

[21] S. Jiang and J. Wu, "A reward response game in the blockchain-powered federated learning system," *Int. J. Parallel Emergent Distrib. Syst.*, vol. 37, no. 1, pp. 68–90, 2022.

[22] L. Witt, U. Zafar, K. Shen, F. Sattler, D. Li, and W. Samek, "Reward-based 1-bit compressed federated distillation on blockchain," 2021, *arXiv:2106.14265*.

[23] K. Toyoda and A. N. Zhang, "Mechanism design for an incentive-aware blockchain-enabled federated learning platform," in *Proc. Int. Conf. Big Data*, 2019, pp. 395–403.

[24] K. Toyoda, J. Zhao, A. N. S. Zhang, and P. T. Mathiopoulos, "Blockchain-enabled federated learning with mechanism design," *IEEE Access*, vol. 8, pp. 219744–219756, 2020.

[25] Y. Liu, Z. Ai, S. Sun, S. Zhang, Z. Liu, and H. Yu, *FedCoin: A Peer-to-Peer Payment System for Federated Learning*. Cham, Switzerland: Springer, 2020, pp. 125–138.

[26] "Matic whitepaper." Accessed: Oct. 18, 2022. [Online]. Available: https://github.com/maticnetwork/whitepaper

[27] "Binance chain whitepaper." Accessed: Oct. 18, 2022. [Online]. Available: https://github.com/bnb-chain/whitepaper/blob/master/WHITEPAPER.md

[28] "Avalanche Whitepaper." Accessed: Oct. 18, 2022. [Online]. Available: https://assets.website-files.com/5d80307810123f5ffbb34d6e/6008d7bbf8b10d1eb01e7e16_Avalanc%20Platfo%20Whitepaper.pdf

[29] D. G. Wood. "Polkadot: Vision for a heterogeneous multi-chain framework." Accessed: Oct. 18, 2022. [Online]. Available: https://polkadot.network/PolkaDotPaper.pdf

[30] E. Androulaki et al., "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proc. EuroSys Conf.*, 2018, pp. 1–15.

[31] N. Nisan et al., "Introduction to mechanism design (for computer scientists)," in *Algorithmic Game Theory*, vol. 9. Cambridge, U.K.: Cambridge Univ. Press, 2007, pp. 209–242.

[32] S. Chakrabarti, T. Knauth, D. Kuvaiskii, M. Steiner, and M. Vij, "Trusted execution environment with intel SGX," in *Proc. Responsible Genomic Data Sharing*, 2020, pp. 161–190.

[33] X. Tu, K. Zhu, N. C. Luong, D. Niyato, Y. Zhang, and J. Li, "Incentive mechanisms for federated learning: From economic and game theoretic perspective," *IEEE Trans. Cogn. Commun. Netw.*, vol. 8, no. 3, pp. 1566–1593, Sep. 2022.

[34] H. V. Stackelberg et al., *Theory of the Market Economy*. Oxford, U.K.: Oxford Univ. Press, 1952.

[35] L. U. Khan et al., "Federated learning for edge networks: Resource optimization and incentive mechanism," 2019, *arXiv:1911.05642*.

[36] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, "A learning-based incentive mechanism for federated learning," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6360–6368, Jul. 2020.

[37] M. Vojnović, "Contest theory," *Commun. ACM*, vol. 60, no. 5, pp. 70–80, Apr. 2017.

[38] P. Bolton and M. Dewatripont, *Contract Theory*. Cambridge, MA, USA: MIT Press, 2004.

[39] V. Krishna, *Auction Theory*. Burlington, MA, USA: Academic, 2009.

[40] G. Tullock, "Efficient rent seeking," in *Efficient Rent-Seeking*. Boston, MA, USA: Springer, 2001, pp. 3–16.

[41] H. Yu et al., "A fairness-aware incentive scheme for federated learning," in *Proc. AAAI/ACM Conf. AI Ethics Soc. (AIES)*, Feb. 2020, pp. 393–399.

[42] Y. Zhan, J. Zhang, Z. Hong, L. Wu, P. Li, and S. Guo, "A survey of incentive mechanism design for federated learning," *IEEE Trans. Emerg. Topics Comput.*, vol. 10, no. 2, pp. 1035–1044, Apr–Jun. 2022.

[43] D. Hou, J. Zhang, K. L. Man, J. Ma, and Z. Peng, "A systematic literature review of blockchain-based federated learning: Architectures, applications and issues," in *Proc. ICTC*, 2021, pp. 302–307.

