Emerging Technologies in Federated Learning

Felix Plantenberg
Department of Computer Science
Technical University of Munich (TUM)
Munich, Germany
felix.plantenberg@tum.de

Ivayla Ivanova
Department of Computer Science
Ludwig Maximilian University of Munich (LMU)
Munich, Germany
Ivayla.Ivanova@campus.lmu.de

***Abstract*—Federated learning is a machine learning paradigm that solves distributed optimization tasks. Data is shared across a large number of devices which learn a joint prediction model. The goal is to preserve privacy-sensitive data locally on each device, while propagating encrypted updates to a central server. This review presents the general principles and architectures of federated learning and analyzes solutions to various privacy issues in the field of secure computation, verifiability and other privacy-preserving techniques. By investigating the question whether federated learning preserves privacy, five factors are identified that emerge when utilizing proven techniques incorporated into the federated learning model which are additional model complexity, computational overhead, decreased model utility, reduced model accuracy and introducing trusted intermediate parties.**

I. Introduction

In recent years, the rapid evolution of machine learning (ML) and deep learning (DL) has earned a great interest across a variety of industries. However, there is still need for innovative applications leveraging this potential. A significant challenge emerges in the form of non-independent and non-identically distributed (non-i.i.d.) data across clients [1]. For example, deploying a machine learning model to predict patient diagnoses across multiple hospitals. The data within each hospital is non-independent as patients share common characteristics. In addition, it is non-identically distributed due to distinct demographics, spending capabilities or healthcare practices. This diversity poses difficulties for model generalization among hospitals, which highlights the relevance of new approaches to address this issue. Moreover, privacy concerns associated with personal data like the sensitivity of health records have increased, creating a demand for a more ethical deployment of machine learning models.

Federated learning (FL) is a technique used in machine learning in order to solve distributed optimization tasks [2]. Data is shared across a large number of remote entities who collaboratively learn a joint prediction model. The goal is to preserve privacy-sensitive data locally on each device, while propagating encrypted updates of the model to a central server. Moreover, concepts like federated averaging (FedAvg) over multiple clients as well as sharing of model parameters instead of raw data with the global model ensure anonymity of single clients [3]. In doing so, federated learning addresses some elementary issues of privacy, ownership, and locality of data [4]. Therefore, it is suitable for scenarios where data privacy is a concern and centralizing data is not feasible or desirable. The concept has first been described by Konečny et al. [5] in 2016 and was generally defined by McMahan and Ramage [2] in 2017.

FL benefits from accessing the client's data locally, thereby facilitating the training of a personalized model. One of the first prominent use-cases has been Google´s next word prediction [6], [7]. Other than that, Liu et al. [8] present a broad scope of promising application fields for predictive models utilizing FL such as edge computing, healthcare, smart cities, physical information systems, finance or industrial manufacturing. For example, in the healthcare domain FL is applied to confidentiality-preserving estimation of pain from facial expressions without sharing raw personal data (face images) [9]. Rudovic et al. [9] demonstrate that the performance of the personalized FL approach is comparable and even outperforms the conventional centralized learning methodologies.

While solving some privacy related problems, FL also poses challenges arising from the decentralized approach. Guendouzi et al. [3] distinguish between four major categories, namely:

1) Privacy concerns: This category refers to a secure connection, encrypted communication as well as the data that is shared between the different parties.
2) Costly communication: The problem of efficiently exchanging all the necessary information between server and client.
3) System heterogeneity: This issue addresses challenges arising from the different environments on every device.
4) Statistical heterogeneity: This challenge mirrors potential variations in data distributions like the aforementioned nature of non i.i.d. data along with learning patterns across distinct nodes.

In the following section II a short overview of the most prominent research is given. Followed by section III that provides an overview of the classical FL architecture as well as modifications of the classical approach, such as hierarchical FL, cluster-based FL or decentralized FL. Section IV concentrates on techniques that offer privacy guarantees. Through the combination of proven concepts together with aforementioned FL architectures, many promising technologies emerge that collectively aim to enhance the security and privacy of federated learning systems. By following this structure we want to investigate the core assumption of federated learning: Does it preserve privacy?

