Blockchain-Based Federated Learning for Securing Internet of Things: A Comprehensive Survey

Article *in* ACM Computing Surveys · September 2022

DOI: 10.1145/3560816

CITATIONS
169

READS
2,328

5 authors, including:

Wael Issa
Helwan University
5 PUBLICATIONS **215** CITATIONS

SEE PROFILE

Nour Moustafa
UNSW Canberra
254 PUBLICATIONS **15,225** CITATIONS

SEE PROFILE

Benjamin Peter Turnbull
UNSW Sydney
84 PUBLICATIONS **4,254** CITATIONS

SEE PROFILE

Blockchain-Based Federated Learning for Securing Internet of Things: A Comprehensive Survey

WAEL ISSA, NOUR MOUSTAFA, and BENJAMIN TURNBULL, UNSW Canberra
NASRIN SOHRABI and ZAHIR TARI, RMIT

The Internet of Things (IoT) ecosystem connects physical devices to the internet, offering significant advantages in agility, responsiveness, and potential environmental benefits. The number and variety of IoT devices are sharply increasing, and as they do, they generate significant data sources. Deep learning (DL) algorithms are increasingly integrated into IoT applications to learn and infer patterns and make intelligent decisions. However, current IoT paradigms rely on centralized storage and computing to operate the DL algorithms. This key central component can potentially cause issues in scalability, security threats, and privacy breaches. Federated learning (FL) has emerged as a new paradigm for DL algorithms to preserve data privacy. Although FL helps reduce privacy leakage by avoiding transferring client data, it still has many challenges related to models' vulnerabilities and attacks. With the emergence of blockchain and smart contracts, the utilization of these technologies has the potential to safeguard FL across IoT ecosystems. This study aims to review blockchain-based FL methods for securing IoT systems holistically. It presents the current state of research in blockchain, how it can be applied to FL approaches, current IoT security issues, and responses to outline the need to use emerging approaches toward the security and privacy of IoT ecosystems. It also focuses on IoT data analytics from a security perspective and the open research questions. It also provides a thorough literature review of blockchain-based FL approaches for IoT applications. Finally, the challenges and risks associated with integrating blockchain and FL in IoT are discussed to be considered in future works.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Computing methodologies** → **Machine learning**; • **Security and privacy** → **Distributed systems security;**

Additional Key Words and Phrases: Blockchain, federated learning (FL), deep learning (DL), Internet of Things (IoT), security of data analytics

ACM Reference format:
Wael Issa, Nour Moustafa, Benjamin Turnbull, Nasrin Sohrabi, and Zahir Tari. 2023. Blockchain-Based Federated Learning for Securing Internet of Things: A Comprehensive Survey. *ACM Comput. Surv.* 55, 9, Article 191 (January 2023), 43 pages.
https://doi.org/10.1145/3560816

The UNSW team received financial support from UNSW Canberra for carrying out this work. The RMIT team received financial support from CloudTech Group for some of the work carried out in this article. The CloudTech project relates to developing Proofs of Space-Time (PoST) consensus algorithms for Large-Scale Green Bitcoin Platforms.
Authors' addresses: W. Issa, N. Moustafa (corresponding author), and B. Turnbull, School of Engineering and Information Technology, 124 La Trobe St, Melbourne, VIC 3000 this for the last two authors; emails: w.issa@adfa.edu.au, {nour.moustafa, benjamin.turnbull}@unsw.edu.au; N. Sohrabi and Z. Tari, Centre of Cyber Security Research & Innovation (CCSRI), School of Engineering and Information Technology, University of New South Wales, Northcott Drive, Canberra, ACT, Australia, 2602; emails: {nasrin.sohrabi, zahir.tari}@rmit.edu.au.

0360-0300/2023/01-ART191 $15.00
https://doi.org/10.1145/3560816

1 INTRODUCTION

The **Internet of Things (IoT)** has become a quickly growing technology and is attracting consumer, industry, and academic interest. IoT has the potential to integrate with and improve virtually every aspect of our lives: our homes, transportation, cities, sports, commercial outlets, education, and industrial manufacturing. IoT refers to lightweight devices connected to the Internet, which links physical and digital domains to offer automated services to end users and organizations [92]. IoT systems enable people and organizations to work more efficiently—saving time, energy, and money [30]. It is expected that the number of IoT devices will be approximately 50 billion by 2030 [139]. By 2025, the volume of data generated by IoT devices will be 79.4 zettabytes and revenue of $3 trillion by 2026 over all the world [139].

IoT systems are complex and heterogeneous by nature, which is reflected in the data they produce. These data sources include sensor telemetry data, actuator data, network traffic, command and control, videos, and images [67]. The value of IoT is in the quick, automated decision making that occurs once sensor data is analyzed, which will affect change in the environment. It is critical to securely analyze these data sources to drive new insights and make smart decisions using **artificial intelligence (AI)** techniques. However, there are several reasons that security and privacy are seen as two of the most severe issues confronting IoT adoption. The diverse and resource-constrained IoT devices make upgrades and patching challenging, and many devices are designed without security in mind. Further, IoT devices have limited energy and computational resources, so they cannot execute sophisticated security mechanisms. Since IoT devices are often connected to core infrastructure, gather sensitive data, and deal with significant assets, they are prime targets for security and privacy attacks. The other major issue is that most IoT architectures rely on centralized trusted servers for coordination, data collection, and storage. The collation of such data poses additional potential security risks and is a high-impact target for adversaries [88, 131].

Machine learning (ML) and **deep learning (DL)** algorithms have been widely used in IoT systems to make real-time decisions and interact intelligently with their surrounding environment [79]. In addition, they can anticipate IoT threats early on by observing system behavior to ensure the security and privacy of IoT applications. However, assaults such as inference, poisoning, inversion, and impersonation may compromise model training and effectiveness [131, 136]. The conventional ML and DL approaches employed in IoT data analysis depend entirely on sending all data to a central server(s) responsible for pre-processing, scaling the collected data, and then training the ML models. This architecture ensures that deployed IoT devices can be low powered and cheap to produce and sell, relying only on networking infrastructure and moving the data storage and processing capacity to external systems or cloud-based solutions. However, this model is susceptible to issues related to communication overhead, privacy leakage, and computation delay, and may be vulnerable to specific types of attacks. As the scale and complexity increase, the traditional and centralized ML approaches are unsuitable for IoT data analysis [67].

Federated learning (FL) is currently one of the fastest-growing paradigms of ML due to its decentralized approach to data, security, and privacy, the latter of which may comply with the impending data privacy regulations [132]. The concept of FL was initially invented by Google, primarily for locally training ML models on devices, with the benefit of providing additional privacy preservation. This invention is motivated by the issues associated with centralized ML. Furthermore, the distributed nature of IoT devices and their reasonable computational and storage capabilities encourage the concept of decentralized learning by using FL [5]. FL is a distributed ML paradigm that allows clients to train and validate ML models locally with their data and then send the model parameters to the central server(s). The central server aggregates the model parameters received from clients and sends the aggregated global model back to the clients for the

next training round. However, this paradigm still utilizes a central architectural design, is partially decentralized, and relies on a central node [5].

Although there are benefits to FL, a centralized FL paradigm suffers from a single point of failure, which indicates that the entire FL system could be affected if the central server crashes due to a malfunction or in the event of a malicious attack. In addition, it is vulnerable to **distributed denial of service (DDoS)** attacks and privacy breaches caused by malicious threats, such as those attacks that attempt to change or extract sensitive information about participating clients from model updates. Moreover, centralized FL systems are not scalable enough to handle the rising amount of local updates made by IoT devices due to the growing number of IoT devices [77, 109]. Blockchain is a potential decentralized ledger system with benefits such as decentralization, immutability, and security, which would be employed to address the centralized FL challenges. These remarkable properties make it an appealing solution for delivering decentralized, improved security, and scalability FL training without relying on a central server with a single point of failure and security breaches. The combination of blockchain and FL allows for the creation of decentralized, improved security, enhanced privacy-preserving, and trusted architectures for untrustworthy decentralized environments such as IoT [109, 145].

1.1 Research Motivation

Although the centralized FL paradigm avoids moving local data to a central server(s), it still has challenges [67]. First, the centralized design is a single point of failure and is susceptible to attacks on availability, such as DDoS. Second, this paradigm suffers from potential privacy leakage issues, where any malicious central server can infer the users' sensitive information from the local model updates provided. Third, a communication overhead occurs due to the extensive exchange of model updates between the enormous number of devices and the central server, creating inefficiencies and bottlenecks. Fourth, the central server is not scalable enough to manage and aggregate the local model from the increasing number of end devices [109]. At this scale, many of the benefits of centralization are lost. Due to their limited communication resources, in a non-responsive central server or network failure, many IoT devices drop their local parameter update payload and cease updating or become unresponsive.

To overcome the issues of how IoT devices create robust and resilient communications paths and networks, **collaborative federated learning (CFL)** has been developed. CFL enables IoT devices in environments with limited or intermittent communication resources to share their local updates with their neighbor devices, allowing for local aggregation. Afterward, the locally aggregated updates are sent to the central server for global model aggregation. CFL alleviates the issue of IoT devices with insufficient communication resources while partially mitigating some privacy issues by sending aggregated model updates to the central server instead of the original local updates. However, it suffers from robustness, privacy, and security issues [5, 67]. It also requires significant trust with neighboring devices. There is a need for a decentralized FL to overcome the potential security and scalability issues associated with the central FL and CFL. Moreover, dispersed FL has been proposed to employ a distributed learning architecture and provide distributed learning for a global FL model. This, therefore, provides FL with more efficient communication resource reuse. However, dispersed FL still depends on central entities to aggregate the sub-global models and does not guarantee the privacy of participating clients [67].

Blockchain technology has remarkable characteristics that enable the development of secure platforms, and these have been applied to decentralized FL in IoT networks [122, 132]. Thus, several academians have regarded blockchain technology as a suitable platform for developing a decentralized computing system based on FL without needing a centralized entity [132]. Blockchain technology offers promising potential to mitigate certain forms of threat and raise cybersecurity

maturity in untrusted environments. This is possible, as blockchain provides novel methods to design trusted FL solutions for IoT systems [54]. Hence, the integration of blockchain and FL is widely leveraged in various IoT applications to enhance robustness, security, and privacy, as blockchain is becoming a significant complementary technology to FL [110, 122].

1.2 Contributions

As we can see in the next section, different survey articles recently comprehensively reviewed the FL for IoT and blockchain for IoT. There are, however, a few survey studies that comprehensively analyze blockchain-enabled FL for IoT. Consequently, the related surveys addressed the fundamentals of blockchain and FL. They explored the critical role of combining blockchain and FL in enhancing the security and privacy of IoT applications briefly. Furthermore, some of these surveys explored the IoT applications that might benefit from the blockchain and FL integration. They did not, however, examine recent works of blockchain-enabled FL and its limitations in IoT contexts. Furthermore, they did not address the security and privacy concerns raised by FL in IoT, nor did they examine the suggested solutions to these concerns. As a result, this comprehensive survey study aims to bridge gaps in current surveys by providing the reader with a comprehensive review of FL, blockchain, and the significance of combining them in allowing secure intelligence in IoT environments. Accordingly, the key contributions of this survey can be summarized as follows:

- We provide a comprehensive review of the fundamental concepts of blockchain and FL in IoT systems.
- We provide a new taxonomy for FL that considers the recent advances in FL. Moreover, we discuss how the blockchain can be leveraged in IoT applications.
- We present a detailed discussion of recently proposed approaches that integrate blockchain and FL to build secure, robust, efficient, and risk-free IoT applications.
- We propose a threat model for FL and a general architecture for blockchain-enabled FL.
- Finally, we discuss the lessons learned and key challenges of integrating blockchain and FL in IoT applications. Furthermore, the possible future directions to build efficient and risk-free blockchain-enabled FL architecture for IoT data analytics are also outlined.

1.3 Outline

The rest of the article is organized as follows. Section 2 explains the current state of the art and outlines related surveys, focusing on their contributions compared with our work. Section 3 provides the background of FL and investigates the recent security and privacy challenges when applied to IoT data analytics. An overview of IoT, its architectures, and its security concerns are discussed in Section 4. Section 5 provides a review of blockchain, including its architecture, consensus algorithms, platforms, and smart contracts. A detailed literature review for blockchain-based FL approaches for IoT applications is presented in Section 6. Section 7 summarizes the key current issues in these fields' nexus and highlights lessons learned. In Section 8, the current challenges and future research directions are highlighted. Finally, Section 9 concludes the survey and summarizes the key findings.

2 RECENT SURVEYS RELATED TO FL AND BLOCKCHAIN FOR IOT

Several survey studies have reviewed the adoption of FL or blockchain technologies in IoT applications. Our survey has reviewed approximately 180 studies that concentrate on FL, blockchain, and IoT, and how these technologies have been applied to address challenges related to the security of IoT ecosystems. Moreover, some of these studies have focused on investigating the complementary relationship between blockchain and FL in alleviating security and privacy issues associated

with IoT systems. This work analyzed the surveys published in these fields between 2018 and 2022. It includes the studies published in peer-reviewed conferences and journal papers from libraries such as IEEE, Elsevier, ACM, and Springer. The summary of the related surveys and their key contributions is listed in Table 1. We categorize the related survey papers into three categories based on the focus of each set of them. The first category examines FL for IoT applications, whereas the second reviews blockchain for IoT applications. Finally, the third category comprehensively investigates blockchain-enabled FL for IoT applications. The following sections demonstrate the authors' contribution to each category.

2.1 FL for IoT

FL is a distributed and collaborative ML paradigm that can be critically employed to analyze the massive amounts of data generated by IoT devices, discover patterns and processes, and then enable intelligent decisions. Using FL, each IoT device trains ML models locally on its data without having to submit that data to a central server. As a result, FL alleviates the security and privacy concerns that traditional centralized ML algorithms have raised for IoT applications [5, 32]. To enable the next generation of intelligent edge networks and handle security and scalability concerns associated with the central server, it is critical to build more decentralized FL methods [109].

Nguyen et al. [108] presented a comprehensive review of the role of FL in developing safe and privacy-enhanced IoT applications. In addition, this work discussed how FL could be leveraged in various IoT services and outlined the challenges behind the integration of FL and IoT. Imteaj et al. [57] studied FL methods for IoT devices with limited resources. Following that, the difficulties of adopting FL for resource-constrained IoT devices are highlighted. The latest FL advances in enabling smart IoT applications are addressed in depth by Khan et al. [68]. This work also presents a taxonomy for recent advances in FL. Finally, the authors have established a set of measures that are used to assess recent advances in FL for IoT networks. These metrics included sparsification, robustness, quantization, scalability, security, and privacy. Nguyen et al. [110] provided a comprehensive survey for the FL-enabled **Internet of Medical Things (IoMT)**. The authors discussed the shortcomings of IoMT and the recent advances in FL-enabled healthcare. They also discussed how to use the blockchain to secure FL-enabled healthcare applications.

2.2 Blockchain for IoT

Two key advantages of blockchain are safeguarding storage and providing secure **peer-to-peer (P2P)** interactions. Consequently, blockchain can benefit IoT applications by protecting data storage against attacks and securing peer-to-peer interactions in blockchain-based IoT applications. The distributed and tamper-proof ledger of blockchain can improve the reliability of IoT networks. Furthermore, blockchain employs cryptographic techniques, making it suited for reducing privacy concerns in IoT networks [76]. This is a consistent theme throughout much of the academic literature. Wu et al. [157] offered a comprehensive study on blockchain consensus methods and applications, particularly in the IoT industry. The authors divided the research effort in blockchain for IoT into four layers: the data layer, network layer, consensus layer, and application layer. Furthermore, the proposed approaches in each layer were reviewed.

These distinctions provide insight into the different facets of blockchain that can be used for securing IoT applications. Panarello et al. [118] introduced a thorough study of blockchain applications in IoT. This study divided blockchain usage in IoT into two categories: adoption for device manipulation and adoption for data management. An investigation of the combination of blockchain for IoT (BCoT) was discussed by Dai et al. [35]. This survey provided a comprehensive review of the opportunities of adopting blockchain in IoT. A systematic analysis of the security of IoT-based blockchain approaches was provided by Da Xu et al. [34]. This article

Table 1. Summary of Related Surveys and Their Key Contributions

References	Review IoT Background Information	Review IoT Privacy and Security	Review Blockchain Background Information	Explain the Significance of Blockchain for IoT	Review Smart Contracts	Discuss Concerns of Blockchain for IoT	Review FL Background Information	Propose FL Taxonomy	Explain the Significance of FL for IoT	Demonstrate FL Privacy and Security Concerns	Review Recent Works for Blockchain-Enabled FL	Propose General Architecture	List Challenges and Future Directions
Nguyen et al. [108]	✓						✓	✓	✓	✓			
Imteaj et al. [57]							✓	✓	✓				
Khan et al. [68]							✓	✓	✓	✓			
Nguyen et al. [110]	✓						✓		✓	✓			
Wu et al. [157]			✓	✓	✓	✓							
Panarello et al. [118]			✓	✓	✓								
Saxena et al. [129]	✓	✓	✓	✓	✓	✓							
Dai et al. [35]	✓	✓	✓	✓	✓	✓							
Da Xu et al. [34]		✓	✓	✓	✓								
Reyna et al. [125]			✓	✓	✓	✓							
Kumari et al. [73]	✓	✓	✓	✓	✓	✓							
Majeed et al. [93]			✓	✓		✓							
Al Sadawi et al. [3]			✓	✓	✓	✓	✓						
Alfandi et al. [4]	✓	✓	✓	✓	✓								
Kumar and Sharma [72]	✓	✓	✓	✓	✓	✓	✓						
Noby and Khattab [114]		✓	✓	✓									
Khan and Salah [69]		✓	✓	✓									
Iftikhar et al. [56]	✓		✓	✓	✓	✓							
Banerjee et al. [16]		✓		✓									
Dwivedi et al. [41]				✓	✓								
Huo et al. [55]	✓		✓	✓	✓								
Latif et al. [75]	✓	✓	✓	✓	✓								
Alladi et al. [6]				✓	✓								
Nguyen et al. [109]			✓			✓	✓			✓	✓		✓
Hou et al. [54]				✓							✓		
Lee and Kim [76]			✓				✓						✓
Ali et al. [5]			✓	✓	✓	✓	✓	✓	✓	✓			✓
Li et al. [77]				✓		✓		✓			✓		
This survey	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

comprehensively reviewed the security features, problems, and approaches of IoT-based blockchain. Finally, the authors concluded that integrating IoT and blockchain is vital in building state-of-the-art decentralized and secure IoT applications.

Reyna et al. [125] investigated the feasibility of incorporating blockchain into IoT applications. This work has been instrumental in examining the fundamental issues that emerge due to that adoption. This survey concluded that blockchain adoption revolutionizes IoT applications, particularly when evaluating and overcoming existing issues. A full analysis of blockchain-based IoT for smart city applications was conducted by Kumari et al. [73], which examined how blockchain and 6G networks might help solve a single point of failure, security, and privacy challenges in IoT applications, particularly in the smart city sector. Similarly, Majeed et al. [93] also reviewed the importance of blockchain in smart city applications.