[44] A. Ali, I. Ilahi, A. Qayyum, I. Mohammed, A. Al-Fuqaha, and J. Qadir. "Incentive-driven federated learning and associated security challenges: A systematic review." TechRxiv. Jul. 2021. [Online]. Available: https://www.techrxiv.org/articles/preprint/Incentive-Driven_Federated_Learning_and_Associated_Security_Challenges_A_Systematic_Review/14945433

[45] D. C. Nguyen et al., "Federated learning meets blockchain in edge computing: Opportunities and challenges," *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12806–12825, Aug. 2021.

[46] Z. Wang and Q. Hu, "Blockchain-based federated learning: A comprehensive survey," 2021, *arXiv:2110.02182*.

[47] D. Moher et al., "Preferred reporting items for systematic review and meta-analysis protocols (PRISMA-P) 2015 statement," *Syst. Rev.*, vol. 4, no. 1, p. 1, 2015.

[48] C. Okoli and K. Schabram, "A guide to conducting a systematic literature review of information systems research," *SSRN Electron. J.*, vol. 10, p. 2, Dec. 2015.

[49] B. Kitchenham, *Procedures for Performing Systematic Reviews*, vol. 33, Keele Univ., Keele, U.K., 2004, pp. 1–26.

[50] S. Kumar, S. Dutta, S. Chatturvedi, and M. P. S. Bhatia, "Strategies for enhancing training and privacy in blockchain enabled federated learning," in *Proc. BigMM*, 2020, pp. 333–340.

[51] W. Zhang et al., "Blockchain-based federated learning for device failure detection in industrial IoT," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5926–5937, Apr. 2021.

[52] Y. Zhao et al., "Privacy-preserving blockchain-based federated learning for IoT devices," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1817–1829, Feb. 2021.

[53] S. Rahmadika and K.-H. Rhee, "Reliable collaborative learning with commensurate incentive schemes," in *Proc. IEEE Int. Conf. Blockchain*, 2020, pp. 496–502.

[54] Q. Zhang, Q. Ding, J. Zhu, and D. Li, "Blockchain empowered reliable federated learning by worker selection: A trustworthy reputation evaluation method," in *Proc. WCNCW*, 2021, pp. 1–6.

[55] Z. Zhang et al., "Refiner: A reliable incentive-driven federated learning system powered by blockchain," *VLDB Endowm.*, vol. 14, no. 12, pp. 2659–2662, 2021.

[56] S. Rahmadika and K.-H. Rhee, "Unlinkable collaborative learning transactions: Privacy-awareness in decentralized approaches," *IEEE Access*, vol. 9, pp. 65293–65307, 2021.

[57] M. H. ur Rehman, K. Salah, E. Damiani, and D. Svetinovic, "Towards blockchain-based reputation-aware federated learning," in *Proc. INFOCOMW*, 2020, pp. 183–188.

[58] S. Rahmadika, M. Firdaus, S. Jang, and K.-H. Rhee, "Blockchain-enabled 5G edge networks and beyond: An intelligent cross-silo federated learning approach," *Security Commun. Netw.*, vol. 2021, Mar. 2021, Art. no. 5550153.

[59] H. Chai, S. Leng, Y. Chen, and K. Zhang, "A hierarchical blockchain-enabled federated learning algorithm for knowledge sharing in Internet of Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 3975–3986, Jul. 2021.

[60] 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.

[61] X. Bao, C. Su, Y. Xiong, W. Huang, and Y. Hu, "FLChain: A blockchain for auditable federated learning with trust and incentive," in *Proc. BIGCOM*, Aug. 2019, pp. 151–159.

[62] S. Ma, Y. Cao, and L. Xiong, "Transparent contribution evaluation for secure federated learning on blockchain," in *Proc. ICDEW*, 2021, pp. 88–91.

[63] L. Feng, Z. Yang, S. Guo, X. Qiu, W. Li, and P. Yu, "Two-layered blockchain architecture for federated learning over the mobile edge network," *IEEE Netw.*, vol. 36, no. 1, pp. 45–51, Jan./Feb. 2022.

[64] B. Kansra, H. Diddee, T. H. Sheikh, A. Khanna, D. Gupta, and J. J. P. C. Rodrigues, "BlockFITS: A federated data augmentation modelling for blockchain-based IoVT systems," in *Proc. ICICC*, 2022, pp. 253–262.