II. Existing research

Driven by an increasing interest in more privacy and numerous innovative approaches and architectures in the FL field, many reviews and surveys have been conducted in the past three years. A very comprehensive study of current advances and open problems in FL was performed by Kairouz et al. [10]. Ji et al. [11] conducted their research on emerging trends, differentiating between model fusion approaches such as adaptive aggregation, regularization, clustered methods, and Bayesian methods as well as other learning paradigms terming them federated X learning, where X represents e.g. multitask, transfer or unsupervised. Four recent noteworthy reviews extensively examined and assessed federated learning with respect to privacy and security by looking at threats and presenting according countermeasures [12]–[15].

Shanmugarasa et al. [16] explored FL challenges and available solutions from a client-side. They conducted a comprehensive literature review on 238 primary studies and additionally identified whether solutions could be transferred to other challenges. The research of Liu et al. [17] takes a look at privacy preserving federated learning (PPFL) with a specific focus on aggregation techniques. They evaluate various protocols regarding their advantages and disadvantages. Additionally, they discuss open-source FL frameworks supporting privacy preserving aggregation.

Mothukuri et al. [18] directed their attention to privacy-specific threats in contrast to security threats and present mitigation techniques. Gu et al. [14] studied privacy enhancing methods specifically in the healthcare domain.

Another emerging trend in the field of decentralized learning approaches is the combination of blockchain technology with the FL paradigm. The study of Issa et al. explores [19] the potential of utilizing blockchain and smart contracts to enhance the security of FL in Internet of Things (IoT) ecosystems.

With all the focus on new technologies that aim to enhance privacy and security, it is equally important to evaluate whether such a technique provides sufficient enough protection and how attackers can circumvent them. Several papers take a look on potential attacks to FL systems such as [15], [20], [21]. For example, Boenisch et al. claim 'federated learning is not private' by demonstrating that through the observation of propagated model parameters, such as gradients, an potential attacker can reconstruct data of individual users [21, p.1]. However, in our review we will not particularly focus on that research direction.

III. Overview of federated learning categories

Given the distributed nature of the framework, FL consistently deals with the problems caused by device and communication unreliability. As a result, alternatives to the standard server-client paradigm have emerged. The following sections describes the main attributes of the classical FL and how the other approaches differ from it. Despite the effectiveness of the proposed methods in addressing certain FL-related issues, there is an increasing effect on the complexity of the system design and new vulnerabilities are introduced.

A. Classical FL framework

FL is employed when there is a machine learning task that should be collaboratively solved by multiple participants unrelated to each other [2]. To describe the classical features of FL, we refer to the definition provided by Kairouz et al. [10]:

> Federated learning is a machine learning setting where multiple entities (clients) collaborate in solving a machine learning problem, under the coordination of a central server or service provider. Each client's raw data is stored locally and not exchanged or transferred; instead, focused updates intended for immediate aggregation are used to achieve the learning objective.

Figure 1 demonstrates the iterative learning process within a FL setting. In the first phase a central server shares prior model parameters with a subset of participating clients. Next, the local model is trained by the clients and then aggregated into an update that is shared with the server. Using an appropriate aggregation algorithm such as classical averaging the server includes this update into the global model. Lastly, the updated model parameters are shared with the clients again [10].

The data sets held by clients may show similarities in terms of features but have different samples, or the scenario might occur where clients share similar or identical data samples but with different features. For example, while application A and application B may share a common user base, the domains of the data collected by each application is different, or Application A and B collect similar data about their users, but have different user base. There is also a third option where clients have neither intersections regarding the data samples nor regarding the feature space. The distribution of data has an effect on the federated learning process. Thus, the literature established a categorization into: horizontal (similar features, different samples), vertical (different features, similar samples), and transfer federated learning. Horizontal distribution is assumed in the majority of federated learning use-cases [22].