Saxena et al. [129] discussed the critical significance of blockchain in resolving data and privacy challenges related to IoT applications. Moreover, the authors highlighted the security gains made by adopting blockchain into IoT ecosystems. The authors recommended the adoption of blockchain in the development of safe, transparent, and reliable IoT systems. Al Sadawi et al. [3] outlined the issues associated with blockchain and IoT integration. The authors proposed a three-layer architecture to demonstrate the benefits of blockchain features in constructing safe and decentralized IoT applications: the IoT devices layer, dew layer, and cloudlet layer. The suggested architecture's primary goal is to bring servers closer to IoT devices, although secondary benefits of this paradigm were also outlined.

Alfandi et al. [4] concentrated on identifying existing security and privacy concerns connected to IoT architectures. This work investigated blockchain as a promising technology solution for overcoming such challenges. According to the authors, combining blockchain, IoT, and ML will offer value, notably in intrusion detection. Kumar and Sharma [72] demonstrated the significance of integrating blockchain to ensure IoT device trust. Furthermore, this study investigated the challenges of implementing trustworthy IoT through blockchain. This survey concluded that combining blockchain with IoT has alleviated many security and trust problems; however, additional research is required to develop effective blockchain-based IoT solutions.

Noby and Khattab [114] examined the benefits of blockchain in various IoT applications. This work stated that using blockchain in IoT applications is beneficial, yet several issues have been presented that should be addressed in future research. Khan and Salah [69] outlined the IoT security issues. This work classified these issues according to the three layers of IoT system design. This work also provides potential solutions for each of the issues depicted in the work. One of the recommended solutions is to investigate the use of blockchain to address various IoT security challenges. This is consistent with the work of Iftikhar et al. [56], which focused on the privacy concerns raised by IoT applications. This work investigates several blockchain applications to identify the potential of leveraging the blockchain in overcoming IoT privacy challenges. The authors claimed that the blockchain is a promising solution for monitoring and tracking IoT applications. Banerjee et al. [16] focused on blockchain for data integrity and secure data sharing in IoT applications. Finally and certainly not least, the state-of-the-art implementations of blockchain in **Industrial Internet of Things (IIoT)** were reviewed by Dwivedi et al. [41].

Huo et al. [55] presented an in-depth analysis of the use of blockchain technology in IIoT. This work provided a review of the many advantages that blockchain technology has for IIoT applications. In addition to this, it discussed the layered architecture of blockchain, as well as the three-tier architecture of IIoT, and how to implement this architecture in a way that ensures the security of IIoT applications. The authors concluded their discussion by outlining the many applications of blockchain in IIoT and the difficulties associated with integrating the blockchain and IIoT. However, this work demonstrates blockchain's great potential for IIoT applications. Also highlighting

the significant potential of blockchain for IIoT applications, Latif et al. [75] provided a comprehensive analysis of the IIoT's security issues and how the blockchain may help mitigate them. This study began by examining the basics, and general designs of blockchain and the IIoT, then examined the numerous IIoT application cases for blockchain technology and concluded with the authors describing the obstacles posed by the integration of blockchain and IIoT.

2.3 Blockchain-Enabled FL in IoT

The integration of blockchain and FL ensures the secure exchange of local model updates for IoT end devices [115]. This integration, in turn, leads to the development of blockchain-based FL IoT applications. In blockchain-enabled FL applications, the blockchain replaces the central server by implementing and executing smart contracts [76]. In recent years, few works have highlighted the potential benefits and issues behind integrating FL and blockchain in IoT applications. For example, Nguyen et al. [109] reviewed the opportunities of combining the blockchain with FL to enable decentralized, secure, privacy-enhancing, and intelligent MEC networks. The authors have presented an overview of FLchain, a new paradigm in MEC enabled by the combination of FL with blockchain. A general FLchain architecture is shown, paving the way for scalable and secure edge intelligence.

Hou et al. [54] presented a systematic literature review for blockchain-based FL architectures and applications. The main goal of this survey was to discuss how to leverage blockchain to build secure FL architectures. According to the authors, using the consortium or private types of blockchain is recommended for building a more secure blockchain-based FL. In addition, it is necessary to use a suitable privacy-preserving technique to avoid privacy leakage. An overview of how blockchain may be used to improve the security and privacy of FL was presented by Lee and Kim [76]. This article examined some blockchain-enabled FL applications, notably those connected to IoT, the **Internet of Vehicles (IoV)**, and healthcare. The authors concluded that using blockchain technology in FL applications is critical in improving those systems' security, privacy, and performance. Ali et al. [5] presented a comprehensive survey for blockchain and FL and their applications in IoT. Moreover, it discussed the privacy preservation techniques employed to enhance blockchain-enabled IoT systems. The related privacy issues in IoT applications were also outlined. To enhance privacy, the authors integrate a blockchain-enabled traceability function in dispersed FL and investigate its impact on real scenarios.

Li et al. [77] illustrated the significance of combining FL with the blockchain. This article presented background information on FL and blockchain and an evaluation of many works that seek to enhance the performance of FL utilizing blockchain technology. In addition, it describes the applications briefly that profit from the combination of FL and blockchain. One of these applications is IoT applications.

3 FEDERATED LEARNING

As the number of IoT devices increases, so does the generated data volume. This results in more privacy concerns, especially in the case of centralized AI solutions. Hence, the centralized AI solution is less appropriate for modern IoT applications due to the heterogeneous nature of IoT devices' data, resources, and distribution across multiple geographical locations [109]. Recently, FL architectures have been proposed that aim to mitigate the privacy and security issues associated with centralized AI solutions by avoiding sharing the private local data of IoT devices with the central server [109]. Instead, each IoT device participating in the FL process trains the ML model locally with its data. Consequently, the devices only share the learned model parameters with the central server for global model aggregation. Thus, FL helps improve privacy, save communication

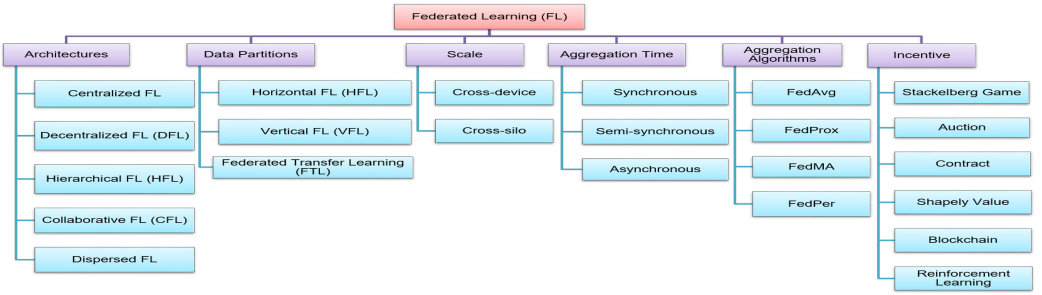

Fig. 1. FL taxonomy.

resources, and make the training process more robust compared to centralized ML [109]. In general, the FL process can be outlined in the following basic three steps [57, 102, 169]:

(1) *Model selection and initialization:* This step determines which type of ML is the most suitable and whether it is a pretrained or new model that needs to be trained from scratch. Besides that, the model parameters are initialized W_G^0, and the model is broadcast to the participating clients N to start the FL training rounds E, where W_G^0 is the initial global model, N is the total number of clients, and E refers to the total number of global training rounds.

(2) *Local model training:* In this step, every client n receives the W_G^0 and begins the local training on its data $D_n \in D$, where $D = \{D_1 \cup D_2 \cup \cdots \cup D_N\}$. Hence, in every global round $e \in E$, each client n calculates the local model update w_n^e by minimizing a loss function $\mathcal{F}(w_n^e)$ as follows:

$$w_n^{e*} = \arg\min \mathcal{F}(w_n^e), n \in \{1, \ldots, N\}, e \in \{1, \ldots, E\}. \tag{1}$$

After the local training is finished for round e, the local learned parameters w_n^e will be sent to the central entity for aggregation.

(3) *Global model aggregation:* Finally, the central entity receives the local model parameters from various clients and aggregates the global model as follows:

$$W_G^e = \frac{1}{|D|} \sum_{n=1}^{N} |D_n| w_n^e, n \in \{1, \ldots, N\}, e \in \{1, \ldots, E\}, \tag{2}$$

where $|D|$ is the size of the whole dataset D and $|D_n|$ is the size of the local dataset for client n. Afterward, the new version of global model W_G^e is broadcast again to all of the participating devices for the next global training round $e + 1$.

The second and the third steps are repeated until the number of training rounds is reached or the predefined accuracy threshold is achieved. The server stops the training at this point, and the trained global model is broadcast to all clients [95]. In Figure 1, we present the taxonomy of FL according to recent FL architectures, data partitions, scale of FL, aggregation time schemes, aggregation algorithms, and FL incentive mechanisms. The details of FL taxonomy are explained in the following sections.

3.1 FL Architectures

In this section, we will discuss various architectures of FL and how the parameters exchange between the clients and FL coordinator.

Centralized FL. Centralized FL depends on a centralized server to initiate the FL process and aggregate the global model [5]. It represents the traditional FL and follows the same flow described in the three preceding steps. It alleviates the privacy issues by avoiding sending the client's sensitive

data to the central server; however, it suffers from robustness issues due to a single point of failure [57, 109].

Decentralized/fully distributed FL. This is an FL architecture that differs from the centralized FL in which it does not depend on a third-party entity to aggregate the global model. Alternatively, all clients connect in a P2P or mutual communication manner to exchange local model updates and aggregate the global model. Centralized FL architectures are unsuitable for environments where it is likely that communication between participating clients and the central entity will be unstable. In such cases, it is recommended to use the distributed FL. For instance, clients in a P2P network can communicate via blockchain ledger to store their local updates and aggregate the global model in a trusted and secure way [109].

Hierarchical FL. **Hierarchical federated learning (HFL)** decreases communication latency while maintaining accuracy. In HFL, mobile users are clustered into groups (clusters) based on their locations, and each cluster is assigned to a **small cell base station (SBS)**. Mobile users train models on their local data and send their local updates to the SBS. This process is repeated for a specific number of iterations. Afterward, all SBS' send the aggregated local updates to the mobile base station (MBS) to aggregate the global model. The experimental results show that the HFL helps reduce the communication latency, speeding up the training of the global model [1].

Collaborative FL. In centralized FL, there are cases where a subset of IoT devices can fail to send their local model updates to the central server. This can happen for several reasons, such as the low energy requirements of the devices, limited communication resources, or transmission delay. Consequently, a CFL framework has been proposed to overcome such challenges. In the CFL framework, not all devices are required to be connected to the central server. Instead, some IoT devices connect to the central server, and the rest connect to the neighboring devices based on the distance between them. This creates a mesh-like network structure. The proposed framework can assist in overcoming some of the limitations of central FL; however, and it also has limitations. These limitations include a slower convergence speed than the central FL; the training process may be affected by imperfect communication between devices and different ML convergence for each group of associated devices. Besides this, the centralized server still represents a single point of failure [31].

Dispersed FL. This is a distributed FL framework that offers the global model learning in two stages. In the first stage, the sub-global model is aggregated within various groups. Each of these groups includes a set of closely located devices. In the second stage, the global model is computed by aggregating the sub-global model in a centralized or distributed manner. There are two categories of dispersed FL: centralized dispersed FL and distributed dispersed FL [67]. However, as reported by Khan et al. [67], dispersed FL still has limitations regarding client privacy and non-IID data.

3.2 FL Schemes Based on Data Partitions

It is important to consider data distributions and partitions while designing FL models. Based on the sample and feature spaces, FL can be categorized into three categories as follows [57, 109].

Horizontal FL. HFL is dubbed as sample-based FL, where the clients participating in the FL process have different data samples with the same feature space. Therefore, participating clients can use the same ML to be trained locally due to the same feature space. For instance, next word prediction models learn from datasets with different sample spaces and the same feature space to predict the next word. More specifically, FL is horizontal FL if the following condition is satisfied:

$$\forall i, j \in \{1, \ldots, N\}, i \neq j \quad D^i_{feature} = D^j_{feature} \quad \wedge \quad D^i_{sample} \neq D^j_{sample}, \tag{3}$$

where N represents the total number of client participants in FL, i and j refer to different clients, $D^i_{feature}$ is the feature space of the client i, $D^j_{feature}$ is the feature space of client j, D^i_{sample} is the sample space from which the dataset D_i is sampled, and D^j_{sample} is the sample space from which the dataset D_j is sampled.

Vertical FL. Vertical FL is known as feature-based FL, where clients' datasets have the same sample space with different feature spaces. For example, in IoT applications, the ML model can be shared among different entities such as banks and e-commerce companies that serve clients in the same city (the same sample space). At the same time, different users will collect various features. For example, a bank may contain a set of features that differs from the features collected by the e-commerce application for the same client (different feature space). Furthermore, the bank and the e-commerce company can cooperatively train ML models using the vertical FL scheme to predict the personalized loans based on the online shopping activities of clients. Hence, FL is called *vertical FL* if the following condition is satisfied:

$$\forall i, j \in \{1, \ldots, N\}, i \neq j \quad D^i_{feature} \neq D^j_{feature} \quad \wedge \quad D^i_{sample} = D^j_{sample}. \tag{4}$$

Hybrid or federated transfer learning. Hybrid or **federated transfer learning (FTL)** is a combination of vertical FL and horizontal FL, where the clients have datasets with different sample spaces and different feature spaces. Furthermore, FTL is vital in transforming different sample spaces' features into the same representation. For instance, FTL can be used in disease diagnosis applications where the global model training depends on clients from different countries (sample space) and have different medical tests (feature space). Hence, FL is called *federated transfer learning* (FTL) if the following condition is satisfied:

$$\forall i, j \in \{1, \ldots, N\}, i \neq j \quad D^i_{feature} \neq D^j_{feature} \quad \wedge \quad D^i_{sample} \neq D^j_{sample}. \tag{5}$$

3.3 FL Scale

FL systems can be categorized into either cross-device or cross-silo, based on the number of participating clients and their data volume [102, 109].

Cross-device. Cross-device is an FL approach that involves a large number of clients with limited data size. Thus, the number of the involved devices ranges from millions to billions. Examples of cross-device FL systems include IoT devices and smartphones. Due to the volume of devices, selecting the most qualified devices (i.e., with enough computational, communication, and energy resources) to participate effectively in the FL is important.

Cross-silo. Unlike the cross-device FL approach, the cross-silo approach has a small number of clients with a large data volume. Such clients may be data centers or organizations. For instance, the Amazon product recommendation model learns via the contribution of numerous data centers, each using its local data.

3.4 FL Aggregation Time Schemes

Based on the time when global model aggregation occurs, FL approaches can be classified into three synchronization schemes, as described in Figure 2, and the details of these three schemes are as follows.

Synchronous FL. Synchronous FL does not take into consideration the heterogeneity of edge devices, whereas edge devices can have heterogeneous computational and energy resources. It therefore does not effectively utilize the resources of participating devices as the high computational devices remain idle until the other devices complete their local training. Moreover, in real scenarios, some of these devices may join midway through the process, whereas others may fail to submit their local updates. However, the speed of rounds in synchronous FL is restricted to the

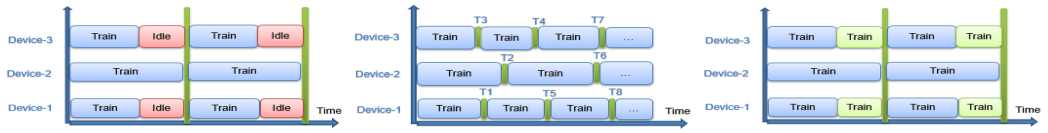

Fig. 2. FL synchronisation. (a) The synchronous FL scheme where T1 and T2 represent the synchronization points at which the devices can communicate with the FL coordinator. (b) The asynchronous FL scheme where each device can communicate with the FL whenever it has finished local training. (c) The semi-synchronous FL scheme where T1 and T2 represent the synchronization points at which the devices can communicate with the FL coordinator, and there is no idle time as in the synchronous scheme.

speed of the slowest device. This causes a "straggler effect," which can cause inefficient processing. Even though synchronous FL consumes low communication resources, it also has a slow learning convergence [42, 140].

Asynchronous FL. In contrast to synchronous FL, asynchronous FL does not have a specific synchronization point, but it allows the participating devices to submit their local updates and download the new versions of the computed global model whenever any of them complete the local training. As a result, asynchronous FL approaches have high convergence and consume more communication resources compared to synchronous FL [26, 140]. An asynchronous FL approach is proposed by Capota et al. [26]. The authors reported that this approach is robust and able to work under imbalanced and non-IID data distributions, and it can therefore overcome the issue of device failures. Another approach for asynchronous FL was proposed by Feng et al. [42] to improve the scalability and efficiency of FL while overcoming the poisoning attacks that target asynchronous FL.

Semi-synchronous FL. Semi-synchronous FL is considered a middle ground solution between the synchronous and asynchronous FL methods. In a semi-synchronous FL, the participating devices are permitted to train the ML locally up to a certain synchronization point where the global model is calculated. As a result, this method lowers communication costs and makes better use of the resources of participating devices. In general, the semi-synchronous FL approach has been proposed to balance communication costs and resource usage [140]. Stripelis et al. [140] proposed a semi-synchronous FL approach that accelerates the model convergence while reducing communication cost. In addition, it improves resource utilization by eliminating the idle time of high computational devices and involving the local update of low computational devices in the global model computation.

More specifically, blockchain as a distributed ledger is utilized to increase the security, privacy, scalability, and efficiency of asynchronous FL. Due to the decentralized nature of blockchain, it is possible to construct a decentralized FL, which prevents DDoS and single point of failure threats. A malicious client cannot tamper with model changes since blockchain is immutable. Furthermore, consensus mechanisms help deploy decentralized FL among untrusted clients, and smart contracts are used to authenticate FL clients and validate local model alterations to prevent malicious model updates [162]. It is worth noting that blockchain is adopted in asynchronous FL to track the aggregation status by maintaining the global round index and enabling secure aggregation and storage for local model updates. For example, Lu et al. [90] proposed a blockchain-enabled asynchronous FL for improving privacy and trust among heterogeneous devices. In this research, blockchain is adapted to store, verify the local updates, and keep track of the asynchronous FL aggregation. Similarly, Lu et al. [89] adopted blockchain to enable secure and validated asynchronous local model aggregation for IoV.

It is worth mentioning that the FL aggregation algorithms and the FL incentive mechanisms are discussed in detail in the supplementary material included with this survey.

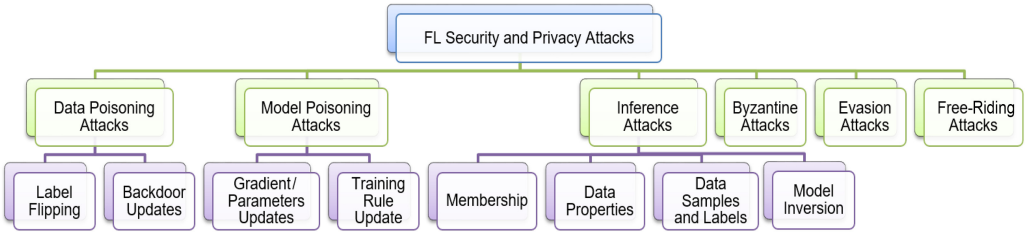

Fig. 3. FL security and privacy concerns.