[65] V. Mugunthan, R. Rahman, and L. Kagal, "BlockFLow: Decentralized, privacy-preserving, and accountable federated machine learning," in *Proc. BLOCKCHAIN*, 2022, pp. 233–242.

[66] A. Fadaeddini, B. Majidi, and M. Eshghi, "Privacy preserved Decentralized deep learning: A blockchain based solution for secure AI-driven Enterprise," in *Proc. High Perform. Comput. Big Data Anal.*, 2019, pp. 32–40.

[67] C. He, B. Xiao, X. Chen, Q. Xu, and J. Lin, "Federated learning intellectual capital platform," *Pers. Ubiquitous Comput.*, to be published.

[68] G. Qu, H. Wu, and N. Cui, "Joint blockchain and federated learning-based offloading in harsh edge computing environments," in *Proc. Int. Workshop Big Data Emergent Distrib. Environ.*, 2021, pp. 1–6.

[69] H. B. Desai, M. S. Ozdayi, and M. Kantarcioglu, "BlockFLA: Accountable federated learning via hybrid blockchain architecture," in *Proc. 11th ACM Conf. Data Appl. Security Privacy*, 2021, pp. 101–112.

[70] X. Zhu and H. Li, "Privacy-preserving decentralized federated deep learning," in *Proc. ACM Turing Award Celeb. Conf. China*, 2021, pp. 33–38.

[71] Z. Li, J. Liu, J. Hao, H. Wang, and M. Xian, "CrowdSFL: A secure crowd computing framework based on blockchain and federated learning," *Electronics*, vol. 9, no. 5, p. 773, 2020.

[72] S. Rathore, Y. Pan, and J. H. Park, "BlockDeepNet: A blockchain-based secure deep learning for IoT network," *Sustainability*, vol. 11, no. 14, p. 3974, 2019.

[73] I. Martinez, S. Francis, and A. S. Hafid, "Record and reward federated learning contributions with blockchain," in *Proc. Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discovery. (CyberC)*, Oct. 2019, pp. 50–57.

[74] Y. Zou, F. Shen, F. Yan, J. Lin, and Y. Qiu, "Reputation-based regional federated learning for knowledge trading in blockchain-enhanced IoV," in *Proc. IEEE WCNC*, Mar. 2021, pp. 1–6.

[75] Q. Hu, Z. Wang, M. Xu, and X. Cheng, "Blockchain and federated edge learning for privacy-preserving mobile crowdsensing," *IEEE Internet Things J.*, early access, Nov. 16, 2021, doi: 10.1109/JIOT.2021.3128155.

[76] Y. Wang, Z. Su, N. Zhang, and A. Benslimane, "Learning in the air: Secure federated learning for UAV-assisted crowdsensing," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1055–1069, Apr.–Jun. 2021.

[77] X. Qu, S. Wang, Q. Hu, and X. Cheng, "Proof of federated learning: A novel energy-recycling consensus algorithm," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 8, pp. 2074–2085, Aug. 2021.

[78] R. Zhang et al., "Democratic learning: Hardware/software co-design for lightweight blockchain-secured on-device machine learning," *Int. J. High Perform. Syst. Archit.*, vol. 118, Sep. 2021, Art. no. 102205.

[79] L. Gao, L. Li, Y. Chen, W. Zheng, C. Xu, and M. Xu, "FIFL: A fair incentive mechanism for federated learning," in *Proc. ICPP*, 2021, pp. 1–10.

[80] S. Xuan, M. Jin, X. Li, Z. Yao, W. Yang, and D. Man, "DAM-SE: A blockchain-based optimized solution for the counterattacks in the Internet of federated learning systems," *Security Commun. Netw.*, vol. 2021, Jul. 2021, Art. no. 9965157.