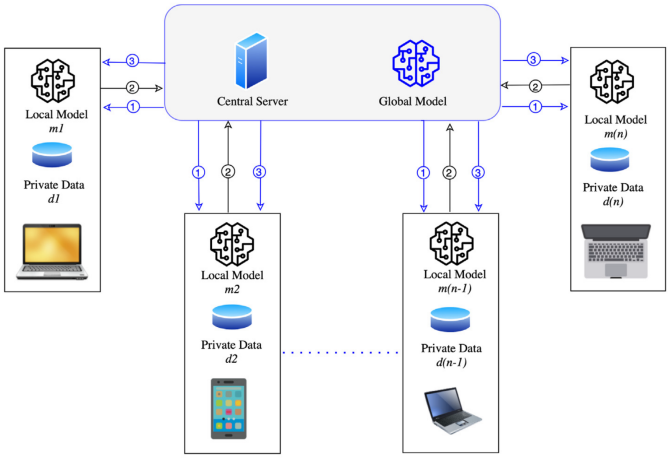

Fig. 1. Process flow of Federated Learning. Adapted from [18].

A second categorization performed in the literature is differentiating between scale and purpose of the federation. Cross-device settings involve a huge number of devices, each with data sets containing different samples of data but similar features. Cross-silo FL usually involve a smaller number of large organizations, that are not only interested in the horizontal distribution of the data but also in the distribution of features [10].

B. Cluster-based FL

Cluster-based FL involves organizing clients into clusters, typically based on computational and storage resources or the distribution of data on the devices. Unlike selecting individual nodes randomly or using more advanced selection algorithms, this approach first groups nodes into clusters. A selection and scheduling strategy is applied to the clusters by the centralized server [23] [24].

The first clustering type aims to enhance efficiency by grouping clients with similar capacities into clusters. However, this approach tends to overlook the uneven distribution of data, as well as the quality and relevance of the locally held data sets on devices. While this clustering strategy may improve overall training efficiency, there is a risk of declining model accuracy if the clustering and selection mechanism prioritizes clients that do not contribute significantly to the global model improvement [24].

In the literature, there currently exists a stronger focus on the second type of clustering within FL [25] [24] [26]. This type seeks to address statistical heterogeneity by organizing devices into groups that show similarities in the probability distribution of data or label distribution [25]. Unlike the first clustering approach, which mainly considers computational capacities and stable connectivity, this second type aims to soften the impact of uneven data distribution and enhance the convergence of models by grouping devices based on the statistical characteristics of their local data sets [24]. This strategic clustering is particularly relevant in situations where statistical heterogeneity among devices challenges the achievement of optimal global model performance in federated learning environments [24].

The clustering approach is closely related to the general client selection and client scheduling topic in FL. Ye et al. [27], Fu et al. [23] and Xu et al. [26] provide a comprehensive overview of currently discussed techniques contributing to the overall improvement of model accuracy and efficient resource utilization. Advanced selection and scheduling algorithms require additional information, either regarding the client devices themselves or the data they hold [25]. These algorithms play a crucial role in shaping the efficiency and effectiveness of FL systems by making informed decisions about which clients to involve in the collaborative learning process and when to schedule their participation.

However, the requirement for additional information carries additional privacy risks. While some of these risks are theoretically covered by well-known privacy methods as mentioned in IV the continuous evolution of new threats and attack approaches remains a challenge [25].

C. Hierarchical FL

In the clustered-based approach discussed in the previous section, it is assumed that there is one central server responsible for node selection, cluster formation, and scheduling. The central server can be implemented as either a cloud or edge server. Liu et al. [28] discusses the advantages and disadvantages of cloud and edge servers, introducing the concept of hierarchical federated learning as a combination of both approaches.

Cloud servers, as in the traditional FL approach, can reach a broad range of clients. However, due to the costs and unreliability of communication networks, cloud servers may not provide an optimal training experience. On the other hand, edge servers have a limited reach, making a single edge server unsuitable for many FL scenarios and use-cases [28]. Hierarchical FL proposes a hybrid approach where a single cloud server orchestrates a network of edge servers responsible for a more restricted number of clients.

In this setup, the cloud server initializes the global model and handles the final aggregation of local models. Edge servers act as central servers for their group of clients and also function as clients for the cloud server. During the training phase, clients exclusively communicate with their respective edge servers, exchanging updates on model parameters.

Hierarchical FL shows potential of increased efficiency and can be integrated with clustering and selection techniques discussed in the previous section. Ye et al. [27] and Fu et al. [23] categorize this topic as a promising sub-field that require further research and can significantly contribute to addressing system and statistical heterogeneity in FL. Although additional servers are introduced in the framework, there is still a single server at the top of the hierarchy similar to the classical FL.