3.5 FL Security and Privacy Concerns

This section investigates the various security and privacy threats that harm the FL process, as shown in Figure 3, which presents a taxonomy for various attacks that target FL in an IoT environment. Due to the massive amount of data exchanged between IoT devices, privacy risks have increased in IoT environments. Accordingly, these risks have motivated the emergence of FL as an ML paradigm to preserve the privacy of client data by allowing clients to train ML models locally. Because of the collaborative training of ML models through FL, direct data leakage from clients can be prevented [11]. Even though one of the significant benefits of FL over traditional ML solutions is privacy preservation [169], FL does not guarantee adequate privacy protection [47]. Privacy cannot be guaranteed, as adversaries can infer sensitive information about participants from model parameters [180]. For this reason, the FL process is still vulnerable to various privacy and security threats [11].

The primary goal of attacks is to exploit vulnerabilities to compromise system security and expose users' private information (privacy) [102]. In FL, adversaries attempt to gain access to one or more participating clients, and they can manipulate local model updates, train hyperparameters, and inject poisoned data. Consequently, those attacks harm the global model and negatively affect the privacy of clients [14]. The various attacks that target the FL process can be outlined as follows:

(1) *Data poisoning attacks:* Data poisoning attacks aim to disrupt the performance of an ML model by compromising the training data [60]. Such attacks can occur in a variety of ways, including the ones listed next:

—*Label flipping:* Label flipping intentionally fools the ML model by exchanging the labels in the training while preserving the features as is. For example, the poisoning attacks harm MINST by replacing the number 1 labels with number 7 labels and vice versa. At the same time, the original images remained as is other works [60, 63]. Consequently, the trained model will deviate from the expected prediction boundaries. Zhang et al. [171] proposed a data poisoning attack known as PoisonGAN attack because it utilizes a generative artificial neural network to generate poisoned data. According to the authors, this approach effectively compromises the global model by executing both label-flipping and backdoor attacks.

—*Backdoor updates:* By watermarking and adding noise to training data samples, adversaries tamper with them. This attack is difficult to spot because it only affects the model's accuracy on the harmed data, whereas the model is unaffected with other samples [60, 63]. The FL objective is to improve privacy by allowing the participants to train the ML model locally on their datasets. This paves the way for the backdoor attacks to inject noised data during the local training. The backdoor attack manipulates a subset of training data so that models trained on the manipulated dataset will become vulnerable to the test set with the same noised data [159]. Xie et al. [159] proposed DBA, a distributed backdoor attack. DBA broke up the global trigger pattern (noised data) into separate local patterns and added them

to the training data of adversarial clients. Experiments showed that DBA is a powerful backdoor attack against FL.

(2) *Model poisoning attacks:* Model poisoning can manipulate the FL training procedure to change the model gradients/parameters or the learning rule [14, 17, 60]:

—*Gradients/parameters update:* Malicious clients or external adversaries can inject malicious gradients/parameters or manipulate them to degrade the global model's overall performance [60]. For example, Bagdasaryan et al. [14] investigated the impact of the gradients manipulation by malicious clients and found that it negatively impacts global model accuracy. Further, the backdoor updates can be made for the model parameters in which the adversary can tamper with the client's model updates by compromising a set of participants, such attacks are named *Sybil attacks* [2].

—*Training rule update:* Adversaries can perform the model poisoning by modifying the hyperparameters or the objective function of one or more participating clients [60]. For example, the work of Bhagoji et al. [17] aims to modify the objective function to reduce the distinction between the malicious and benign parameter updates.

(3) *Inference attacks:* Inference attacks attempt to infer sensitive information about participants, training data, and labels. The objectives of inference attacks can be divided into the four categories as follows [167]:

—*Inference of membership:* Adversaries examine the training data to see if it contains specific data samples. An adversary may, for example, discover whether or not a certain patient record was used to train an ML model and whether or not that record was linked to a specific disease. Furthermore, the adversary can compromise the model updates of a particular participant in FL, allowing it to determine if a given data sample belongs to this participant's private data or not [96]. Zhang et al. [174] proposed a passive membership attack that malicious participants in FL can perform. This attack used a binary classifier to predict the membership state. The authors mentioned that they employed the **generative adversarial network (GAN)** to increase the diversity of attack data, promoting the training data for the attack model. In FL, membership attack misuses the global model to check specific data in the training data and then extracts information about the clients. Therefore, it can infer information about the training data using guesswork and training a model to predict the actual training data [60, 102]. Nasr et al. [106] looked into the ability of a membership attack to compromise participants' privacy by enlisting the help of adversarial participants, especially when the global model has a high level of prediction accuracy.

—*Inference of data properties:* Adversaries attempt to infer metadata about the training data, such as the distribution of a specific class. More specifically, adversaries adopted GANs to use model parameters to reconstruct synthetic samples. These samples reflect statistical information about the training data. Therefore, GAN attacks can leak private information from the client parameters in FL [60]. Hitaj et al. [53] proposed a GAN attack that can deduce meaningful information about the FL clients. This attack can manipulate the local model parameters of clients—even if they obfuscate their parameters by **differential privacy (DP)**. Adversaries attempt to reconstruct data samples that are not real data samples, but the reconstructed samples can be used to extract sensitive information from the training data. For example, adversaries can use face recognition models to generate data samples similar to the original data, or they can discover how many times each class appears in the training data and the fraction of data for each class [45].

—*Inference of data samples and labels:* The adversary's goal is to reconstruct the original data samples and labels. Zhu et al. [181] presented a deep leakage from gradients (DLG) approach. This approach relies on an optimization algorithm to reconstruct the training data

samples and labels. The experimental results showed that this technique could accurately reconstruct images and texts used in DL model training. Similarly, Wang et al. [153] proposed the mGAN-AI attack. This attack can reassemble actual training samples and target specific clients to compromise and leak their privacy. The authors concluded that the proposed attack outperformed the other attacks in reconstructing the victim's training data samples.

—*Model inversion:* By knowing the confidence scores of the predicted classes in advance, model inversion attacks rely on reverse engineering to infer private information about the clients from model parameters. Moreover, model inversion attacks are adequate for simple structure ML architectures, but they are not serious for DL models [60]. In the work of Fredrikson et al. [44], model inversion attacks have successfully inferred patient genetic markers by exploiting model parameters.

(4) *Byzantine attacks:* Byzantine attacks are intended to degrade the global model convergence. Malicious clients update their data or models so that the global model does not converge well [18, 128]. So et al. [137] presented a Byzantine resilience approach. Based on an integrated stochastic quantization, verifiable outlier detection, and secure model aggregation, this approach aims to ensure model convergence and privacy while defending against the Byzantine attack. Furthermore, many recent works address the issue of Byzantine attacks [28, 48, 82].

(5) *Evasion attacks:* The objective of evasion attacks is to deceive the target model by creating adversarial samples during the prediction phase. Therefore, these adversarial samples might be utilized as input for the global model, causing it to provide inaccurate prediction results. For example, evasion attacks may add undetectable minor noise to the global model's input, which leads to misleading prediction results. Therefore, these attacks hurt the global model after it has been deployed to the end devices [21, 84].

(6) *Free-riding attacks:* In FL, all participants receive the global model after each training round, irrespective of their contributions. Such circumstances promote the existence of "free-riders" who benefit from the global model without participating in the training procedure. As a result, these free-riders are accountable for free-riding attacks, as they do not engage in training and instead provide fake model updates. Free-riders do so for a variety of reasons. First, they may not wish to share their private information or consume their computational and energy resources. Second, they may also want to undermine the privacy of other participants. Third, they lack sufficient data and wish to benefit from the experience of other participants, particularly if the global model has a high economic value [21, 91]. Fraboni et al. [43] developed a theoretical framework to explore and analyze the effect of free-rider attacks on FL. The authors mentioned that the iterative parameters aggregation process in FL leads to an increase in the number of free-riders, as each can download the global model and benefit from it without any contribution. Moreover, the disguised free-riding, which uses stochastic perturbations, was explored. The authors reported that its essential to develop techniques to evaluate the contribution of each participant in FL to detect the free-riding attack and its disguised versions.

As aforementioned, FL is subject to a variety of attacks. We offer a threat model to identify the potential vulnerabilities in the FL process in the IoT environment. As shown in Figure 4, the suggested FL threat model is divided into four stages: the training phase, the parameters exchange phase, the parameters aggregation phase, and the prediction phase. It is worth noting that the FL is subject to a range of security and privacy attacks at each phase, which must be mitigated to safeguard participants' security and privacy. We can summarize the four phases of the threat model as follows.

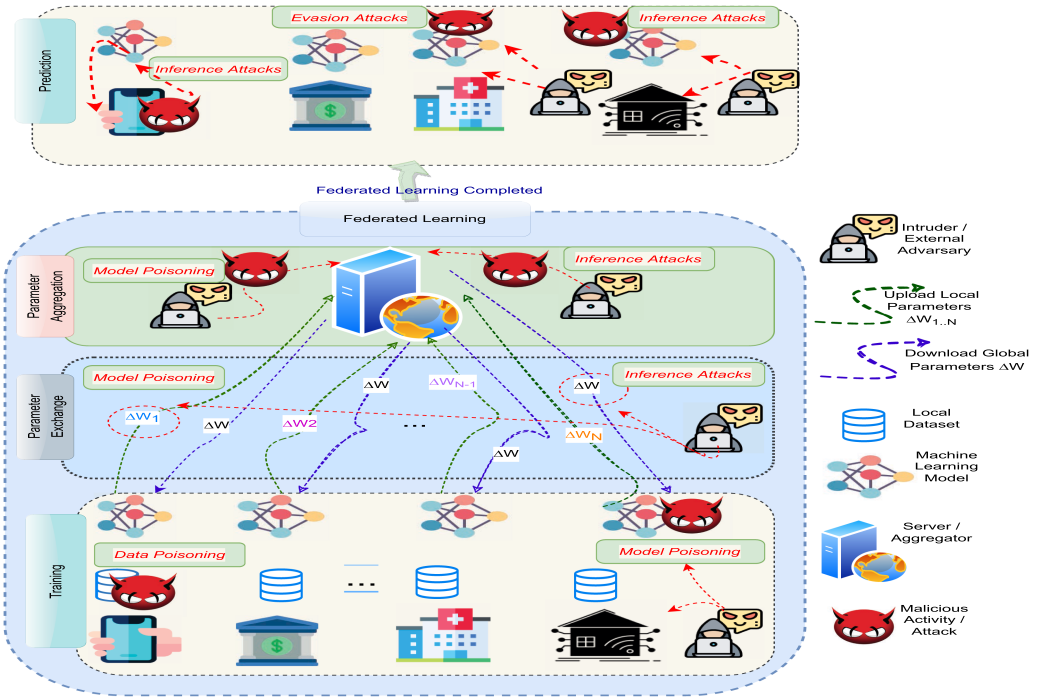

Fig. 4. FL threat model.

Training phase. FL participants may not use effective security protection mechanisms and hence have various vulnerabilities. Adversaries may use these vulnerabilities to carry out data poisoning attacks such as model flipping and backdoor assaults. Furthermore, some participants may be malicious and carry out data poisoning attacks. In addition, model poisoning attacks, such as gradient updates and training rule update assaults, may be carried out either by malicious participants or external adversaries. More specifically, these attacks try to undermine the FL process by undermining the global model's performance.

Parameters exchange phase. During the parameters exchange phase, when participants are either downloading the global model parameters or uploading the local model parameters, intruders have the opportunity to steal and eavesdrop on the updates being exchanged to carry out model poisoning. In addition, intruders have the opportunity to steal the parameters to extract sensitive information about participants, reconstruct the dataset, or infer meta-information about the distribution of data. All of this is accomplished via the use of inference attacks.

Parameters aggregation phase. In FL, the aggregation of parameters is the responsibility of the central server. However, the server itself may be either honest-but-curious or malicious and may carry out inference and model poisoning attacks. In addition, attackers can penetrate the central server and either erroneous poison parameters in the global model or obtain sensitive information about the participants, the training data, and the ML model.

Prediction phase. After the FL has been completed, the global model will be deployed to end devices, regardless of whether any of those devices are malicious or not. As a result, malicious devices or adversaries continue to target the global model by performing evasion and inference attacks. By employing noise and adversarial samples that the end devices cannot see, evasion assaults cause the prediction to be inaccurate, whereas inference attacks collect information about

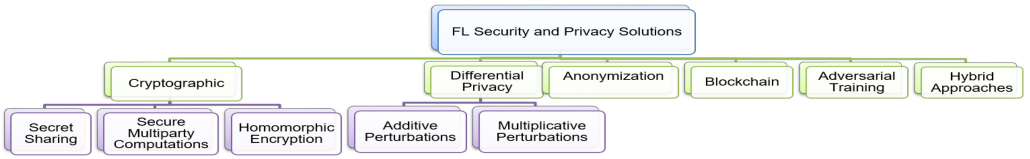

Fig. 5. FL security and privacy solutions.

the data distribution, verify the presence of specific examples in the training data, infer information about the participants, and extract information about the model hyperparameters.

3.6 FL Security and Privacy: Proposed Solutions

Several techniques aim to design a secure and privacy-preserved FL architecture. However, designing a secure and privacy-preserving FL while maintaining model accuracy and optimizing computational and communication resources is challenging. Figure 5 depicts a taxonomy for several solutions proposed in the literature to mitigate the privacy breaches and security threats that target FL in IoT environments. Hence, the most common approaches designed to produce secure FL can be summarized as follows.

Homomorphic encryption. **Homomorphic encryption (HE)** is a method of exchanging encrypted model parameters to protect the privacy of users and reduce the risk of data leakage. However, HE requires a trade-off between accuracy and privacy [166, 167]. Liu et al. [86] presented a privacy-enhanced FL (PEFL) framework that employed HE to defend against poisoning attacks. According to the experimental results, the presented framework effectively defends against two types of poisoning attacks: label flipping and backdoor attacks. Further, Zhang et al. [172] proposed an FL privacy preservation approach based on HE to encrypt the local gradients. Further, it employed distributed selective stochastic gradient descent to minimize the computation cost.

Secure multiparty computation. **Secure multi-party computation (SMC)** is a multi-party proof-of-zero-knowledge security mechanism. Each party has no knowledge of the other and only has direct exposure to its related data. Because of its complicated calculations, achieving zero-knowledge proof is usually impossible. Therefore, whenever the required security constraints are satisfied, the partial zero-knowledge proof is acceptable [166]. Moreover, SMC relies on cryptographic techniques for securing client updates in the FL. SMC outperforms traditional cryptographic mechanisms by encrypting only the model parameters rather than a large volume of data, making it preferable in FL applications. The main issue with SMC is the trade-off between privacy and efficiency, as SMC execution takes time and has a negative impact on FL training. As a result, lightweight SMC solutions for FL are still required [102]. In the work of Bonawitz et al. [19], Google developed a secure aggregation protocol that employs the SMC to safely aggregate the weighted average of local mode updates. In this, the authors outline that the central server is unable to obtain information about any of the clients since it only decrypts the average of updates after verifying that a sufficient number of clients are participating in the secure aggregation.

Secret sharing. Shamir [130] proposed secret sharing as a cryptographic technique that disseminates a secret among a bunch of participants. The secret can be reconstructed if and only if at least t (qualified subset) of the participants cooperate [155]. Recently, several research works have been published that employ the secret sharing scheme to improve the privacy of FL [167]. For example, Bonawitz et al. [20] and Gao et al. [46] employed a secret sharing mechanism to design FL frameworks that are able to preserve the participant's privacy. The participants in the secret sharing scheme check the consistency of secret shares to validate them and ensure that each of them follows the protocol. This behavior, however, is vulnerable to malicious client and server attacks aimed at updating the training data, local model updates, or even the global model to prevent FL

model convergence. *Byzantine attack* is the name given to such attacks. Thus, the secret sharing alone is not enough to ensure privacy and security in FL. Hence, in a verifiable FL framework proposed by Xu et al. [163], this framework employed the secret sharing and the key agreement protocol to protect a participant's privacy and verify the correctness of the aggregated model as well.

Differential privacy. DP is an approach that aims to protect the client's sensitive data from privacy leakage in FL [167]. DP adds a small amount of noise to the local model parameters to make it difficult for attackers to extract personal information about the participants. However, the DP approaches reduce the privacy leakage risk, but there is still a trade-off between the amount of added noise and the overall model accuracy [133]. Furthermore, adding more noise will prolong the model convergence time [67]. To avoid privacy leakage of FL in IoV environments, Wei et al. [154] proposed a **local differential privacy (LDP)** mechanism. This LDP mechanism reduced communication costs while preventing adversaries from recreating exact training data from vehicle gradients. Experiments confirmed that there is a trade-off between convergence performance and privacy level. Furthermore, increasing the number of participating clients helps improve convergence performance if the level of privacy protection and the number of aggregation times are both fixed. Compared to SMC, DP does not have any extra communication costs. As a result, DP is more efficient than SMC [86], but DP has a negative effect on the global model accuracy [40].

In general, there are two categories of DP: global DP and local DP. The central server applies global DP after aggregating the local model updates received from participants. As a result, when malicious participants obtain the global model, they can not easily obtain any information about the other participants. Because the central server receives the original local model updates from the clients, global DP is susceptible to malicious server attacks [167], whereas LDP is a type of DP in which the random noise is applied locally by participants rather than centrally by the server. Therefore, the central server will be unable to infer any information about the participants [25, 85]. In contrast to global DP, local DP provides a strong guarantee of privacy [167]. The DP techniques can be applied in additive perturbation and multiplicative perturbation. Additive perturbation adds random noise to weight or gradient updates, whereas multiplicative perturbation transforms the data into a different space and is more effective than additive perturbation [65, 167].

Anonymization. Anonymization techniques are used to protect privacy by removing personally identifiable and sensitive data while maintaining data utility. Three techniques are commonly used to achieve the data anonymization: k-anonymity, l-diversity, and t-closeness [81, 167]. These techniques have been applied in recent peer-reviewed research. For example, Choudhury et al. [33] and Song et al. [138] used anonymization techniques to protect FL privacy. It is experimentally reported that the anonymization techniques promote FL privacy while achieving higher model performance and privacy levels than the DP approaches.

Blockchain. Blockchain is a distributed ledger technology characterized by immutability, transparency, reliability, trustworthiness, auditability, and accountability features. These promoting features make the blockchain a more convenient for defending against FL attacks. Due to its security and traceability features, blockchain is an excellent choice for serving as a decentralized coordinator in FL [133]. Short et al. [133] proposed an algorithm to protect FL from model poisoning attacks. The algorithm is implemented and run in the smart contract on the blockchain. Unlike other algorithms that use data sample size [71, 156] or reputation [64] to verify the quality of local updates, this algorithm uses accuracy as a metric to verify the accuracy of local updates before aggregating the global model. Similarly, Qi et al. [120] introduced a blockchain-enabled FL framework for securing the control of urban traffic flow. The authors revealed that blockchain has been used to allow decentralized FL, in which model updates can be verified by miners to avoid fraudulent updates and so mitigate the impact of data poisoning attacks. In addition, DP has been used to prevent inference attacks and safeguard vehicles' privacy.

For IoT big data analysis, Unal et al. [145] presented a secure blockchain-based FL framework. This framework can protect against the model and data poisoning attacks by combining the blockchain with a one-way fuzzy function. Besides this, the authors concluded that combining blockchain and FL can provide a secure and privacy-preserving environment for IoT data analysis. Moreover, Sun et al. [141] designed a reliable and secure FL platform by combining FL and permissioned Hyperledger Fabric. Local updates from various participants are encrypted and stored in the blockchain using homomorphic threshold encryption. Blockchain is often considered an ideal solution for data security, data validation, and guaranteeing model parameter consistency among untrustworthy participants in FL due to its decentralized nature, traceability, and tamper resistance [141]. However, the combination of blockchain and FL introduces new privacy and security concerns.