[81] Y. Liu, Y. Qu, C. Xu, Z. Hao, and B. Gu, "Blockchain-enabled asynchronous federated learning in edge computing," *Sensors*, vol. 21, no. 10, p. 3335, May 2021.
[82] Y. Liu, X. Yuan, Z. Xiong, J. Kang, X. Wang, and D. Niyato, "Federated learning for 6G communications: Challenges, methods, and future directions," *China Commun.*, vol. 17, no. 9, pp. 105–118, Sep. 2020. [Online]. Available: https://doi.org/10.239%2Fjcc.2020.09.009
[83] R. Brown, J. Carlyle, I. Grigg, and M. Hearn, "Corda: An introduction," *R3 CEV*, vol. 1, no. 15, p. 14, Sep. 2016.
[84] J. Benet, "IPFS—Content addressed, versioned, P2P file system," 2014, *arXiv:1407.3561*.
[85] M. Shayan, C. Fung, I. Beschastnikh, and C. J. M. Yoon, "Biscotti: A ledger for private and secure peer-to-peer machine learning," 2018, *arXiv:1811.09904*.
[86] C. Fung, C. J. M. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," Aug. 2018, *arXiv:1808.04866*.
[87] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.
[88] G. Radanovic, B. Faltings, and R. Jurca, "Incentives for effort in crowdsourcing using the peer truth serum," *ACM Trans. Intell. Syst. Technol.*, vol. 7, no. 4, pp. 1–28, 2016.
[89] L. S. Shapley, *A Value for N-Person Games*, A. M. Morse, Ed. Princeton, NJ, USA: Princeton Univ. Press, 1953, p. 343.
[90] K. Bonawitz et al., "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. CCS*, 2017, pp. 1175–1191.
[91] L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the Web]," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.
[92] A. Krizhevsky, V. Nair, and G. Hinton. "The CIFAR-10 dataset." 2014. Accessed: Oct. 18, 2022. [Online]. Available: https://www.cs.toronto.edu/∼kriz/cifar.html
[93] M. Everingham, L. van Gool, C. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.
[94] M. S. Jere, T. Farnan, and F. Koushanfar, "A taxonomy of attacks on federated learning," *IEEE Security Privacy*, vol. 19, no. 2, pp. 20–28, Mar./Apr. 2021.
[95] Z. Wang, Q. Hu, and Z. Xiong, "Resource optimization for blockchain-based federated learning in mobile edge computing," 2022, *arXiv:2206.02243*.
[96] A. M. Pinto, "An introduction to the use of zk-SNARKs in blockchains," in *Mathematical Research for Blockchain Economy*. Cham, Switzerland: Springer, 2020, pp. 233–249.
[97] H. Yang, J. Zhao, Z. Xiong, K.-Y. Lam, S. Sun, and L. Xiao, "Privacy-preserving federated learning for UAV-enabled networks: Learning-based joint scheduling and resource management," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 10, pp. 3144–3159, Oct. 2021.
[98] 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.
[99] L. U. Khan et al., "Federated learning for edge networks: Resource optimization and incentive mechanism," *IEEE Commun. Mag.*, vol. 58, no. 10, pp. 88–93, Oct. 2020.
[100] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for Internet of Things: Recent advances, taxonomy, and open challenges," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1759–1799, 3rd Quart., 2021.
[101] K. Guo, S. Han, S. Yao, Y. Wang, Y. Xie, and H. Yang, "Software–hardware codesign for efficient neural network acceleration," *IEEE Micro*, vol. 37, no. 2, pp. 18–25, Mar./Apr. 2017.
[102] Z. Wang et al., "PipeFL: Hardware/software co-design of an FPGA accelerator for federated learning," *IEEE Access*, vol. 10, pp. 98649–98661, 2022.
[103] S. Caldas et al., "LEAF: A benchmark for federated settings," Dec. 2018, *arXiv:1812.01097*.
[104] W. Wang, "Implementation of federated learning on raspberry pi boards: Implementation of federated learning on raspberry pi boards with paillier encryption," Dept. Inf. Netw. Eng., KTH, Stockholm, Sweden, 2021.
[105] B. Y. Lin et al., "FedNLP: Benchmarking federated learning methods for natural language processing tasks," 2021, *arXiv:2104.08815*.
[106] M. Servetnyk, C. C. Fung, and Z. Han, "Unsupervised federated learning for unbalanced data," in *Proc. Global Commun. Conf. (GLOBECOM)*, Dec. 2020, pp. 1–6.
[107] F. Sattler, A. Marban, R. Rischke, and W. Samek, "CFD: Communication-efficient federated distillation via soft-label quantization and delta coding," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 4, pp. 2025–2038, Jul./Aug. 2022.
[108] T. F. Schranghamer, A. Oberoi, and S. Das, "Graphene memristive synapses for high precision neuromorphic computing," *Nat. Commun.*, vol. 11, no. 1, pp. 1–11, 2020.
[109] H. Yang et al., "Lead federated neuromorphic learning for wireless edge artificial intelligence," *Nat. Commun.*, vol. 13, no. 1, pp. 1–12, 2022.
[110] D. Oliveira, T. Gomes, and S. Pinto, "uTango: An open-source TEE for IoT devices," *IEEE Access*, vol. 10, pp. 23913–23930, 2022.
[111] R. Jia et al., "Towards efficient data valuation based on the Shapley value," 2020, *arXiv:1902.10275*.
[112] T. Wang, J. Rausch, C. Zhang, R. Jia, and D. Song, "A principled approach to data valuation for federated learning," 2020, *arXiv:2009.06192*.
[113] Y. Liu and J. Wei, "Incentives for federated learning: A hypothesis elicitation approach," 2020, *arXiv:2007.10596*.
[114] H. Lv et al., "Data-free evaluation of user contributions in federated learning," in *Proc. 19th Symp. WiOpt*, 2021, pp. 1–8.
[115] D. Kahneman and A. Tversky, "Prospect theory: An analysis of decision under risk," *Econometrica*, vol. 47, no. 2, pp. 263–292, 1979.
[116] A. Tversky and D. Kahneman, "Advances in prospect theory: Cumulative representation of uncertainty," *J. Risk Uncertainty*, vol. 5, no. 4, pp. 297–323, 1992.