Both classical and hierarchical FL are vulnerable to single-point-of-failure or man-in-the-middle attacks [29], [30]. However, the additional servers introduce additional potential privacy leakage or model and data manipulation possibilities. In the following section, we are going to discuss the fully decentralized FL framework, where the entire federated learning process runs without a central server.

D. Decentralized FL

FL becomes relevant in scenarios where collaborative models need to be trained and participants prefer not to directly share their data with a central instance. As previously noted, the single-server architectures of FL imply that the entire federated learning process heavily relies on only one instance, requiring trust from all participants. Roy et al. [31] argue that having a single instance that everyone can trust is also a non-trivial problem in many use-cases, such as in the field of healthcare. Therefore, the possibility of a decentralized communication in FL is extensively studied.

Gabrielli et al. [29] identify two classes of decentralizing methods in FL. The first achieves decentralization through distributed computing mechanisms such as peer-to-peer systems,

while the second utilizes blockchain technologies. Both types of decentralized design can be combined with cross-device and cross-silo settings and they do not influence the data distribution properties. Similar to classical FL, decentralized approaches can suffer from statistical heterogeneity, where potential variations in data distributions might occur [32].

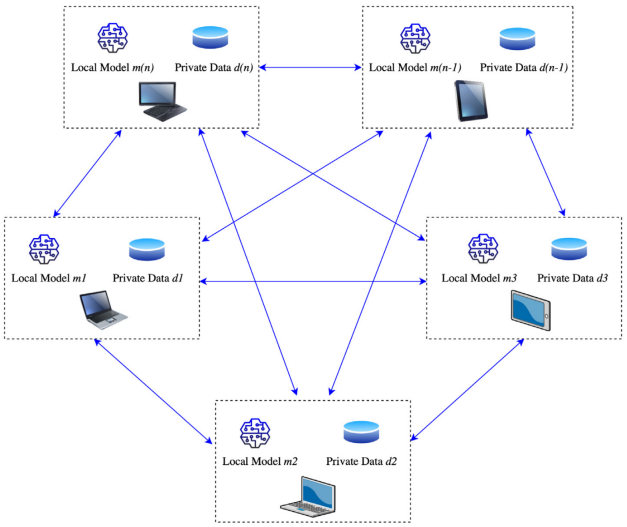

Fig. 2. Decentralized Federated Learning architecture. Adapted from [18].

Figure 2 demonstrates a basic fully-interconnected network topology within the context of a decentralized FL setting.

1) Distributed FL: BrainTorrent is a framework introduced by Roy et al. [31] that facilitates direct communication between clients in a fully-interconnected manner. In each round a client is randomly selected to serve as the server. In this framework a client should be able to initiate an update process at any time by sending a "ping_request" to the rest of the clients.

The fully-interconnected network topology, as seen in the case of BrainTorrent, is not the only known network topology in FL. Beltrán et al. [32] classify and describe three different types of network topologies. They discuss their advantages and disadvantages regarding communication costs, fault tolerance and robustness. In addition to fully-interconnected networks, the paper identifies partially connected networks such as star- or ring-structured networks and node clustering. Node clustering in decentralized FL is organized similarly to cluster-based FL with a central server and shares similar advantages and disadvantages. In contrast to cluster-based FL, there is not a single coordinating server, but rather multiple coordinating nodes. In comparison to centralized FL, Beltrán et al. [32] identify several advantages of distributed FL, including improved fault tolerance and trust issues as well as a reduction of the network bottleneck issue by evenly distributing communication and workload among participating nodes. Peer-to-peer systems also have the advantage of being self-organized to form an overlay network that enables self-scaling federation in FL.

However, both Beltrán et al. [32] and Gabrielli et al. [29] argue that peer-to-peer systems also require improvements and face similar problems as the centralized approach. Peer-to-peer systems can also face significant communication overhead and the failure of numerous nodes can become a challenge. Decentralized systems are not always able to scale properly and manage nodes optimally in a self-organized manner. A survey of potential solutions to overcome these shortcomings can be found in [29].