Adversarial training. By injecting adversarial samples into the training data, adversaries can perform evasion attacks. The goal of these perturbed samples is to reduce the FL model's effectiveness. Hence, adversarial training is proposed to train FL models with adversarial samples to defend against such attacks. By adversarial training, the FL model will be robust enough to distinguish the known adversarial attacks [170, 173].

Hybrid approaches. Due to the data utility limitations, communication overhead, and computation costs of DP and cryptographic techniques, hybrid approaches for FL privacy preservation have recently been proposed [167]. Hybrid approaches attempt to combine the benefits of multiple techniques to promote FL security and privacy while avoiding the drawbacks of such techniques. For example, Hao et al. [49, 50] used HE with DP to preserve privacy while avoiding data utility issues and computational cost. In addition, Truex et al. [144] and Xu et al. [164] combined SMC and DP for FL privacy preservation. Furthermore, Sun et al. [141] leveraged permissioned blockchain and HE to design a secure FL architecture.

4 INTERNET OF THINGS

IoT significantly influences our daily lives. Education, manufacturing, transportation, entertainment, sport, individual homes, and complete cities are a few uses of IoT applications. Furthermore, the pervasiveness of IoT simplifies some daily activities, enriches people's interactions with their environment, and improves social interactions with people and objects. Actuators, sensors, mobile phones, and other IoT devices collect and transmit data using embedded communication technologies like Bluetooth, WiFi, ZigBee, Lora, and GSM. These technologies allow IoT devices to send and receive data and commands from other devices or the control servers, resulting in direct interaction between IoT devices and their surroundings without human intervention. Moreover, IoT has been employed in many industrial applications ranging from transportation and smart grid applications and energy distribution [8, 94]. The details about the most common IoT architectures are discussed in the supplementary material included with this survey.

4.1 IoT Security and Privacy Concerns

IoT applications face many security and privacy risks, as IoT devices are resource constrained, geographically distributed, and heterogeneous. Moreover, the complex and layered architecture of IoT results in a wide range of attacks that target each layer of this architecture. Accordingly, using traditional cybersecurity techniques and protocols and publishing software updates for billions of connected IoT devices is difficult. Security threats for IoT applications can be defined as the threats that target data integrity of the network availability of IoT applications, such as denial-of-service attacks and their variations, **man-in-the-middle (MiTM)** attacks, and malware. However, privacy threats for IoT applications are those threats that target data confidentiality in IoT applications, such as man-in-middle attacks and data privacy attacks. The most common security and privacy

threats that target IoT applications are discussed by Asharf et al. [12] and Waheed et al. [148], and we can summarize them as follows.

Denial of service. Denial of service is a common security attack in IoT applications due to their weak security features. It congests the network with invalid requests, which consume the network resources and make IoT services unavailable for genuine users. There is an enhanced version of DoS codenamed *DDoS*. In DDoS, multiple sources of attack target IoT applications, making it difficult to detect and prevent the source of the attack. Moreover, DDoS has variations such as SYN flooding, internet Control Message Protocol (ICMP), crossfire attack, User Datagram Protocol (UDP), and Botnet attack. All of the DDoS variations have the same objective as a DDoS attack.

Man in the middle. An MiTM attack occurs when the attacker puts himself in the middle of communication between two communicating parties to spoof, eavesdrop on, or impersonate one of the parties. It can be classified into active MiTM attack and passive MiTM attack. An active MiTM attack observes and collects the data exchanges between the two parties, but it does not alter the exchanged data. Eavesdropping and sniffing attacks are examples of an active MiTM attack. However, an passive MiTM attack includes data abuse by impersonating one or more communicating parties to others or getting authorized access to one or more users. Impersonation and authorization are examples of a passive MiTM attack.

Malware. Malware is a malicious software that has many forms, including viruses, spyware, worms, Trojan horses, rootkits, and malvertising. In IoT, malware aims to install malicious software on IoT devices to perform malicious activities. For example, Mirai malware affected approximately 1.2 million IoT devices in 2016 and attacked several famous internet businesses such as Google and Amazon. Most of the malware used today is produced by either copying the source code and following the instructions provided on the internet or by using a variant of the same harmful code written by the person who created the malware [107].

Data privacy. The objective of a data privacy attack is to infer sensitive information about the users or reveal their identity. Therefore, there are two categories of data privacy attacks: active data privacy attack and passive data privacy attack. An active data privacy attack includes attacks that alter data in IoT applications, such as tampering attacks. However, passive data privacy attack include attacks that leak the data and re-identify the identity of the users, such as re-identification (inference attacks). Inference attacks are based on de-anonymization and aggregate information about the users to extract sensitive information about them and reveal their identities.

As the number of IoT devices grows, so does the number and severity of security threats and privacy breaches [51]. These devices are considered the primary source of big data in IoT applications. Hence, IoT data is distinguished by its volume, velocity, and correlation to time and location. Therefore, intelligent data analysis plays a pivotal role in IoT applications because it enables them to discover new insights, make control decisions, and predict future insights. Therefore, they provide better services to users in an intelligent manner [92]. It is essential to use the revolution of AI, especially the state-of-the-art ML techniques codenamed *deep learning* (DL), which contributes to smartly analyzing IoT data and consequently enhancing the effectiveness of IoT applications [99].

Notably, IoT applications often deal with sensitive data and are linked to critical infrastructure, making them susceptible to various security and privacy risks [108]. Therefore, IoT data should be processed in a low-latency, secure, privacy-preserving, and reliable manner to enable effective and intelligent IoT applications. Therefore, traditional centralized ML approaches that collect and share IoT data from various devices to a centralized server are no longer suitable. There are several reasons for this: central servers have a single point of failure, are vulnerable to a wide range of security threats, and can leak personal information. Furthermore, as the volume of collected data grows, they require more computational and communication resources [76]. FL is a distributed collaborative AI method that allows the development of decentralized intelligent and privacy-enhanced

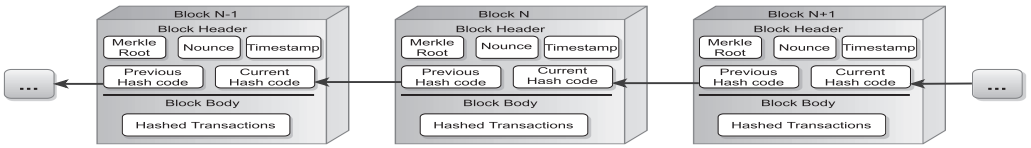

Fig. 6. Structure of a block in the blockchain.

IoT applications. In addition, FL provides a paradigm that helps increase IoT networks' communication capacity by transmitting only trained models rather than raw data. However, FL is still susceptible to client- and server-side security and privacy issues. For example, malicious clients may transmit malicious parameters to the aggregator or provide false data to train the machine learning model. In addition, the aggregator may be attacked to impair the training process and obtain sensitive information about participating customers. The aggregator itself also represents a single point of failure and may compromise the privacy of participating clients by extracting sensitive information about them [108].

Parallel to FL, blockchain is a decentralized technology that has the potential to significantly enhance the security and privacy of IoT applications because of its remarkable characteristics such as immutability, traceability, decentralization, and non-repudiation. Accordingly, blockchain facilitates the development of decentralized applications through smart contracts. Consequently, various risks associated with centralized applications, such as single point of failure, data loss, data corruption, and human mistakes, are reduced [119]. Recently, blockchain-based FL approaches have been proposed to enable decentralized, secure, and privacy-preserving IoT data analytics that address some of the key issues associated with centralized IoT data analytics. These approaches, however, create a new surface of security threats and privacy leakage that must be investigated and addressed.

5 BLOCKCHAIN

Blockchain is one of the most popular disruptive technologies adopted in many financial and industrial applications. It is defined as a distributed and immutable ledger of blocks distributed among untrusted parties in a P2P network without any trusted centralized third party [37]. Consensus algorithms validate and verify all transactions before storing them in the blocks. As depicted in Figure 6, each block holds the hash of its previous block in the chain, resulting in the block immutability [87]. To ensure data integrity, each party in the network stores the same copy of the ledger [143]. Whenever a new transaction is generated, it is broadcasted to specific nodes in the network codenamed *miners*. After miners validate the received transaction by verifying the associated signature, they create a new block and broadcast it throughout the network by solving the consensus problem in a distributed manner [151]. Once miners verify the new block, it is added to the distributed ledger [101].

Many IoT devices communicate with a central server in the traditional IoT system architecture. The server is responsible for controlling those devices and storing and analyzing the large amounts of data they generate. Unfortunately, these architectures are ineffective due to their reliability, availability, security, and privacy issues [3]. However, blockchain has remarkable characteristics for solving security issues in IoT, especially data integrity and reliability issues [3, 151]. Therefore, the integration of blockchain technology in IoT applications results in many benefits for them as follows [157].

Removal of the central authority. Decentralization is a key feature of blockchain since it allows for the distributed exchange, processing, storage, and verification of transactions. Asymmetric cryptography is used to establish confidence between nodes in the blockchain network, eliminating the

need for a central authority. Thus, the blockchain nodes securely exchange the transaction, and the failure of any node does not impact the whole blockchain network, ensuring the blockchain's robustness and reliability. Accordingly, adopting a decentralized blockchain ledger in an IoT network eliminates the single point of failure and improves scalability and fault tolerance. In addition, the P2P nature of blockchain networks accelerates the messages exchange between peers in IoT applications while storing the state of devices in the distributed ledger.

Trustworthiness. In the blockchain, asymmetric cryptography safeguards the interchange of transactions between nodes, thus maintaining the integrity of the modification and storage of assets in the distributed ledger. Moreover, the consensus algorithms ensure the consistency of transactions before recording them in the distributed ledger. Consensus algorithms thereby assure the trustworthiness of transactions across blockchain nodes. As blockchain provides data trust by consensus mechanisms, this will help IoT devices ensure that the stored data is trusted and tamper proof.

Enhance data privacy and security. Each block is hashed and included in all successive blocks in the chain. Thus, it is difficult to update the data stored in the blockchain unless the adversaries have the capability to update the hashes stored in all successive blocks. To ensure that no node other than the node with the private key can access the data, the blockchain uses private-public key pairs. Furthermore, smart contracts can be employed to define a set of rules and conditions that must be satisfied to access the data.

Maintaining data integrity. It is extremely difficult to change the data contained in a blockchain distributed ledger, and any effort to change or falsify such data will be caught by the consensus processes. As a result, blockchain enables IoT designs to ensure data integrity by avoiding attacks that damage stored data. However, it is critical to have a method that detects malicious nodes that attempt to push invalid transactions into the blockchain network.

It is worth mentioning that the key features of the blockchain are discussed in the supplementary material included with this survey.

5.1 Blockchain Architecture

The blockchain architecture is decoupled into six stacked layers. These layers are the data layer, network layer, consensus layer, incentive layer, and the application layer, as shown in Figure 7 [87, 160]. Each of these are discussed next.

Data layer. The data layer involves the techniques that are required for storing transaction records generated from various data sources. Each block in the chain stores a set of timestamped, verified, and hashed transactions in the form of a Merkle tree data structure. Furthermore, a block comprises two parts, as depicted in Figure 6. The first part is the block header that stores metadata about the block, such as block version, block timestamp, Merkle tree root, and a hash of its previous block. The second part of the block includes the transactional records. The chain is formed by linking each block with the hash of its previous block, called the *parent block* [87, 168].

Network layer. The network layer includes distributed network, communication, and verification mechanisms. In general, nodes in the blockchain network interact with each other in a P2P decentralized manner. To this end, the network layer in blockchain helps in broadcasting, forwarding, and verifying transactions among numerous nodes. When a node in the network generates a transaction, the node signs it with its private key and broadcasts this transaction to its neighbors. Once the neighbors receive the transaction, they verify it with its public key. If the transaction is valid, it will be added to the block and broadcast to the other nodes. Otherwise, the transaction will be discarded [87, 160, 168].

Consensus layer. In the blockchain, consensus algorithms are critical, as they provide ledger integrity, security, and efficiency among the untrusted nodes in the P2P network. The consensus

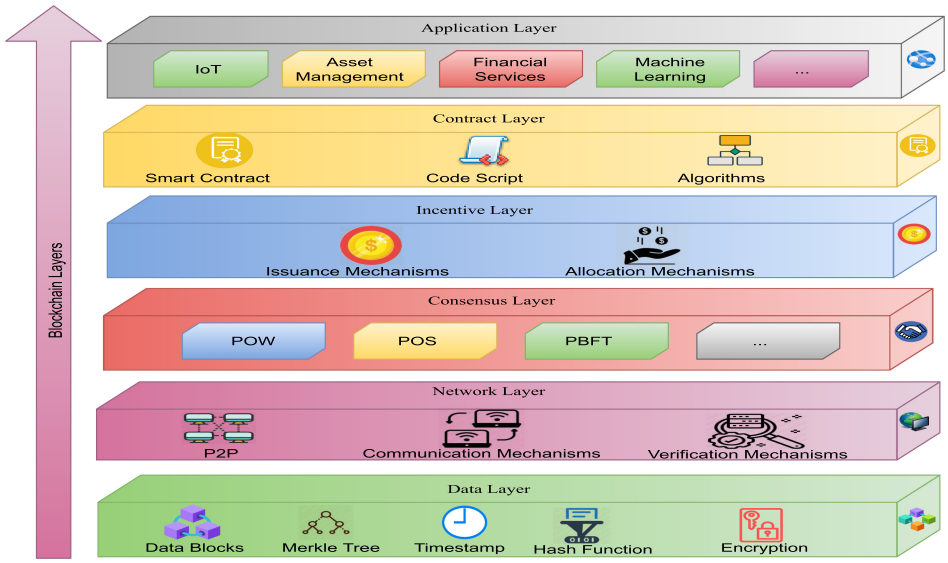

Fig. 7. Systematic blockchain architecture.

algorithms' primary goal is to have all nodes agree on adding a new block to the ledger. Various consensus algorithms are applied in the blockchain systems, each of which has its advantages and disadvantages. Some of the more well known of these are discussed in this survey [87, 168].

Incentive layer. The incentive layer is a fundamental component that is responsible for economic incentives in blockchain systems, especially public blockchains. This layer is therefore required, as blockchains must include economic factors such as economic incentive issuance and allocation mechanisms to issue and distribute some economic incentives (e.g., digital currencies) [160]. The economic incentives that are given to specific nodes in the blockchain network encourage nodes to contribute to verifying new blocks and keep the decentralization nature of the blockchain network [87].

Contract layer. The contract layer includes a set of script codes, algorithms, and smart contracts embedded in the blockchain to extend the logic of transactions and add additional complex business rules. These rules are automatically performed when a set of pre-determined conditions are met and agreed upon by nodes [87, 160, 168].

Application layer. The application layer includes a wide range of applications that benefit from the powerful features of the blockchain. The application layer is where the blockchain provides benefits. There are many examples of the blockchain application layer in academia and industry, including IoT, AI, security systems, cloud computing, and financial systems. For example, digital currency transactions represent the application layer in the Bitcoin system, whereas in other implementations, such as in Ethereum and the Hyperledger Fabric platforms, the application layer involves both currency transactions and decentralized applications (Dapp) [87, 150, 160, 168].

5.2 Consensus Algorithms Used in Blockchain

In the blockchain, consensus algorithms ensure ledger consistency, safety, and efficiency among different untrusted nodes. The main aim behind the consensus algorithms is to agree between all nodes on adding a new block to the ledger. The consensus algorithms for blockchain are divided into two groups [7]. The first one is proof-based consensus algorithms, where nodes that intend to

contribute to the consensus process should show their ability to do extra work. They have more computational power to do that work than others. The second group is known as voting-based consensus algorithms, in which the final consensus decision is taken based on the majority of nodes' verification decisions. Therefore, each node should share its own verification decision with others before the final consensus decision is taken [9, 111]. Further, each one of the consensus algorithms is suitable for specific network scenarios [115]. The most popular consensus mechanisms are explained next.

Proof of work. **Proof of work (PoW)** is the first proposed consensus algorithm for Bitcoin. In PoW, nodes with sufficient computational power (miners) compete to solve a mathematical problem. The first node that solved the mathematical problem can create the next block and receive the reward. Moreover, this node is responsible for broadcasting its result to all of its peers in the network [9, 27, 97]. Despite the ability of PoW to make an agreement between all nodes about adding the new block, it consumes extensive power and computational resources. It is still vulnerable to Sybil and denial-of-service attacks [135].

Proof of stake. **Proof of stake (PoS)** is a consensus mechanism that depends on the economic stake as proof to select the validator of the next block. It is proposed to overcome the issues of PoW, especially high energy consumption and intensive computational power. Thus, PoS is energy efficient and consumes less computational resources than PoW. The validator of the next block is selected randomly based on a combination of many factors such as coinage and the available amount of coins. Once the validator node is determined, it validates the transactions inside the block, signs the block, and adds it to the chain. In addition, the validator receives the transaction fees as a reward [7, 117]. It is difficult to attack the PoS because the attackers would need enough coins and wait for a potentially long time before being able to perform an attack [27, 97].

Practical Byzantine fault tolerance. **Practical Byzantine fault tolerance (PBFT)** is an efficient voting-based consensus mechanism that consumes less computational power than the other consensus mechanisms [135]. It has been proposed as a way to ensure the distributed system consistency when it has Byzantine failure nodes, especially for private blockchains. PBFT consists of three protocols: consistency, view-change, and the checkpoint protocol. The consistency protocol is used to ensure the integrity and consistency of transactions and blocks. Where there is one node called a *primary node*, it is responsible for generating the block and broadcasting it to the other nodes in the network, known as replica nodes. The view-change protocol mainly aims to replace the primary node when it has a failure to guarantee the system stability. Finally, the checkpoint protocol clears the consensus certification messages of the node to save the memory and avoid system faults due to accumulated node inconsistency [147, 152].

Proof of capacity. **Proof of capacity (PoC)** chooses miners with substantial storage capacity. PoC is similar to PoW, which selects miners with higher computational resources. However, PoC is more energy efficient than PoW. The more a miner can store in terms of plots (datasets), the more likely it is to be picked to mine the next block. PoC is not a viable consensus mechanism for IoT because of its latency (block generation takes approximately 4 minutes in the current implementations) [127].

Proof of elapsed time. **Proof of elapsed time (PoET)** is developed by Intel and exhibits the same behavior as POW but consumes less energy than PoW. In PoET, miners compete to solve a hash problem, and the winner is chosen at random based on a random wait period. The winner miner should have the shortest wait time among competitors. Furthermore, a Trusted Execution Environment (TEE) validates the wait time operation. PoET has a minimal computational need, a low energy consumption, and a high throughput. These characteristics make it suited for IoT applications; nevertheless, its reliance on Intel renders it centralized, which contradicts the decentralized nature of blockchain [127].