Leon Witt received the first master's degree in mechanical engineering and business administration from RWTH Aachen University, Aachen, Germany, in 2016, with exchange semesters in Zürich and Los Angeles, and the second master's degree in industrial engineering from Tsinghua University, Beijing, China, in 2017, where he is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology.

His research interests lie at the intersection of federated artificial intelligence, blockchain, and mechanism design.

Mathis Heyer received the bachelor's degree in mechanical engineering from RWTH Aachen University, Aachen, Germany, in 2021. He is currently pursuing the master's degree in chemical engineering and industrial engineering with RWTH Aachen University and Tsinghua University, Beijing, China.

As a visiting student, he spent the academic year 2019/2020 with Carnegie Mellon University, Pittsburgh, PA, USA. His current research interests lie in the applications of artificial intelligence in fields, such as chemical engineering and industrial engineering.

Kentaroh Toyoda (Member, IEEE) was born in Tokyo, Japan, in 1988. He received the B.E. and M.E. degrees and the Ph.D. degree in engineering degree from the Department of Information and Computer Science, Keio University, Yokohama, Japan, in 2011, 2013, and 2016, respectively.

He was an Assistant Professor with Keio University from April 2016 to March 2019 and is currently a Scientist with the Institute of High Performance Computing, Agency for Science, Technology and Research, Singapore. His research interests include blockchain, mechanism design, security and privacy, and data analysis.

Wojciech Samek (Member, IEEE) received the master's degree in computer science from the Humboldt University of Berlin, Berlin, Germany, in 2010, and the Ph.D. degree from the Technical University of Berlin, Berlin, in 2014.

He is a Professor with the Department of Electrical Engineering and Computer Science, Technical University of Berlin and is jointly heading the Department of Artificial Intelligence, Fraunhofer Heinrich Hertz Institute, Berlin. He is an Associate Faculty with Berlin Institute for the Foundation of Learning and Data (BIFOLD) and the ELLIS Unit Berlin. He has coauthored more than 150 peer-reviewed publications, several of which were listed by Thomson Reuters as highly cited papers. His research interest includes deep learning, explainable and thrustworthy AI, and federated learning.

Prof. Samek is a recipient of multiple best paper awards, including the 2020 Pattern Recognition Best Paper Award and the 2022 Digital Signal Processing Best Paper Prize. He is a Senior Editor at IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and serves on the editorial boards of *Pattern Recognition*. Furthermore, he is an Elected Member of the IEEE MLSP Technical Committee.

Dan Li (Senior Member, IEEE) received the B.Sc. degree in computer science from Beijing Normal University, Beijing, China, in 2003 and the Ph.D. degree in computer science from Tsinghua University, Beijing, in 2008.

He is currently a Full Professor with the Department of Computer Science and Technology, Tsinghua University. He authored the Network Architecture, System, and Protocols research group, which is part of the networking research lab. He joined the faculty of Tsinghua University in March 2010, after two years working in the Wireless and Networking Group of Microsoft Research Asia as an Associate Researcher. His main research direction includes trustworthy Internet, data center networks, and data-driven networking.