Beltrán et al. [32] offer a detailed overview of the possible security and privacy concerns regarding the distributed FL approach. Similar to centralized FL, distributed FL is vulnerable to a number of threats, including malware or adversarial attacks. Since there is no central instance providing coordination and control, detecting malicious nodes becomes difficult. Potentially malicious nodes have more leverage in a decentralized setting. Deployed malware can easily reach numerous nodes, especially in fully- or partially interconnected network topologies. Poisoning attacks can go unnoticed, as malicious nodes can keep their data and aggregations private. Implementing well-known techniques such as differential privacy and secure multiparty computation in a decentralized setting is a demanding task due to significant computational overhead.

2) Blockchain FL: The second decentralizing mechanism gaining increasing interest in FL is blockchain [29]. Qu et al. [33] provide a definition and description of blockchains in the context of FL. Blockchain is a decentralized and distributed ledger technology that facilitates secure and transparent recording of transactions across a network of computers.

In the context of FL, all nodes in the federation are considered equal and each node has the potential to lead the aggregation process and contribute to the public ledger. This occurs after a consensus algorithm ensures that the local model update is verified and validated by all involved participants, making it suitable for integration into the global model. Blockchain provides a decentralized and trustworthy framework for managing the collaborative learning process in FL. Qu et al. [33] also summarize the advantages of using blockchain in the FL setting. They claim that the technology is capable of preventing manipulation of data by malicious nodes through its consensus mechanisms. Blockchain provides an increased fault tolerance and scalability compared to the centralized FL approach. Besides, it can offer anonymity to participants by using pseudo-names in order to ensure an additional degree of privacy.

Qu et al. [33] and Beltrán et al. [29] summarize the training process as follows: Firstly, the participating nodes are divided into clusters of miners and a cluster of end nodes holding the training data where each node is associated with a randomly chosen miner. The end devices train local models for a fixed number of iterations and then upload them to their chosen miner. All miners verify the reliability of the model parameters so that the verified data can be stored in the potential block of the miner until all other model parameters from the other end devices are also verified and stored in their respective blocks. After the consensus algorithm is executed and a winning miner is allowed to generate a candidate block all other miners must

stop competing for the block-generating opportunity. Instead, they append the candidate block to their local ledger. The winning miner is then permitted to aggregate all the local models and write the new global parameters into a block that is appended to the public ledger. This public ledger is accessible for download by all end devices in the new training round. Beltrán et al. [29] provide a broad overview of the implementation of various consensus algorithms in the FL field. They explore how these algorithms benefit FL and outline their associated challenges.

Blockchain is a promising technology that can provide decentralization in various FL scenarios. Li, Hu and Wang [30] discuss the benefits of achieving a single-server-free architecture in the field of hierarchical FL using blockchain. However, due to its fundamental design, blockchain is associated with enormous computational overhead and energy consumption [30], [33].

IV. Preserving privacy in federated learning

Privacy is perceived as one of the main attractions of federated learning. However, it is a complex objective demanding for a thorough analysis and sophisticated tools in order to provide privacy guarantees. It is important to regard the learning system as composed of several units and actors that act independently of each other or in a joint manner. Thus, some privacy risks might be solved through improvements of the technology in single parts of the system, while others can only be mitigated by cross-disciplinary efforts. Most techniques that are applied to address privacy concerns are not very recent discoveries per se [34]. However, there are a lot of emerging FL technologies that combine several old and new paradigms, which are presented in the following.

A. Actors and threat models

We view the FL system as composed of three major actors, namely: the client, the server and the system administrator. According to [10] the client is someone who has root-access to the client device, whereas the server is depicted by an entity that has root-access to the server. In addition, the system administrator is someone who has white-box access to the learning model as well as information in form of the model output, iterates of it or the update-parameters.

Rodríguez-Barroso et. al. [35] describe the extent of knowledge among different participants in a FL federation and explain how this knowledge might be misused in potential attacks. In the field of federated learning research, it is commonly assumed that participants, particularly the server, are honest-but-curious rather than malicious. Honest-but-curious attackers primarily attempt to obtain private information about other participants, while malicious attackers aim to corrupt the model or inject a secondary task. Although there may be malicious nodes in the federation, our primary focus is on the privacy implications of potential attacks. Therefore, we maintain the assumption that attackers are not malicious, but rather honest-but-curious.