Table 2. Summary of Blockchain Platforms

Platform	Year	Access	Smart Contract	Consensus	Script Language	Currency	Industry Focus
Bitcoin	2008	Permissionless	Simple logic	PoW	Bitcoin Script	BTC	Cryptocurrency
Ethereum	2013	Permissionless	Complex logic	PoW & PoS	Solidity	Ether	Cross-industry
IoTA	2016	Permissionless	Complex logic	PoW	Solidity	None	Cross-industry
Corda	2016	Permissioned	Complex logic	Pluggable consensus	Java or Kotlin	None	Cross-industry
Hyperledger Fabric	2016	Permissioned	Complex logic	Pluggable consensus	Go, Java, and Node.js	None	Cross-industry

5.3 Blockchain Platforms

There are currently several competing blockchain platforms available in the market. However, most platforms are still immature and do not offer long-term support for different versions. Each blockchain platform has individual capabilities, and some are designed for specific domain requirements. For instance, Bitcoin is designed for cryptocurrency and financial applications. Thus, selecting the most appropriate blockchain platform for a particular application is a challenging task dependent on the application requirements. To simplify selecting a suitable blockchain platform based on the pre-defined application requirements, Nanayakkara et al. [104] proposed a methodology for comparing and selecting the suitable and promising blockchain platform that provides reasonable solutions for adopting blockchain in the industry. This study compared 24 blockchain platforms actively used in various industrial applications as of August 2020. Table 2 summarizes and compares the most common blockchain platforms.

Bitcoin. Bitcoin is considered the first cryptocurrency that allows the development of basic smart contracts. In Bitcoin, the basic smart contracts assisted in validating Bitcoin transactions based on the fulfilment of preset requirements. However, it is infeasible to develop smart contracts with sophisticated logic in Bitcoin due to the restricted capabilities of Bitcon's script language, which only supports simple arithmetic, logical, and cryptographic operations [103, 149].

Ethereum. Ethereum is one of the most popular public blockchain platforms for creating sophisticated smart contracts. Bitcoin depends on the Unspent Transaction Output (UTXO) paradigm, whereas Ethereum is based on the idea of accounts and supports two types of accounts: externally owned accounts and contract accounts. The first kind is governed by its private key, whereas the second is by the contract code. Users with an externally owned account can start transactions that involve both the payload (binary data) and Ether. When other users receive transactions, the smart contract is generated if the recipient does not have any accounts and is executed on local EVM if the recipient has a contract account. Then, these transactions are broadcast to the blockchain network, where miners can validate them [23, 149].

IoTA. IoTA was launched in 2016 to support IoT applications with blockchain solutions. IoTA heavily depends on Tangle technology, which makes distributed immutable ledger blockchains that do not have a chain, blocks, or fees. In addition, IoTA does not use a chain as bitcoin does; instead, it uses a **directed acyclic graph (DAG)**. Tangle is a technology to build a distributed ledger that saves time and computer resources. It uses a DAG, which links each transaction to the two that came before it. Each transaction is like a block in the tangle, and the PoW consensus algorithm is used to add transactions to the DAG [58].

Corda. R3 (a consortium of the world's biggest leading financial institutions) has developed Corda, an open source blockchain platform. It was originally designed for use in the financial sector but has since found its way into various other sectors, including healthcare, supply chain management, and government agencies. Corda is permissioned, relies on pluggable consensus methods, and offers smart contracts that may be developed in Java or Kotlin. Corda is not ideal for IoT applications since it was created primarily for financial purposes, which has led to decisions that are not optimized for these environments [123, 127].

Hyperledger Fabric. Hyperledger Fabric is an open source and permissioned blockchain platform hosted by the Linux Foundation and aimed toward business. Because it provides pluggable consensus mechanisms, the fabric may be tailored to a wide range of industrial applications [10, 59, 149]. In contrast to Bitcoin and Ethereum, there is no cryptocurrency used in Hyperledger. Furthermore, only approved members can access the network where the transactions are regulated by smart contracts (chaincode) [59]. The PBFT is the consensus mechanism used to validate transactions and produce blocks in Hyperledger Fabric [24, 105]. In addition, Hyperledger Fabric introduced the channel concept, which allows a group of participants to create their ledger of transactions [109]. In the work of Nanayakkara et al. [104], Hyperledger Fabric was selected as the best suitable platform for developing blockchain-based business applications.

Custom blockchain. Numerous researchers choose to construct custom blockchains for usage in a variety of applications, especially those that leverage the blockchain to exchange and verify the local model parameters of FL in IoT applications. Numerous researchers choose custom blockchain implementations due to their flexibility, extensibility, and programmability [79]. For example, Salim et al. [126] proposed a differentially private blockchain-based explainable FL (DP-BFL) architecture for safeguarding IoT-based Social Media 3.0 networks. The blockchain used by this framework is a Python-implemented custom blockchain.

5.4 Blockchain-Based Smart Contracts

A trusted third party must carry out conventional contracts in a centralized manner, So they consume extensive time and add additional costs [178]. In contrast, smart contracts were introduced by Nick Szabo in 1994 [52]. These smart contracts are decentralized, self-enforcing, and self-executing programs that incorporate pre-programmed conditions and event flows. These programs run automatically when pre-defined conditions are met through an agreement between blockchain participants. For example, these conditions can be a specific balance value or a specific timestamp. Therefore, smart contracts essentially attempt to enhance and extend the blockchain network while eliminating the centralized third party and assuring service availability. The code scripts within smart contracts are immutable; hence, they are tamper proof and can not be updated unless the blockchain nodes reach an agreement. Consequently, smart contracts are trusted by all nodes in the blockchain. Moreover, smart contracts are leveraged in a variety of applications, including IoT, financial applications, e-voting, and e-commerce [52, 178].

Many existing blockchain platforms have simple interfaces to make it easier for developers to create smart contract applications. Blockchain platforms that support smart contracts are Ethereum, Hyperledger Fabric, Corda, Stellar, and Rootstock [178]. Hewa et al. [52] studied how smart contracts and blockchain technology may be applied in various industries. Furthermore, they demonstrated the technical aspects of blockchain-based smart contracts and the future research path. Meanwhile, Zheng et al. [178] reviewed the challenges of smart contracts and the recent advances to overcome such challenges. Moreover, the authors reported that the combination of software engineering, natural language processing, and AI might be introduced, promising solutions for smart contract challenges.

6 BLOCKCHAIN-ENABLED FL FOR IOT ECOSYSTEMS

As time goes by, blockchain technology has proved to be a complementary technology to FL due to its promising features such as being tamper-proof and decentralization. Thus, blockchain-based FL solutions are widely employed in a wide range of IIoT, IoT, and IoV applications to enhance their robustness and privacy protection. With its promising features, blockchain introduces reliable and secure solutions for edge computing with untrusted parties [122]. However, the key feature of FL is

to protect the user's privacy [175]. Hence, integrating blockchain and FL technologies can improve the security and efficiency of distributed intelligent systems such as IoT [121].

The integration of blockchain and FL technologies has recently inspired researchers to offer solutions with increased security, retained privacy, enhanced audibility, high accountability, and facilitated incentive mechanisms. Furthermore, smart contracts have the potential to play a critical role in the FL process coordination. These contracts can validate node contributions, compute the global model, record node performance on the ledger, and provide node incentives depending on performance [134]. Jin et al. [62] proposed a blockchain-enabled FL system (FL-Block) to address issues associated with fog computing's privacy leakage, efficiency, and poisoning attacks. FL-Block aims to protect privacy by integrating FL and blockchain to learn and save the model updates while using the distributed hash table (DHT) to save data locally on each device. In addition, the authors reported that FL-Block helps in improving FL learning efficiency. However, there is still a need to optimize the trade-off between efficiency and privacy protection.

Qu et al. [122] proposed cross-blockchain-enabled federated learning (CFL). Which aims to overcome the data sparsity issue on IoMT applications. To do this, multiple blockchain-based FL clusters are employed to aggregate model updates. Afterward, the aggregated updates are exchanged between clusters to enrich devices with sparse training data and improve efficiency. In contrast, the CFL demands heavy computational and communication resources, so the authors suggest combining it with edge computing instead of end devices. Zhang et al. [175] introduced the FTL scheme that is based on blockchain and edge computing to improve system security and performance in IIoT. The central concept of using edge computing is to solve the problem of resource-constrained IIoT devices. Furthermore, blockchain technology can be employed to ensure the security of data sent by IIoT end devices to the edge computing servers. Qin et al. [121] integrated blockchain and FL technologies to build a privacy-preserving framework for Marine Internet of Things (MIoT) on edge computing. This framework depends on FL to ensure privacy, whereas the blockchain is used as a decentralized ledger that stores local model updates from various end devices. In addition, the authors designed a new mechanism known as proof of quality (PoQ) to select the more qualified nodes that will participate in FL and be involved in the blockchain network.

Zhang et al. [176] proposed a blockchain-based federated learning approach for device failure detection in IIoT. The designed architecture depended on the edge servers to train the ML model locally, whereas the local updates were coordinated and aggregated by a central server. Blockchain is exploited to verify the clients' data integrity and manage incentives granted to the clients participating in the FL. Moreover, the authors proposed a new algorithm to reduce the impact of data heterogeneity, and this algorithm is called *centroid distance weighted federated averaging* (CDW-FedAvg). The experimental results showed that the proposed approach achieved good feasibility, accuracy, and performance. However, CDW-FedAvg has several limitations due to its dependency on a central server and inability to handle crashed and delayed clients. Otoum et al. [115] proposed a trusted and secure framework powered by the integration of FL and blockchain. This framework used reinforcement learning to train local models on end devices and then the updates were maintained in the blockchain. Only trusted devices can be allowed to add updates to the blockchain. Next, the local updates were verified by miners, and the FedAvg approach was then used to aggregate the global model on each end device.

Short et al. [133] proposed an algorithm for discarding unreliable model updates in blockchain-based FL systems. Therefore, this algorithm aimed to protect the global model against model poisoning attacks. In this algorithm, the coordinator receives the local model updates from each device and verifies the weights using the pre-defined validation set. They are verified if the local model updates increase the overall model accuracy. Otherwise, they are discarded. The authors note that this algorithm is included in a smart contract. However, the proposed algorithm reduces the

effect of poisoning attacks in FL, with multiple limitations. Unal et al. [145] integrated blockchain and FL to train a distributed ML model for IoT data analysis while preventing model poisoning attacks. The authors used fuzzy hashing to detect deviations and abnormalities in FL-trained models to prevent model poisoning attempts. Otoum et al. [115] integrated reinforcement learning with blockchain-enabled FL to ensure network trustworthiness and security via a Q-learn system. Q-learn runs on end devices to support the FL process. Only trusted devices can add their local updates to the blockchain. Therefore, the trusted devices are selected based on a proposed trust scoring mechanism. Short et al. [134] presented a theoretical approach that improves the trust and security of the FL process using private blockchain networks and smart contracts together. In this approach, smart contracts are critical to implement and run verification algorithms to verify model updates and give rewards.

Zhao et al. [177] designed a privacy-preserving blockchain-based FL system to help home appliance manufacturers. The system is based on two phases. In the first phase, the initial model is trained by the customers' mobile phones. Once the customers finish the training, they apply DP on their model updates, sign them, and send them to the blockchain. In the second phase, a set of customers are selected as miners to aggregate the global model and upload the aggregated model to the blockchain. Due to the limited size of blocks in the blockchain, the authors suggest using IPFS (the InterPlanetary File System) for off-chain storage. Hence, all local and aggregated global models are stored in IPFS, and their hash is saved in the blockchain. Furthermore, a new normalization technique is proposed to be used instead of the default batch normalization during the model training. The authors reported that this technique improves the model test accuracy, especially when the DP is used. Finally, a new incentive mechanism is proposed to reward customers who participate honestly in FL.

Otoum et al. [116] introduced a blockchain-enabled FL solution to ensure security and privacy in vehicular networks. The proposed framework allows the end device to train ML models locally without sharing data with the edge server. Furthermore, the authors adopted PBFT to ensure the trustworthiness of model training. The authors reported that the proposed framework achieved high accuracy, low energy consumption, long lifetime rate, high throughput, and low latency.

To enhance data security and model sharing in IoT, a blockchain-enabled FL scheme has been presented by Jia et al. [61]. The authors employed K-means, random forest, AdaBoost, and HE to allow multiple data protection. Based on the empirical results, the authors emphasized how the suggested approach helps to enable secure data sharing and model training in IIoT contexts. A **blockchain-based asynchronous federated learning (BAFL)** framework was proposed by Feng et al. [42]. The main goal behind the design of BAFL is to ensure the security and efficiency of FL. The blockchain distributed ledger is employed to store local model updates and reward participating parties for achieving such a goal. Accordingly, a novel entropy weight method is designed to rank the participated parties. Furthermore, the BAFL optimized the energy consumption and the block generation rate. Doku and Rawat [38] proposed a framework called *iFLBC edge*. This framework aims to make ML and intelligent services closer to the data sources (end devices). To achieve this purpose, blockchain technology and FL are collaboratively employed. Further, the framework mainly depended on the idea of Interest Group. The devices with relevant datasets will be members of the same group and collaboratively share the global model training. The proof of common interest (PoCI) mechanism determines the relevant data members.

Specifically, FL mitigates the privacy challenges in IoT environments, and the blockchain enables decentralized, reliable, and fault-tolerant FL for enabling decentralized and secure intelligence in IoT applications. Blockchain allows the secure record of the contribution degree of each IoT device, and smart contracts can be used for evaluating the local models of the IoT devices before recording them in the blockchain. This improves the convergence of the global model, punishes dishonest

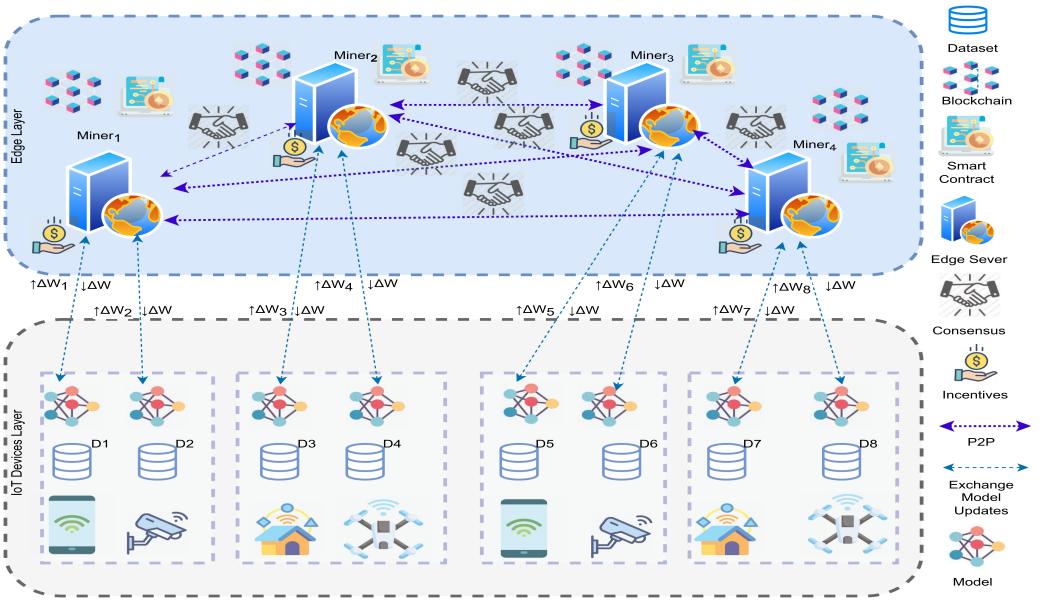

Fig. 8. The architecture for blockchain-enabled FL.

and malicious clients, and rewards honest clients based on their contribution. To indicate how the blockchain can be coupled with FL for IoT, we propose an abstracted architecture as depicted in Figure 8. This architecture consists of two different layers: the IoT devices layer and the edge layer. These are described as follows.

The IoT devices layer. The IoT devices layer consists of sensors and actuators that are responsible for regulating, monitoring, and interacting with their surroundings. These devices acquire significant amounts of data. Due to the limited resources of IoT devices, they only train the ML model and send local model updates to the associated edge servers for aggregation. The objective behind the local training on IoT devices is to preserve the security and privacy of IoT devices and mitigate the impact of adversaries who want to disclose critical information about IoT devices.

The edge server layer. The edge server layer is responsible for maintaining the distributed ledger and validating the local model updates received from the IoT devices by using smart contracts. In addition, smart contracts help authenticate the IoT devices and control the access to the blockchain. To encourage the IoT devices to participate in FL, the edge servers can adopt an incentive mechanism to reward the devices that provide higher contribution degrees. It is worth noting that each device's contribution degree and performance are also stored in the blockchain. This data assists the edge server in detecting potentially malicious, lazy, and straggler devices. Initially, the genesis block contains metadata about the IoT devices, the ML model's structure, the ML model, and the hyperparameters of the ML model, such as learning rate, number of global rounds, number of local epochs, and threshold of FL accuracy. Once the IoT devices receive the initial model, they start to train the model locally on their local datasets. Afterward, each device sends the model updates to the associated server, which authenticate the device and validate the model updates using smart contracts. The edge server then exchanges the model updates with other edge servers in a P2P manner. Consequently, each edge server starts to organize the model updates in a new block and perform the consensus process. When the edge servers (miners) reach an agreement about the new block, it will be added to the chain. This process will continue until the specified accuracy threshold is reached or the global rounds are completed.

7 LESSONS LEARNED

This section outlines the lessons learned and the significant factors to consider when designing and developing blockchain-enabled FL for IoT. Therefore, the lessons learned can be summarized as follows:

- The traditional FL mechanism has various limitations. One of these limits is the reliability of local model updates, which might be manipulated by malicious devices to have a detrimental effect on the global model. Furthermore, any adversary modifications to the central server's global model will negatively influence all subsequent training rounds. Another constraint is the lag effect, also known as synchronous FL, in which the completion duration of each training round is proportional to the speed of the slowest device. As a result, this behavior will impact the efficiency of high computational devices, as they will be unable to submit their local model updates until all of the other devices have completed the local training.
- Aggregation algorithms are a critical component of FL. They are responsibly aggregating the global model. There have recently been several aggregation algorithms proposed to improve the FL process. However, it is still necessary to design an aggregation algorithm that takes into account data heterogeneity, system heterogeneity, and participant personalization while optimizing communication and computational costs.
- IoT devices generate enormous amounts of data. The limitations of classic centralized machine learning approaches associated with security, privacy, reliability, and scalability make them unsuitable for IoT applications. To address these concerns, blockchain-based FL approaches are presently being offered as a way to provide decentralized, secure, and privacy-preserving data analytics in an IoT environment. However, these approaches are vulnerable to security risks and data leaks, which must be studied and resolved in future works.
- To ensure the quality and reliability of the global model, it is vital to verify the local model updates before the global model aggregation. In other words, the verification mechanism assists in aggregating only the valid model updates from honest participants while discarding the untrustworthy model updates from malicious participants. It is worth noting that smart contracts can be vitally involved in designing and running the verification mechanisms in blockchain-based FL architectures. According to the literature, designing and implementing effective and optimized verification mechanisms is an open research direction.
- The adoption of blockchain technology in IoT applications enhanced many issues such as security, data privacy, reliability, and availability. However, this adoption results in new issues that must be tackled. Many of these issues arise because the blockchain technology was originally designed to be deployed and operated on powerful computers, not IoT devices with limited resources.
- IoT devices are resource constrained. On the contrary, blockchain technology requires more storage, computational, and energy resources for block verification, storage, and consensus. Currently, additional research efforts are made to develop novel consensus methods appropriate for IoT systems. Moreover, the distributed ledger requires considerable storage resources. For example, the Bitcoin blockchain size neared 185 GB by September 2018. Consequently, it is impossible to store the blockchain ledger directly on IoT devices unless new lightweight blockchains are proposed.
- The usage of cloud computing, fog servers, and edge servers when integrating blockchain and IoT can alleviate the challenges associated with resource-constrained IoT devices. For instance, edge servers can act as full nodes that store the distributed ledger, mine, and validate the blocks. At the same time, IoT devices can represent lightweight nodes that initiate transactions and keep track of the blockchain by storing the hash of blockchain data. This

architecture is known as the flexibly coupled blockchain FL architecture. Consequently, more research is required to design and implement lightweight blockchain and DL models that are convenient for running on resource-constrained IoT devices.

- Public blockchains scale well with the number of participant nodes. However, it is a challenge to ensure the privacy of transactions, as the public blockchain allows any public node to join and trace the transactional history. As a result, it is easy to analyze the transactions in the distributed ledger and then get information about the identities of the other participating nodes.
- In contrast to public blockchains, private blockchain nodes can assure the privacy of the nodes, but their scalability is poor when the number of nodes increases. Recently, smart contracts have been employed to authenticate and control access to blockchain nodes, especially in IoT applications. In this way, smart contracts help guarantee the privacy of nodes.
- Even though blockchain provides improved security, privacy, and facilitates data sharing for the FL, the blockchain-based FL is still vulnerable to various attacks, such as adversarial attacks, data poisoning, and inference attacks. In addition, the blockchain stores and broadcasts the trained model parameters across all devices. Hence, all devices can download and trace the model updates stored in the blockchain. Accordingly, the devices can infer information about one another based on the blockchain's content. This is partially mitigated by encrypting the model parameters. Such actions violate privacy and degrade the reliability of the overall system, especially in a public blockchain network with no access authority for the participating devices.
- Blockchain has been combined with FL to achieve decentralization by replacing the central server with a P2P blockchain network. The blockchain nodes then complete the global model aggregation. This eliminates the single point of failure and improves FL's reliability. Furthermore, the blockchain can verify local model updates to eliminate unqualified and malicious updates, ensuring the quality of the global model. Besides this, blockchain incentivizes FL participants to encourage them to participate in the FL procedure effectively.
- Smart contracts have the potential to play a critical role in FL process coordination. Smart contracts can validate client contributions by verifying model accuracy, computing the global model, recording participant contributions in the distributed ledger, and providing clients with incentives based on the effectiveness of their models. Until now, it has been critical to investigate and design new verification algorithms that effectively verify clients' effective and honest participation in the FL process.
- Despite their critical role in blockchain-based FL architectures, smart contracts face multiple security threats that must be addressed. There is a direct relationship between smart contract attacks and bugs in smart contract code when it comes to these threats. The majority of smart contract attacks are the result of programming language issues.

8 CHALLENGES OF BLOCKCHAIN-ENABLED FEDERATED LEARNING FOR IOT APPLICATIONS

In this section, we will highlight some of the fundamental issues that are faced by blockchain-enabled FL approaches when used in IoT applications. Due to these issues, the blockchain-enabled FL is susceptible to a variety of security breaches, and the privacy of customers who participate in the ledger FL is also put at risk. In addition, we highlight several recent and future directions to consider to address those challenges.

Privacy leakage. Privacy leakage occurs when exchanging local parameter updates between peers in the network. Many works used DP to solve the privacy leakage challenge, but these all report that this solution negatively affected the training convergence and overall system performance. The privacy leakage challenge still requires additional investigation. FL privacy

preservation techniques have been increasing in recent years. However, more effective strategies still need to be developed to balance data utility, privacy, computation cost, communication cost, and security of FL. Personalization FL can reduce privacy leakage risks and inference attacks by allowing clients to train special (personalized) layers that are not included in the global model. Furthermore, hybrid privacy preservation techniques combining multiple techniques, such as HE and DP or DP with anonymization, can yield promising results. These methods, however, still demand significant computing power and suffer from a trade-off between global model effectiveness and privacy level.

As a result, effective and privacy-preserving FL techniques for constrained-resources IoT devices are required. Model effectiveness, participant privacy, computational, and communication resources of IoT devices should all be considered while developing these techniques. Moreover, data poisoning, model poisoning, inference, and evasion attacks represent a challenge in developing blockchain-enabled FL approaches. For example, the local model updates from each client are aggregated to create a new version of a global model. When adversaries inject false data into the local data of any clients, this leads to erroneous local model updates; therefore, the aggregated global model will be negatively affected. Therefore, designing mechanisms to detect and delete false data on IoT devices is vital before being involved in the local model training.

Lightweight blockchain for IoT and FL. Blockchain can be used in various applications across many fields, including the IoT. As a result, blockchain technology contributes to developing verifiable, secure, robust, traceable, and data integrity IoT applications. Despite the benefits of blockchain for IoT applications, adapting blockchain in IoT applications results in many issues, such as computational cost, power consumption, and storage due to the resource-constrained nature of IoT devices [74, 80]. According to the literature, blockchain has a significant impact on resolving some FL issues. As a result, blockchain and FL complement each other, as they share the same decentralization characteristics. Blockchain is a natural fit for FL development, as it is an inherently secure distributed system. However, using blockchain with FL may result in increases in latency when exchanging learning models. It would be preferable to create a low-latency blockchain-based FL. Therefore, combining blockchain and FL is advantageous, as blockchain is a decentralized technology that eliminates the need for a central server to aggregate the global model. There is potential that this may overcome FL's bandwidth limitations. In addition, it exchanges updates while verifying correctness to improve security [80].

From the literature, most works implementing blockchain-based FL for IoT do not consider the need for lightweight blockchain implementation, particularly for IoT resource-constrained devices. Many recent research projects have focused on making blockchain architecture more lightweight and adaptable to the needs of IoT devices. However, these works have not yet designed a comprehensive lightweight blockchain architecture that addresses and balances the limitations of IoT devices. Some research focuses on lightweight blockchain-based identity management and authentication, for example. Bouras et al. [22] and Khan et al. [70] concentrate on computational, communication, and storage costs; Mohanty et al. [100] concentrate on bandwidth, security, and trust, as well as scalability; Danzi et al. [36] concentrate on communication costs; and Doku et al. [39] aim to reduce computational costs. Finally, Bandara et al. [15] developed a lightweight blockchain for IoT applications to improve scalability and transaction processing time. It is therefore important to design and implement a lightweight blockchain architecture that considers and balances the following factors:

- Privacy and security;
- Computational, communication, and storage costs; and
- Power consumption and scalability.

Deep neural networks on resource-constrained devices. DL architectures are currently the most popular feature representation and data classification approach. Deep architectures successfully perform many computer vision, natural language, and signal processing tasks. In addition, these approaches outperform the traditional approaches, mainly based on manually designed feature descriptors such as gradient operators and filter banks [98, 112]. The machine can automatically perform feature learning and classification using the DL approaches without using handcrafted features and traditional ML techniques. In addition, the machine can automatically decide which features are more discriminative and can learn features from unlabeled data [161].

Many IoT applications utilize DL models for IoT big data analytics. DL models achieved superior results in IoT fundamental services such as image, video, and speech analysis, but those models cannot be adapted easily for IoT resource-constrained devices. The need for high performance computing and large storage resources represents challenges for adapting DL models in many IoT applications [99]. DL models are at the heart of establishing intelligent security procedures in IoT environments. They can identify intrusions, malware, and zero-day attacks with better accuracy than traditional ML models. Furthermore, they are embraced in FL, allowing for IoT devices to be trained locally to mitigate data privacy issues resulting from the data transfer to the central entity. However, DL models require more computing resources, which is incompatible with the resource-constrained nature of IoT devices. In summary, deep neural network approaches always require more computational resources and energy than the traditional ML models. Thus, it is challenging to design lightweight and efficient deep neural network models that can train and operate on resource-constrained devices.

Lazy clients. Some clients selected to participate in learning and mining processes do not actually train their local model but instead copy the uploaded parameters from other clients to save the computing resources. Such clients are known as lazy clients. Further, lazy clients add artificial noise to the stolen parameters to conceal their deceptive behaviors. As a result, they contribute to the problem of ineffective FL. In addition, this behavior reduces the effectiveness of the global model. Li et al. [78] presented a blockchain-assisted decentralized federated learning (BLADE-FL) framework. The main goal behind this framework is to improve the security and privacy of FL and mitigate the influence of lazy clients. As well, a bounded loss function was proposed to analyze the relationship between the number of created blocks and the effect of lazy clients on training efficiency. To obtain the optimal ratio of generated blocks, the loss function was optimized, which resulted in improved performance despite the presence of lazy clients. The proposed solution helped lessen the effect of lazy clients but did not eliminate the problem of lazy clients. As a result, still, more effective solutions are needed.

Stragglers. Stragglers are the clients that failed to send their local model updates to the central server at a specific period. There may be several reasons for this—the device's limited resources, network issues, or device crashes. Stragglers are responsible for the stale (out-of-date) model, which harms the global model's convergence. Therefore, when aggregating the global model, a higher level of staleness means a higher error level. At the same time, the central server waits for their updates. Hence, this leads to a delay in aggregating the global model [158]. Asynchronous and semi-synchronous FL approaches have recently been proposed to increase FL efficiency, improve global model quality, and reduce the impact of stragglers on the global model. For example, Wu et al. [158] proposed a semi-synchronous FL approach to reduce the impact of stragglers, improve FL efficiency, and leverage the straggler's models without compromising the global model's quality. Moreover, an asynchronous FL algorithm was introduced by Chen et al. [29] to speed up global model convergence while avoiding straggler performance degradation. In general, it is vital to consider the privacy and security of FL while mitigating the stragglers' effect. In addition,

more research efforts are required to speed up the global model's convergence and manipulate the stragglers' issue while keeping the FL process's security and privacy.

Statistical heterogeneity. The heterogeneous nature of IoT devices results in generating non-IID data. The traditional FL depends on SGD to train the global model. However, although the SGD achieved positive results on IID data, it is not stable when applied to non-IID data. Specifically, SGD does not work when applying DP for model parameters to improve the security and privacy of the clients. Therefore, there is a need to propose new aggregation algorithms that consider the non-IID data issue and achieve a balance between the privacy level and the model utility in FL. Recently, there have been strong efforts aiming to alleviate the effect of the non-IID data while achieving a good level of model utility and privacy. For example, Noble et al. [113] proposed an FL approach called *differential private stochastic controlled averaging for federated learning* (DP-SCAFFOLD), which focused on efficient learning from non-IID data and applying DP to protect the model parameters from adversaries. The experimental results showed that the proposed approach achieved good results, especially when applied with non-IID data. However, the authors note in the work that the proposed approach still suffers from balancing model utility and the privacy level when adding DP to the model parameters.

Similarly, Karimireddy et al. [66] showed that the FedAvg algorithm achieved tight convergence, and it suffers from the client drift issue when the data is non-IID. Therefore, the authors proposed a new approach to mitigate the effect of client drift issues. The new approach is codenamed *stochastic controlled averaging for federated learning* (SCAFFOLD). SCAFFOLD depends on the control variates to reduce the effect of client drift. The experimental results showed that the proposed approach achieved a significant convergence rate on the non-IID data. In addition, the **personalized FL (PFL)** helps enhance the convergence of the global model for heterogeneous data while simultaneously lowering the threats to the user's privacy and security. Tan et al. [142] performed an in-depth analysis of the most recent PFL and proved how the PFL could alleviate the data heterogeneity effect on the model convergence while enhancing the clients' sense of security and privacy. Similarly, Zhu et al. [179] investigated non-IID in FL. In this survey, the authors explained how non-IID data influence the convergence of the global model in FL. In addition, they analyzed the different techniques for improving the convergence of FL on non-IID data.

System heterogeneity. IoT devices may differ in terms of computational resources, power, and communication resources. This heterogeneity has a negative impact on FL, as the IoT devices with heterogenous resources do not able to transmit their trained model synchronously and some of them may fail to transmit the model parameters due to their limited energy and computational resources. Consequently, the system heterogeneity of IoT devices leads to degrading the model convergence and causes the existence of stragglers. Despite the asynchronous requirements, FL and client sampling approaches helped in reducing the impact of system heterogeneity on FL, and it is extremely important to propose new techniques that help select proficient clients capable of effectively sharing in the FL. There is also a need to propose new asynchronous mechanisms to aggregate model parameters from heterogeneous devices in IoT networks. In summary, Li et al. [83] reviewed the impact of system heterogeneity on the performance of FL and outlined the proposed approaches. The authors reported that this system heterogeneity is a challenging task in FL that needs additional research.

Unsupervised FL. Recently, supervised ML models have been used in most works that have introduced FL frameworks. Data generated by IoT devices, however, is unlabeled. Therefore, studying and investigating how to train decentralized unsupervised ML models via FL while maintaining privacy is critical. *Unsupervised FL* is the current name for this approach. Hence, unsupervised FL remains a research topic in progress [80]. Van Berlo et al. [146] developed **federated unsupervised representation learning (FURL)**, which uses of a large amount of

unlabeled data generated by IoT devices. The use of FURL significantly reduced the need for a large amount of labeled data, which is not always available in sufficient quantities in practice.

When compared to supervised and transfer learning, FURL showed promising results. The findings will pave the way for further research and studies on federated unsupervised learning, particularly in IoT environments. As much of the data generated by IoT devices is unlabeled data, researchers have devised unsupervised FL approaches for intrusion detection [165] and malware detection [124] in IoT environments. At this point, unsupervised FL becomes viable as an alternative to allow the development of unsupervised global models that can be trained on unlabeled data to find suspicious activities in the IoT environment. This will improve the client's security and privacy. But these methods still rely on old-fashioned FL aggregation algorithms like FedAvg. Hence, new aggregation algorithms need to be proposed for unsupervised FL to improve security, privacy, and the balance between the model's utility and the level of privacy, especially when adding DP to model parameters.

AI-powered smart contracts. In modern applications, static smart contracts are not able to meet security, dynamicity, and intelligence requirements because their rules are static [87]. It is anticipated that the development of ML-enabled smart contracts will result in improved application security, automation, and dynamism. As a result, ML can potentially create more complex and effective smart contracts to detect and prevent malicious activities and unlawful transactions in blockchain networks [13]. Furthermore, compared to current traditional smart contract implementations, it is expected that the AI-enabled smart contract can verify the model parameters of FL before saving it to the blockchain and effectively prevent poisoning attacks. Thus, the adequate verification of the FL model can help enable accurate incentives for participants based on honest and influential contributions.

9 CONCLUSION

This article discussed FL, blockchain, and their integration in IoT systems to secure decision making and data analytics. Blockchain-based FL architectures have recently been promising solutions for addressing the security and privacy problems associated with data analytics in IoT environments. However, these architectures create the possibility of new security, privacy, and efficiency concerns. Therefore, new enhanced architectures and algorithms should investigate and address these concerns. This survey examined blockchain and FL to show the significance of combining the two technologies to enable secure, effective, and robust IoT applications. Then, FL security and privacy threats were investigated along with the recently proposed solution for those threats. Next, this work provided comprehensive literature about blockchain-based FL solutions for IoT applications. The lessons learned from this survey were then listed. Finally, this work highlighted several potential challenges and future directions for blockchain-based FL IoT architectures, which will necessitate further research and investigation in the future.

REFERENCES

[1] Mehdi Salehi Heydar Abad, Emre Ozfatura, Deniz Gunduz, and Ozgur Ercetin. 2020. Hierarchical federated learning across heterogeneous cellular networks. In *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'20)*. IEEE, Los Alamitos, CA, 8866–8870.

[2] Ranwa Al Mallah, Godwin Badu-Marfo, and Bilal Farooq. 2021. Cybersecurity threats in connected and automated vehicles based federated learning systems. In *Proceedings of the 2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops'21)*. IEEE, Los Alamitos, CA, 13–18.

[3] Alia Al Sadawi, Mohamed S. Hassan, and Malick Ndiaye. 2021. A survey on the integration of blockchain with IoT to enhance performance and eliminate challenges. *IEEE Access* 9 (2021), 54478–54497.

[4] Omar Alfandi, Salam Khanji, Liza Ahmad, and Asad Khattak. 2021. A survey on boosting IoT security and privacy through blockchain. *Cluster Computing* 24, 1 (2021), 37–55.

[5] Mansoor Ali, Hadis Karimipour, and Muhammad Tariq. 2021. Integration of blockchain and federated learning for Internet of Things: Recent advances and future challenges. *Computers & Security* 108 (2021), 102355.
[6] Tejasvi Alladi, Vinay Chamola, Reza M. Parizi, and Kim-Kwang Raymond Choo. 2019. Blockchain applications for Industry 4.0 and Industrial IoT: A review. *IEEE Access* 7 (2019), 176935–176951.
[7] Shikah J. Alsunaidi and Fahd A. Alhaidari. 2019. A survey of consensus algorithms for blockchain technology. In *Proceedings of the 2019 International Conference on Computer and Information Sciences (ICCIS'19)*. IEEE, Los Alamitos, CA, 1–6.
[8] Mahmoud Ammar, Giovanni Russello, and Bruno Crispo. 2018. Internet of Things: A survey on the security of IoT frameworks. *Journal of Information Security and Applications* 38 (2018), 8–27.
[9] Averin Andrey and Cheskidov Petr. 2019. Review of existing consensus algorithms blockchain. In *Proceedings of the 2019 International Conference on Quality Management, Transport and Information Security, and Information Technologies (IT&QM&IS'19)*. IEEE, Los Alamitos, CA, 124–127.
[10] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, et al. 2018. Hyperledger fabric: A distributed operating system for permissioned blockchains. In *Proceedings of the 13th EuroSys Conference*. 1–15.
[11] Muhammad Asad, Ahmed Moustafa, and Chao Yu. 2020. A critical evaluation of privacy and security threats in federated learning. *Sensors* 20, 24 (2020), 7182.
[12] Javed Asharf, Nour Moustafa, Hasnat Khurshid, Essam Debie, Waqas Haider, and Abdul Wahab. 2020. A review of intrusion detection systems using machine and deep learning in Internet of Things: Challenges, solutions and future directions. *Electronics* 9, 7 (2020), 1177.
[13] Syed Badruddoja, Ram Dantu, Yanyan He, Kritagya Upadhayay, and Mark Thompson. 2021. Making smart contracts smarter. In *Proceedings of the 2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC'21)*. IEEE, Los Alamitos, CA, 1–3.
[14] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. 2020. How to backdoor federated learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*. 2938–2948.
[15] Eranga Bandara, Deepak Tosh, Peter Foytik, Sachin Shetty, Nalin Ranasinghe, and Kasun De Zoysa. 2021. Tikiri—Towards a lightweight blockchain for IoT. *Future Generation Computer Systems* 119 (2021), 154–165.
[16] Mandrita Banerjee, Junghee Lee, and Kim-Kwang Raymond Choo. 2018. A blockchain future for Internet of Things security: A position paper. *Digital Communications and Networks* 4, 3 (2018), 149–160.
[17] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. 2019. Analyzing federated learning through an adversarial lens. In *Proceedings of the International Conference on Machine Learning*. 634–643.
[18] Alberto Blanco-Justicia, Josep Domingo-Ferrer, Sergio Martínez, David Sánchez, Adrian Flanagan, and Kuan Eeik Tan. 2021. Achieving security and privacy in federated learning systems: Survey, research challenges and future directions. *Engineering Applications of Artificial Intelligence* 106 (2021), 104468.
[19] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, et al. 2019. Towards federated learning at scale: System design. *Proceedings of Machine Learning and Systems* 1 (2019), 374–388.
[20] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 1175–1191.
[21] Nader Bouacida and Prasant Mohapatra. 2021. Vulnerabilities in federated learning. *IEEE Access* 9 (2021), 63229–63249.
[22] Mohammed Amine Bouras, Qinghua Lu, Sahraoui Dhelim, and Huansheng Ning. 2021. A lightweight blockchain-based IoT identity management approach. *Future Internet* 13, 2 (2021), 24.
[23] Vitalik Buterin. 2013. Ethereum white paper. *GitHub Repository* 1 (2013), 22–23.
[24] Christian Cachin et al. 2016. Architecture of the hyperledger blockchain fabric. In *Proceedings of the Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, Vol. 310.
[25] Hui Cao, Shubo Liu, Renfang Zhao, and Xingxing Xiong. 2020. IFed: A novel federated learning framework for local differential privacy in power Internet of Things. *International Journal of Distributed Sensor Networks* 16, 5 (2020), 1550147720919698.
[26] Catalin Capota, Moritz Neun, Lyman Do, and Michael Kopp. 2019. Asynchronous federated learning for geospatial applications. In *ECML PKDD 2018 Workshops*. Communications in Computer and Information Science, Vol. 967. Springer, 21–28.
[27] Natalia Chaudhry and Muhammad Murtaza Yousaf. 2018. Consensus algorithms in blockchain: Comparative analysis, challenges and opportunities. In *Proceedings of the 2018 12th International Conference on Open Source Systems and Technologies (ICOSST'18)*. IEEE, Los Alamitos, CA, 54–63.