Rodríguez-Barroso et. al. [35] categorizes knowledge in the federation as follows: Client-side knowledge represents white-box access to the aggregated model, white-box access to the client's locally trained model, and access to the owned client's dataset. Server-side knowledge represents white-box access to the aggregated model after each communication round, white-box access to the clients' gradients, access to the identifiers of the clients aggregated in each communication round, and access to the labels owned by each client and the size of their dataset.

The paper also categorizes three privacy-related attacks: Reconstruction attacks, which attempt to recover clients' local datasets. Membership inference attacks, which aim to determine if the local data of a particular client was used during the training of the model. Property inference attacks, which try to extract specific properties of the clients. All three types of attacks can be executed with client-side knowledge, but if the attacker has also server-side knowledge, the probability of success and the impact of the attack increase [35].

The described attacks threaten not only the classical FL approach but also its modifications. While the shift from a single-server architecture to a multiple-server architecture, as seen in hierarchical FL or decentralized FL soften the reliability problem of a single point of failure, the knowledge of some clients is extended to server-side knowledge. In cluster-based FL server-side knowledge is extended due to the additional data transferred to the single server for proper selection and cluster-building. In Hierarchical FL multiple servers are introduced, each possessing server-side knowledge. In decentralized FL, multiple nodes have not only server-side knowledge but, as clients themselves, also client-side knowledge.

In the following sections, some privacy-preserving measures are introduced that address the described privacy risks. We classify privacy risks into three categories aligning with Kairouz et al. [10]. Firstly, there is an interest in identifying the participating entities and understanding the information flows between them. In this context, techniques like secure computation [36], [37] and vertical architectures with intermediate third parties [17] are highly important.

Secondly, a focus lies on the amount of data that each participating actor is sharing. For example, when training a predictive text model based on typed texts of an user, information about the frequency of certain word combinations or the relevance of certain suggestions might be sufficient enough compared to sending the actual sentences, where sensitive information such as passwords can be revealed. Hereby, differential privacy or clustered approaches are a crucial element to ensure privacy [13], [18], [38].

Lastly, the issue of verifiability poses some challenges to the FL system. More specifically, proving that every entity performs their tasks correctly without disclosing private data. Approaches that are applied here are e.g. trusted execution environments (TEE) [39] and zero-knowledge proofs (ZKP) [40].

B. Secure computation and other privacy-preserving techniques

Secure computation or multi party computation is a field of computer cryptography [36]. It comprises multiple parties that collectively collaborate by calculating a function with a shared input without disclosing information about that input and intermediate results to the outside [37]. In an ideal scenario, the only information revealed to outside parties is the output of that function. In the following, we will look at algorithms and techniques such as homomorphic encryption that incorporate secure computation and show how they are intertwined with federated learning.

1) Homomorphic encryption: Homomorphic encryption provides strong privacy protection through computations that are performed on encrypted data without decrypting it [41]. It is a beneficial technique for FL frameworks as it allows multiple parties to collaboratively train machine learning models on their individual data sets while keeping the data itself encrypted [17]. Especially, mathematical operations like addition or multiplication can be performed on encrypted data in a similar manner than on non-encrypted data. Interestingly, Phong et al. [42] show that homomorphic encryption achieves a similar performance compared to a training with the same unencrypted data set. Due to potential ciphertext attacks, it is a common practice to renew secret keys on a regular basis. Therefore, a downside of homomorphic encryption is additional overhead and complexity to the FL system [13].

2) Differential privacy: Differential privacy is a mechanism that aims to learn as much as possible from a large data set, while protecting individual data [43]. This can be achieved through techniques like randomization, thereby adding noise to a data set. In federated learning, differential privacy aims to include uncertainty into the model trying to obscure the contribution of an individual user to the entire decentralized learning procedure [10]. Although, the orchestrating server would usually be responsible for realizing the differential privacy mechanism, recent research aims to find new approaches in order to reduce the need for a trusted centralized party.