[28] Jin-Hua Chen, Min-Rong Chen, Guo-Qiang Zeng, and Jia-Si Weng. 2021. BDFL: A Byzantine-fault-tolerance decentralized federated learning method for autonomous vehicle. *IEEE Transactions on Vehicular Technology* 70, 9 (2021), 8639–8652.

[29] Ming Chen, Bingcheng Mao, and Tianyi Ma. 2019. Efficient and robust asynchronous federated learning with stragglers. In *Proceedings of the 2019 International Conference on Learning Representations (ICLR'19)*.

[30] Min Chen, Shiwen Mao, Yin Zhang, and Victor C. M. Leung. 2014. *Big Data: Related Technologies, Challenges and Future Prospects*. Springer Briefs in Computer Science. Springer.

[31] Mingzhe Chen, H. Vincent Poor, Walid Saad, and Shuguang Cui. 2020. Wireless communications for collaborative federated learning. *IEEE Communications Magazine* 58, 12 (2020), 48–54.

[32] Zheyi Chen, Weixian Liao, Kun Hua, Chao Lu, and Wei Yu. 2021. Towards asynchronous federated learning for heterogeneous edge-powered Internet of Things. *Digital Communications and Networks* 7, 3 (2021), 317–326.

[33] Olivia Choudhury, Aris Gkoulalas-Divanis, Theodoros Salonidis, Issa Sylla, Yoonyoung Park, Grace Hsu, and Amar Das. 2020. A syntactic approach for privacy-preserving federated learning. In *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI'20)*. 1762–1769.

[34] Li Da Xu, Yang Lu, and Ling Li. 2021. Embedding blockchain technology into IoT for security: A survey. *IEEE Internet of Things Journal* 8, 13 (2021), 10452–10473.

[35] Hong-Ning Dai, Zibin Zheng, and Yan Zhang. 2019. Blockchain for Internet of Things: A survey. *IEEE Internet of Things Journal* 6, 5 (2019), 8076–8094.

[36] Pietro Danzi, Anders E. Kalør, Čedomir Stefanović, and Petar Popovski. 2019. Delay and communication tradeoffs for blockchain systems with lightweight IoT clients. *IEEE Internet of Things Journal* 6, 2 (2019), 2354–2365.

[37] Natarajan Deepa, Quoc-Viet Pham, Dinh C. Nguyen, Sweta Bhattacharya, B. Prabadevi, Thippa Reddy Gadekallu, Praveen Kumar Reddy Maddikunta, Fang Fang, and Pubudu N. Pathirana. 2022. A survey on blockchain for big data: Approaches, opportunities, and future directions. *Future Generation Computer Systems* 131 (2022), 209–226.

[38] Ronald Doku and Danda B. Rawat. 2020. IFLBC: On the edge intelligence using federated learning blockchain network. In *Proceedings of the 2020 IEEE 6th International Conference on Big Data Security on Cloud (BigDataSecurity'20), the IEEE International Conference on High Performance and Smart Computing (HPSC'20), and the IEEE International Conference on Intelligent Data and Security (IDS'20)*. IEEE, Los Alamitos, CA, 221–226.

[39] Ronald Doku, Danda B. Rawat, Moses Garuba, and Laurent Njilla. 2019. LightChain: On the lightweight blockchain for the Internet-of-Things. In *Proceedings of the 2019 IEEE International Conference on Smart Computing (SMARTCOMP'19)*. IEEE, Los Alamitos, CA, 444–448.

[40] Josep Domingo-Ferrer, David Sánchez, and Alberto Blanco-Justicia. 2021. The limits of differential privacy (and its misuse in data release and machine learning). *Communications of the ACM* 64, 7 (2021), 33–35.

[41] Sanjeev Kumar Dwivedi, Priyadarshini Roy, Chinky Karda, Shalini Agrawal, and Ruhul Amin. 2021. Blockchain-based Internet of Things and Industrial IoT: A comprehensive survey. *Security and Communication Networks* 2021 (2021), 10958.

[42] Lei Feng, Yiqi Zhao, Shaoyong Guo, Xuesong Qiu, Wenjing Li, and Peng Yu. 2021. BAFL: A blockchain-based asynchronous federated learning framework. *IEEE Transactions on Computers* 71, 5 (2021), 1092–1103. https://ieeexplore.ieee.org/abstract/document/9399813?casa_token=cTtOPOQqSFwAAAAA:pz_-rVdCTcSKxwngJToyadwCksJhYVYC6jm20ch_Q8IFIYtAnJdiKjCtx2xEuDQrp4XdBBpYd4.

[43] Yann Fraboni, Richard Vidal, and Marco Lorenzi. 2021. Free-rider attacks on model aggregation in federated learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*. 1846–1854.

[44] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. 1322–1333.

[45] Karan Ganju, Qi Wang, Wei Yang, Carl A. Gunter, and Nikita Borisov. 2018. Property inference attacks on fully connected neural networks using permutation invariant representations. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. 619–633.

[46] Dashan Gao, Yang Liu, Anbu Huang, Ce Ju, Han Yu, and Qiang Yang. 2019. Privacy-preserving heterogeneous federated transfer learning. In *Proceedings of the 2019 IEEE International Conference on Big Data (Big Data'19)*. IEEE, Los Alamitos, CA, 2552–2559.

[47] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. 2020. Inverting gradients—How easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems* 33 (2020), 16937–16947.

[48] Hanxi Guo, Hao Wang, Tao Song, Yang Hua, Zhangcheng Lv, Xiulang Jin, Zhengui Xue, Ruhui Ma, and Haibing Guan. 2021. Siren: Byzantine-robust federated learning via proactive alarming. In *Proceedings of the ACM Symposium on Cloud Computing*. 47–60.

[49] Meng Hao, Hongwei Li, Xizhao Luo, Guowen Xu, Haomiao Yang, and Sen Liu. 2019. Efficient and privacy-enhanced federated learning for industrial artificial intelligence. *IEEE Transactions on Industrial Informatics* 16, 10 (2019), 6532–6542.

[50] Meng Hao, Hongwei Li, Guowen Xu, Sen Liu, and Haomiao Yang. 2019. Towards efficient and privacy-preserving federated deep learning. In *Proceedings of the 2019 IEEE International Conference on Communications (ICC'19)*. IEEE, Los Alamitos, CA, 1–6.

[51] Xinhong Hei, Xinyue Yin, Yichuan Wang, Ju Ren, and Lei Zhu. 2020. A trusted feature aggregator federated learning for distributed malicious attack detection. *Computers & Security* 99 (2020), 102033.

[52] Tharaka Mawanane Hewa, Yining Hu, Madhusanka Liyanage, Salil Kanhare, and Mika Ylianttila. 2021. Survey on blockchain based smart contracts: Technical aspects and future research. *IEEE Access* 9 (2021), 87643–87662.

[53] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. 2017. Deep models under the GAN: Information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 603–618.

[54] Dongkun Hou, Jie Zhang, Ka Lok Man, Jieming Ma, and Zitian Peng. 2021. A systematic literature review of blockchain-based federated learning: Architectures, applications and issues. In *Proceedings of the 2021 2nd Information Communication Technologies Conference (ICTC'21)*. IEEE, Los Alamitos, CA, 302–307.

[55] Ru Huo, Shiqin Zeng, Zhihao Wang, Jiajia Shang, Wei Chen, Tao Huang, Shuo Wang, F. Richard Yu, and Yunjie Liu. 2022. A comprehensive survey on blockchain in Industrial Internet of Things: Motivations, research progresses, and future challenges. *IEEE Communications Surveys & Tutorials* 24, 1 (2022), 88–122.

[56] Zainab Iftikhar, Yasir Javed, Syed Yawar Abbas Zaidi, Munam Ali Shah, Zafar Iqbal Khan, Shafaq Mussadiq, and Kamran Abbasi. 2021. Privacy preservation in resource-constrained IoT devices using blockchain—A survey. *Electronics* 10, 14 (2021), 1732.

[57] Ahmed Imteaj, Urmish Thakker, Shiqiang Wang, Jian Li, and M. Hadi Amini. 2022. A survey on federated learning for resource-constrained IoT devices. *IEEE Internet of Things Journal* 9, 1 (2022), 1–24.

[58] IoTA. 2022. IoTA. Retrieved September 14, 2022 from https://www.iota.org/get-started/what-is-iota/.

[59] Rafiqul Islam, Muhammad Mahbubur Rahman, Md. Mahmud, Mohammed Ataur Rahman, Muslim Har Sani Mohamad, and Abd Halim Embong. 2021. A review on blockchain security issues and challenges. In *Proceedings of the 2021 IEEE 12th Control and System Graduate Research Colloquium (ICSGRC'21)*. IEEE, Los Alamitos, CA, 227–232.

[60] Malhar S. Jere, Tyler Farnan, and Farinaz Koushanfar. 2020. A taxonomy of attacks on federated learning. *IEEE Security & Privacy* 19, 2 (2020), 20–28.

[61] Bin Jia, Xiaosong Zhang, Jiewen Liu, Yang Zhang, Ke Huang, and Yongquan Liang. 2022. Blockchain-enabled federated learning data protection aggregation scheme with differential privacy and homomorphic encryption in IIoT. *IEEE Transactions on Industrial Informatics* 18, 6 (2022), 4049–4058.

[62] Hai Jin, Xiaohai Dai, Jiang Xiao, Baochun Li, Huichuwu Li, and Yan Zhang. 2021. Cross-cluster federated learning and blockchain for Internet of Medical Things. *IEEE Internet of Things Journal* 8, 21 (2021), 15776–15784.

[63] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, et al. 2021. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning* 14, 1–2 (2021), 1–210.

[64] Jiawen Kang, Zehui Xiong, Dusit Niyato, Shengli Xie, and Junshan Zhang. 2019. Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory. *IEEE Internet of Things Journal* 6, 6 (2019), 10700–10714.

[65] Hillol Kargupta, Souptik Datta, Qi Wang, and Krishnamoorthy Sivakumar. 2003. On the privacy preserving properties of random data perturbation techniques. In *Proceedings of the 3rd IEEE International Conference on Data Mining*. IEEE, Los Alamitos, CA, 99–106.

[66] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. 2020. SCAFFOLD: Stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning*, Hal Daumé III and Aarti Singh (Eds.). Proceedings of Machine Learning Research, Vol. 119. PMLR, 5132–5143. https://proceedings.mlr.press/v119/karimireddy20a.html.

[67] Latif U. Khan, Walid Saad, Zhu Han, and Choong Seon Hong. 2021. Dispersed federated learning: Vision, taxonomy, and future directions. *IEEE Wireless Communications* 28, 5 (2021), 192–198.

[68] Latif U. Khan, Walid Saad, Zhu Han, Ekram Hossain, and Choong Seon Hong. 2021. Federated learning for Internet of Things: Recent advances, taxonomy, and open challenges. *IEEE Communications Surveys & Tutorials* PP, 99 (2021), 1.

[69] Minhaj Ahmad Khan and Khaled Salah. 2018. IoT security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems* 82 (2018), 395–411.

[70] Safiullah Khan, Wai-Kong Lee, and Seong Oun Hwang. 2022. AEchain: A lightweight blockchain for IoT applications. *IEEE Consumer Electronics Magazine* 11, 2 (2022), 64–76.

[71] Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. 2019. Blockchained on-device federated learning. *IEEE Communications Letters* 24, 6 (2019), 1279–1283.

[72] Rajesh Kumar and Rewa Sharma. 2021. Leveraging blockchain for ensuring trust in IoT: A survey. *Journal of King Saud University-Computer and Information Sciences*. In press.
[73] Aparna Kumari, Rajesh Gupta, and Sudeep Tanwar. 2021. Amalgamation of blockchain and IoT for smart cities underlying 6G communication: A comprehensive review. *Computer Communications* 172 (2021), 102–118.
[74] Laphou Lao, Zecheng Li, Songlin Hou, Bin Xiao, Songtao Guo, and Yuanyuan Yang. 2020. A survey of IoT applications in blockchain systems: Architecture, consensus, and traffic modeling. *ACM Computing Surveys* 53, 1 (2020), 1–32.
[75] Shahid Latif, Zeba Idrees, Zil e Huma, and Jawad Ahmad. 2021. Blockchain technology for the Industrial Internet of Things: A comprehensive survey on security challenges, architectures, applications, and future research directions. *Transactions on Emerging Telecommunications Technologies* 32, 11 (2021), e4337.
[76] Haemin Lee and Joongheon Kim. 2021. Trends in blockchain and federated learning for data sharing in distributed platforms. In *Proceedings of the 2021 12th International Conference on Ubiquitous and Future Networks (ICUFN'21)*. IEEE, Los Alamitos, CA, 430–433.
[77] Dun Li, Dezhi Han, Tien-Hsiung Weng, Zibin Zheng, Hongzhi Li, Han Liu, Arcangelo Castiglione, and Kuan-Ching Li. 2022. Blockchain for federated learning toward secure distributed machine learning systems: A systemic survey. *Soft Computing* 26, 9 (2022), 4423–4440.
[78] Jun Li, Yumeng Shao, Kang Wei, Ming Ding, Chuan Ma, Long Shi, Zhu Han, and H. Vincent Poor. 2022. Blockchain assisted decentralized federated learning (BLADE-FL): Performance analysis and resource allocation. *IEEE Transactions on Parallel and Distributed Systems* 33, 10 (2022), 2401–2415. https://doi.org/10.1109/TPDS.2021.3138848
[79] Jun Li, Yumeng Shao, Kang Wei, Ming Ding, Chuan Ma, Long Shi, Zhu Han, and Vincent Poor. 2021. Blockchain assisted decentralized federated learning (BLADS-FL): Performance analysis and resource allocation. *IEEE Transactions on Parallel and Distributed Systems* 33, 10 (2021), 2401–2415.
[80] Li Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. 2020. A review of applications in federated learning. *Computers & Industrial Engineering* 149 (2020), 106854.
[81] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. 2007. *t*-closeness: Privacy beyond *k*-anonymity and *l*-diversity. In *Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering*. IEEE, Los Alamitos, CA, 106–115.
[82] Shenghui Li, Edith Ngai, and Thiemo Voigt. 2021. Byzantine-robust aggregation in federated learning empowered Industrial IoT. *IEEE Transactions on Industrial Informatics*. Early access, November 15, 2021.
[83] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems* 2 (2020), 429–450.
[84] Pengrui Liu, Xiangrui Xu, and Wei Wang. 2022. Threats, attacks and defenses to federated learning: Issues, taxonomy and perspectives. *Cybersecurity* 5, 1 (2022), 1–19.
[85] Ruixuan Liu, Yang Cao, Masatoshi Yoshikawa, and Hong Chen. 2020. FedSel: Federated SGD under local differential privacy with top-*k* dimension selection. In *Proceedings of the International Conference on Database Systems for Advanced Applications*. 485–501.
[86] Xiaoyuan Liu, Hongwei Li, Guowen Xu, Zongqi Chen, Xiaoming Huang, and Rongxing Lu. 2021. Privacy-enhanced federated learning against poisoning adversaries. *IEEE Transactions on Information Forensics and Security* 16 (2021), 4574–4588.
[87] Yiming Liu, F. Richard Yu, Xi Li, Hong Ji, and Victor C. M. Leung. 2020. Blockchain and machine learning for communications and networking systems. *IEEE Communications Surveys & Tutorials* 22, 2 (2020), 1392–1431.
[88] Auqib Hamid Lone and Roohie Naaz. 2021. Applicability of blockchain smart contracts in securing internet and IoT: A systematic literature review. *Computer Science Review* 39 (2021), 100360.
[89] Yunlong Lu, Xiaohong Huang, Ke Zhang, Sabita Maharjan, and Yan Zhang. 2020. Blockchain empowered asynchronous federated learning for secure data sharing in Internet of Vehicles. *IEEE Transactions on Vehicular Technology* 69, 4 (2020), 4298–4311.
[90] Yunlong Lu, Xiaohong Huang, Ke Zhang, Sabita Maharjan, and Yan Zhang. 2020. Communication-efficient federated learning and permissioned blockchain for digital twin edge networks. *IEEE Internet of Things Journal* 8, 4 (2020), 2276–2288.
[91] Lingjuan Lyu, Han Yu, Jun Zhao, and Qiang Yang. 2020. Threats to federated learning. In *Federated Learning*. Springer, 3–16.
[92] Mohammad Saeid Mahdavinejad, Mohammadreza Rezvan, Mohammadamin Barekatain, Peyman Adibi, Payam Barnaghi, and Amit P. Sheth. 2018. Machine learning for Internet of Things data analysis: A survey. *Digital Communications and Networks* 4, 3 (2018), 161–175.
[93] Umer Majeed, Latif U. Khan, Ibrar Yaqoob, S. M. Ahsan Kazmi, Khaled Salah, and Choong Seon Hong. 2021. Blockchain for IoT-based smart cities: Recent advances, requirements, and future challenges. *Journal of Network and Computer Applications* 181 (2021), 103007.

[94] Mohsen Marjani, Fariza Nasaruddin, Abdullah Gani, Ahmad Karim, Ibrahim Abaker Targio Hashem, Aisha Siddiqa, and Ibrar Yaqoob. 2017. Big IoT data analytics: Architecture, opportunities, and open research challenges. *IEEE Access* 5 (2017), 5247–5261.

[95] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*. PMLR, 1273–1282.

[96] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. 2019. Exploiting unintended feature leakage in collaborative learning. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP'19)*. IEEE, Los Alamitos, CA, 691–706.

[97] Du Mingxiao, Ma Xiaofeng, Zhang Zhe, Wang Xiangwei, and Chen Qijun. 2017. A review on consensus algorithm of blockchain. In *Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC'17)*. IEEE, Los Alamitos, CA, 2567–2572.

[98] Dandan Mo. 2012. *A Survey on Deep Learning: One Small Step Toward AI*. Department of Computer Science, University of New Mexico.

[99] Mehdi Mohammadi, Ala Al-Fuqaha, Sameh Sorour, and Mohsen Guizani. 2018. Deep learning for IoT big data and streaming analytics: A survey. *IEEE Communications Surveys & Tutorials* 20, 4 (2018), 2923–2960.

[100] Sachi Nandan Mohanty, K. C. Ramya, S. Sheeba Rani, Deepak Gupta, K. Shankar, S. K. Lakshmanaprabu, and Ashish Khanna. 2020. An efficient lightweight integrated blockchain (ELIB) model for IoT security and privacy. *Future Generation Computer Systems* 102 (2020), 1027–1037.

[101] Ahmed Afif Monrat, Olov Schelén, and Karl Andersson. 2019. A survey of blockchain from the perspectives of applications, challenges, and opportunities. *IEEE Access* 7 (2019), 117134–117151.

[102] Viraaji Mothukuri, Reza M. Parizi, Seyedamin Pouriyeh, Yan Huang, Ali Dehghantanha, and Gautam Srivastava. 2021. A survey on security and privacy of federated learning. *Future Generation Computer Systems* 115 (2021), 619–640.

[103] Satoshi Nakamoto. 2008. A Peer-to-Peer Electronic Cash System. Retrieved September 14, 2022 from https://bitcoin.org/en/bitcoin-paper.

[104] Samudaya Nanayakkara, M. N. N. Rodrigo, Srinath Perera, G. T. Weerasuriya, and Amer A. Hijazi. 2021. A methodology for selection of a blockchain platform to develop an enterprise system. *Journal of Industrial Information Integration* 23 (2021), 100215.

[105] Qassim Nasir, Ilham A. Qasse, Manar Abu Talib, and Ali Bou Nassif. 2018. Performance analysis of hyperledger fabric platforms. *Security and Communication Networks* 2018 (2018), Article 3976093.

[106] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP'19)*. IEEE, Los Alamitos, CA, 739–753.

[107] Quoc-Dung Ngo, Huy-Trung Nguyen, Van-Hoang Le, and Doan-Hieu Nguyen. 2020. A survey of IoT malware and detection methods based on static features. *ICT Express* 6, 4 (2020), 280–286.

[108] Dinh C. Nguyen, Ming Ding, Pubudu N. Pathirana, Aruna Seneviratne, Jun Li, and H. Vincent Poor. 2021. Federated learning for Internet of Things: A comprehensive survey. *IEEE Communications Surveys & Tutorials* 23, 3 (2021), 1622–1658.

[109] Dinh C. Nguyen, Ming Ding, Quoc-Viet Pham, Pubudu N. Pathirana, Long Bao Le, Aruna Seneviratne, Jun Li, Dusit Niyato, and H. Vincent Poor. 2021. Federated learning meets blockchain in edge computing: Opportunities and challenges. *IEEE Internet of Things Journal* 8, 16 (2021), 12806–12825.

[110] Dinh C. Nguyen, Quoc-Viet Pham, Pubudu N. Pathirana, Ming Ding, Aruna Seneviratne, Zihuai Lin, Octavia Dobre, and Won-Joo Hwang. 2022. Federated learning for smart healthcare: A survey. *ACM Computing Surveys* 55, 3 (2022), 1–37.

[111] Giang-Truong Nguyen and Kyungbaek Kim. 2018. A survey about consensus algorithms used in blockchain. *Journal of Information Processing Systems* 14, 1 (2018), 101–128.

[112] Michael Nielsen. 2006. Deep Learning. Retrieved September 14, 2022 from http://neuralnetworksanddeeplearning.com/chap6.html/.

[113] Maxence Noble, Aurélien Bellet, and Aymeric Dieuleveut. 2022. Differentially private federated learning on heterogeneous data. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera (Eds.). Proceedings of Machine Learning Research, Vol. 151. PMLR, 10110–10145. https://proceedings.mlr.press/v151/noble22a.html.

[114] Diaa A. Noby and Ahmed Khattab. 2019. A survey of blockchain applications in IoT systems. In *Proceedings of the 2019 14th International Conference on Computer Engineering and Systems (ICCES'19)*. IEEE, Los Alamitos, CA, 83–87.

[115] Safa Otoum, Ismaeel Al Ridhawi, and Hussein Mouftah. 2021. Securing critical IoT infrastructures with blockchain-supported federated learning. *IEEE Internet of Things Journal* (2021).

[116] Safa Otoum, Ismaeel Al Ridhawi, and Hussein T. Mouftah. 2020. Blockchain-supported federated learning for trustworthy vehicular networks. In *Proceedings of the 2020 IEEE Global Communications Conference (GLOBECOM'20)*. IEEE, Los Alamitos, CA, 1–6.

[117] Sunny Pahlajani, Avinash Kshirsagar, and Vinod Pachghare. 2019. Survey on private blockchain consensus algorithms. In *Proceedings of the 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)*. IEEE, Los Alamitos, CA, 1–6.

[118] Alfonso Panarello, Nachiket Tapas, Giovanni Merlino, Francesco Longo, and Antonio Puliafito. 2018. Blockchain and IoT integration: A systematic survey. *Sensors* 18, 8 (2018), 2575.

[119] Marco Picone, Simone Cirani, and Luca Veltri. 2021. Blockchain security and privacy for the Internet of Things. *Sensors (Basel)* 21, 3 (2021), 892.

[120] Yuanhang Qi, M. Shamim Hossain, Jiangtian Nie, and Xuandi Li. 2021. Privacy-preserving blockchain-based federated learning for traffic flow prediction. *Future Generation Computer Systems* 117 (2021), 328–337.

[121] Zhenquan Qin, Jin Ye, Jie Meng, Bingxian Lu, and Lei Wang. 2022. Privacy-preserving blockchain-based federated learning for marine Internet of Things. *IEEE Transactions on Computational Social Systems* 9, 1 (2022), 159–173.

[122] Youyang Qu, Longxiang Gao, Tom H. Luan, Yong Xiang, Shui Yu, Bai Li, and Gavin Zheng. 2020. Decentralized privacy using blockchain-enabled federated learning in fog computing. *IEEE Internet of Things Journal* 7, 6 (2020), 5171–5183.

[123] R3. 2022. Corda. Retrieved May 21, 2022 from https://www.corda.net/why-corda/.

[124] Valerian Rey, Pedro Miguel Sánchez Sánchez, Alberto Huertas Celdrán, and Gérôme Bovet. 2022. Federated learning for malware detection in IoT devices. *Computer Networks* 204 (2022), 108693.

[125] Ana Reyna, Cristian Martín, Jaime Chen, Enrique Soler, and Manuel Díaz. 2018. On blockchain and its integration with IoT: Challenges and opportunities. *Future Generation Computer Systems* 88 (2018), 173–190.

[126] Sara Salim, Benjamin Turnbull, and Nour Moustafa. 2021. A blockchain-enabled explainable federated learning for securing Internet-of-Things-based Social Media 3.0 networks. *IEEE Transactions on Computational Social Systems*. Early access, December 28, 2021.

[127] Mehrdad Salimitari, Mainak Chatterjee, and Yaser P. Fallah. 2020. A survey on consensus methods in blockchain for resource-constrained IoT networks. *Internet of Things* 11 (2020), 100212.

[128] Felix Sattler, Klaus-Robert Müller, Thomas Wiegand, and Wojciech Samek. 2020. On the Byzantine robustness of clustered federated learning. In *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'20)*. IEEE, Los Alamitos, CA, 8861–8865.

[129] Shivam Saxena, Bharat Bhushan, and Mohd Abdul Ahad. 2021. Blockchain based solutions to secure IoT: Background, integration trends and a way forward. *Journal of Network and Computer Applications* 181 (2021), 103050.

[130] Adi Shamir. 1979. How to share a secret. *Communications of the ACM* 22, 11 (1979), 612–613.

[131] Parjanay Sharma, Siddhant Jain, Shashank Gupta, and Vinay Chamola. 2021. Role of machine learning and deep learning in securing 5G-driven Industrial IoT applications. *Ad Hoc Networks* 123 (2021), 102685.

[132] Pradip Kumar Sharma, Jong Hyuk Park, and Kyungeun Cho. 2020. Blockchain and federated learning-based distributed computing defence framework for sustainable society. *Sustainable Cities and Society* 59 (2020), 102220.

[133] Andrew Ronald Short, Helen C. Leligou, Michael Papoutsidakis, and Efstathios Theocharis. 2020. Using blockchain technologies to improve security in federated learning systems. In *Proceedings of the 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*. IEEE, Los Alamitos, CA, 1183–1188.

[134] Andrew R. Short, Helen C. Leligou, and Efstathios Theocharis. 2021. Execution of a federated learning process within a smart contract. In *Proceedings of the 2021 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, Los Alamitos, CA, 1–4.

[135] Arshdeep Singh, Gulshan Kumar, Rahul Saha, Mauro Conti, Mamoun Alazab, and Reji Thomas. 2022. A survey and taxonomy of consensus protocols for blockchains. *Journal of Systems Architecture* 127 (2022), 102503.

[136] Shivani Singh, Razia Sulthana, Tanvi Shewale, Vinay Chamola, Abderrahim Benslimane, and Biplab Sikdar. 2021. Machine-learning-assisted security and privacy provisioning for edge computing: A survey. *IEEE Internet of Things Journal* 9, 1 (2021), 236–260.

[137] Jinhyun So, Başak Güler, and A. Salman Avestimehr. 2021. Byzantine-resilient secure federated learning. *IEEE Journal on Selected Areas in Communications* 39, 7 (2021), 2168–2181.

[138] Mengkai Song, Zhibo Wang, Zhifei Zhang, Yang Song, Qian Wang, Ju Ren, and Hairong Qi. 2020. Analyzing user-level privacy attack against federated learning. *IEEE Journal on Selected Areas in Communications* 38, 10 (2020), 2430–2444.

[139] Jack Steward. 2021. The Ultimate List of Internet of Things Statistics for 2021. Retrieved September 25, 2021 from https://findstack.com/internet-of-things-statistics/.

[140] Dimitris Stripelis, Paul M. Thompson, and José Luis Ambite. 2022. Semi-synchronous federated learning for energy-efficient training and accelerated convergence in cross-silo settings. *ACM Transactions on Intelligent Systems and Technology* 13, 5 (2022), Article 78, 29 pages.

[141] Jin Sun, Ying Wu, Shangping Wang, Yixue Fu, and Xiao Chang. 2022. A permissioned blockchain frame for secure federated learning. *IEEE Communications Letters* 26, 1 (2022), 13–17.

[142] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. 2022. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*. Accepted.

[143] Sudeep Tanwar, Qasim Bhatia, Pruthvi Patel, Aparna Kumari, Pradeep Kumar Singh, and Wei-Chiang Hong. 2019. Machine learning adoption in blockchain-based smart applications: The challenges, and a way forward. *IEEE Access* 8 (2019), 474–488.

[144] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. 2019. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*. 1–11.

[145] Devrim Unal, Mohammad Hammoudeh, Muhammad Asif Khan, Abdelrahman Abuarqoub, Gregory Epiphaniou, and Ridha Hamila. 2021. Integration of federated machine learning and blockchain for the provision of secure big data analytics for Internet of Things. *Computers & Security* 109 (2021), 102393.

[146] Bram van Berlo, Aaqib Saeed, and Tanir Ozcelebi. 2020. Towards federated unsupervised representation learning. In *Proceedings of the 3rd ACM International Workshop on Edge Systems, Analytics, and Networking*. 31–36.

[147] S. Velliangiri and P. Karthikeyan. 2020. Blockchain technology: Challenges and security issues in consensus algorithm. In *Proceedings of the 2020 International Conference on Computer Communication and Informatics (ICCCI'20)*. IEEE, Los Alamitos, CA, 1–8.

[148] Nazar Waheed, Xiangjian He, Muhammad Ikram, Muhammad Usman, Saad Sajid Hashmi, and Muhammad Usman. 2020. Security and privacy in IoT using machine learning and blockchain: Threats and countermeasures. *ACM Computing Surveys* 53, 6 (2020), 1–37.

[149] Shuai Wang, Liwei Ouyang, Yong Yuan, Xiaochun Ni, Xuan Han, and Fei-Yue Wang. 2019. Blockchain-enabled smart contracts: Architecture, applications, and future trends. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 49, 11 (2019), 2266–2277.

[150] Shuai Wang, Yong Yuan, Xiao Wang, Juanjuan Li, Rui Qin, and Fei-Yue Wang. 2018. An overview of smart contract: Architecture, applications, and future trends. In *Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV'18)*. IEEE, Los Alamitos, CA, 108–113.

[151] Xu Wang, Xuan Zha, Wei Ni, Ren Ping Liu, Y. Jay Guo, Xinxin Niu, and Kangfeng Zheng. 2019. Survey on blockchain for Internet of Things. *Computer Communications* 136 (2019), 10–29.

[152] Yuhao Wang, Shaobin Cai, Changlong Lin, Zuxi Chen, Tian Wang, Zhenguo Gao, and Changli Zhou. 2019. Study of blockchains's consensus mechanism based on credit. *IEEE Access* 7 (2019), 10224–10231.

[153] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. 2019. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *Proceedings of the IEEE Conference on Computer Communications (IEEE INFOCOM'19)*. IEEE, Los Alamitos, CA, 2512–2520.

[154] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H. Yang, Farhad Farokhi, Shi Jin, Tony Q. S. Quek, and H. Vincent Poor. 2020. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security* 15 (2020), 3454–3469.

[155] Zhaohui Wei, Qingqi Pei, Ning Zhang, Xuefeng Liu, Celimuge Wu, and Amirhosein Taherkordi. 2021. Lightweight federated learning for large-scale IoT devices with privacy guarantee. *IEEE Internet of Things Journal*. Early access, November 15, 2021.

[156] Jiasi Weng, Jian Weng, Jilian Zhang, Ming Li, Yue Zhang, and Weiqi Luo. 2019. DeepChain: Auditable and privacy-preserving deep learning with blockchain-based incentive. *IEEE Transactions on Dependable and Secure Computing* 18, 5 (2019), 2438–2455.

[157] Mingli Wu, Kun Wang, Xiaoqin Cai, Song Guo, Minyi Guo, and Chunming Rong. 2019. A comprehensive survey of blockchain: From theory to IoT applications and beyond. *IEEE Internet of Things Journal* 6, 5 (2019), 8114–8154.

[158] Wentai Wu, Ligang He, Weiwei Lin, Rui Mao, Carsten Maple, and Stephen Jarvis. 2020. SAFA: A semi-asynchronous protocol for fast federated learning with low overhead. *IEEE Transactions on Computers* 70, 5 (2020), 655–668.

[159] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. 2020. DBA: Distributed backdoor attacks against federated learning. In *Proceedings of the International Conference on Learning Representations*. https://openreview.net/forum?id=rkgyS0VFvr.

[160] Junfeng Xie, Helen Tang, Tao Huang, F. Richard Yu, Renchao Xie, Jiang Liu, and Yunjie Liu. 2019. A survey of blockchain technology applied to smart cities: Research issues and challenges. *IEEE Communications Surveys & Tutorials* 21, 3 (2019), 2794–2830.

[161] Yan Xu, Tao Mo, Qiwei Feng, Peilin Zhong, Maode Lai, and Eric I.-Chao Chang. 2014. Deep learning of feature representation with multiple instance learning for medical image analysis. In *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'14)*. IEEE, Los Alamitos, CA, 1626–1630.

[162] Chenhao Xu, Youyang Qu, Yong Xiang, and Longxiang Gao. 2021. Asynchronous federated learning on heterogeneous devices: A survey. *arXiv:2109.04269*.

[163] Guowen Xu, Hongwei Li, Sen Liu, Kan Yang, and Xiaodong Lin. 2019. VerifyNet: Secure and verifiable federated learning. *IEEE Transactions on Information Forensics and Security* 15 (2019), 911–926.
[164] Runhua Xu, Nathalie Baracaldo, Yi Zhou, Ali Anwar, and Heiko Ludwig. 2019. Hybridalpha: An efficient approach for privacy-preserving federated learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*. 13–23.
[165] Krishna Yadav, Brij B. Gupta, Ching-Hsein Hsu, and Kwok Tai Chui. 2021. Unsupervised federated learning based IoT intrusion detection. In *Proceedings of the 2021 IEEE 10th Global Conference on Consumer Electronics (GCCE'21)*. IEEE, Los Alamitos, CA, 298–301.
[166] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology* 10, 2 (2019), 1–19.
[167] Xuefei Yin, Yanming Zhu, and Jiankun Hu. 2021. A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. *ACM Computing Surveys* 54, 6 (2021), 1–36.
[168] Yong Yuan and Fei-Yue Wang. 2018. Blockchain and cryptocurrencies: Model, techniques, and applications. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 48, 9 (2018), 1421–1428.
[169] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. 2021. A survey on federated learning. *Knowledge-Based Systems* 216 (2021), 106775.
[170] Haichao Zhang and Jianyu Wang. 2019. Defense against adversarial attacks using feature scattering-based adversarial training. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NIPS'19)*. 1831–1841.
[171] Jiale Zhang, Bing Chen, Xiang Cheng, Huynh Thi Thanh Binh, and Shui Yu. 2020. PoisonGAN: Generative poisoning attacks against federated learning in edge computing systems. *IEEE Internet of Things Journal* 8, 5 (2020), 3310–3322.
[172] Jiale Zhang, Bing Chen, Shui Yu, and Hai Deng. 2019. PEFL: A privacy-enhanced federated learning scheme for big data analytics. In *Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM'19)*. IEEE, Los Alamitos, CA, 1–6.
[173] Jiale Zhang, Di Wu, Chengyong Liu, and Bing Chen. 2020. Defending poisoning attacks in federated learning via adversarial training method. In *Proceedings of the International Conference on Frontiers in Cyber Security*. 83–94.
[174] Jingwen Zhang, Jiale Zhang, Junjun Chen, and Shui Yu. 2020. GAN enhanced membership inference: A passive local attack in federated learning. In *Proceedings of the 2020 IEEE International Conference on Communications (ICC'20)*. IEEE, Los Alamitos, CA, 1–6.
[175] Peiying Zhang, Hao Sun, Jingyi Situ, Chunxiao Jiang, and Dongliang Xie. 2021. Federated transfer learning for IIoT devices with low computing power based on blockchain and edge computing. *IEEE Access* 9 (2021), 98630–98638.
[176] Weishan Zhang, Qinghua Lu, Qiuyu Yu, Zhaotong Li, Yue Liu, Sin Kit Lo, Shiping Chen, Xiwei Xu, and Liming Zhu. 2020. Blockchain-based federated learning for device failure detection in Industrial IoT. *IEEE Internet of Things Journal* 8, 7 (2020), 5926–5937.
[177] Yang Zhao, Jun Zhao, Linshan Jiang, Rui Tan, Dusit Niyato, Zengxiang Li, Lingjuan Lyu, and Yingbo Liu. 2020. Privacy-preserving blockchain-based federated learning for IoT devices. *IEEE Internet of Things Journal* 8, 3 (2020), 1817–1829.
[178] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Weili Chen, Xiangping Chen, Jian Weng, and Muhammad Imran. 2020. An overview on smart contracts: Challenges, advances and platforms. *Future Generation Computer Systems* 105 (2020), 475–491.
[179] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. 2021. Federated learning on non-IID data: A survey. *Neurocomputing* 465 (2021), 371–390.
[180] Ligeng Zhu and Song Han. 2020. Deep leakage from gradients. In *Federated Learning*. Springer, 17–31.
[181] Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep leakage from gradients. In *Advances in Neural Information Processing Systems 32 (NeurIPS'19)*.

Received 14 March 2022; revised 23 June 2022; accepted 8 August 2022