One such way is local differential privacy, where each participant in the model applies a private function to their data before propagating it to the central server. However, applying a fully local differential privacy model significantly limits the utility of the model, resulting in an ongoing challenge to find the balance between privacy and utility. This leads to emerging concepts like distributed differential privacy and hybrid approaches that are incorporated into FL system.

In distributed differential privacy each client sends a minimal encoded report to an intermediate secure computation function. Subsequently, the result of that function is transmitted to the central server. Prior to inspection from the server-side all differential privacy requirements should be fulfilled. In almost every case, another intermediate trusted party is introduced. Two strategies for achieving distributed differential privacy involve the utilization of secure aggregation and secure shuffling techniques [10].

Another promising technique is hybrid differential privacy [44]. This approach tries to partition all participating clients based on their trust model preference. Therefore, not only a binary choice between a fully local or a fully distributed model for all clients is possible, but also a combination of multiple models considering all varying preferences. However, introducing noise leads to the obvious downside of affecting the model accuracy.

C. Verifiability

1) Zero knowledge proofs: A zero knowledge proof is a cryptographic concept that allows a prover to demonstrate to a verifier the truth of some information without revealing the secret itself [45]. ZKPs provide a solution to the verifiability issue of private data, therefore being an important technique in privacy-preserving authentication. A prominent example illustrating the fundamental concept of ZKP is 'Ali Baba's Cave' as presented by Quisquater et al. [46]. Imagine Alice wants to prove to Bob that she knows some secret passphrase in order to enter a hidden cave. She does not want to reveal the passphrase itself. Alice could enter the cave, close the entrance, and then come out again. By repeating this process multiple times, Bob can be increasingly convinced that Alice knows the passphrase. It is important to mention that Alice never communicates the secret information openly with any party during this process. Therefore, Bob gains no knowledge of the secret at any point in time.
In summary, ZKP aim for three conditions: completeness, soundness and zero-knowledge. If a statement being proven is true and a honest verifier will be convinced of that truth, a ZKP is complete. In contrast, if a statement being proven is false and both parties follow the protocol, the verifier will refuse the proof. Lastly, no knowledge can be learned from the interaction of both parties except that the statement is true, if and only if the statement is actually true [40]. On the downside, zero-knowledge federated learning (zkFL) introduces performance trade-offs. For example, communication costs are increased among all parties and the verification burden put on the client might result in a new challenge.

2) Trusted execution environments: Hardware-based TEEs provide strong confidentiality and integrity guarantees to any data or process that is executed within these secure memory areas. Moreover, TEEs often support mechanisms for verifying the integrity of the code and data within the enclave [10]. As TEEs are more and more integrated in high-end and mobile devices they are a promising technique for FL. More specifically, TEEs help in securing model updates, because parameters could only be updated within the trusted area. Hence, preventing potential adversaries from intercepting or modifying the model during the update process [39].

Mo et al. [39] demonstrate a feasible FL solution incorporating TEE technology by implementing an algorithm using both, server-side TEE (with Intel SGX) and client-side TEE (with Arm TrustZone). However, due to computation overhead and limited memory access the guarantees of TEEs come at

a price. For example, making it hard to fit the whole training process into the designated memory cells [47].

V. Conclusion

In this survey we gave an overview of noteworthy literature in the field of federated learning. Moreover, we presented the general principles of FL along with common techniques and modifications of the classical approach. Every introduced technology addresses a very specific problem. In addition, we conducted a detailed analysis of privacy issues, prevailing techniques to mitigate them and divided them into three categories: secure computation, verifiability and other privacy-preserving techniques.

We conclude that federated learning is a new field with established roots. It leverages proven concepts and techniques and incorporates them into one system. By doing so federated learning increases privacy and makes it harder for adversaries to attack. However, this comes at a cost. We identified five factors that are usually affected: additional model complexity, computational overhead, decreased model utility, reduced model accuracy and introducing trusted intermediate parties. Future research might focus on the balance of these factors and find hybrid models that are suitable for specific use-cases.

References

[1] Trevor Darrell, Marius Kloft, Massimiliano Pontil, Gunnar Rätsch, and Erik Rodner. Machine Learning with Interdependent and Non-identically Distributed Data (Dagstuhl Seminar 15152). *Dagstuhl Reports*, 5(4):18–55, 2015.

[2] H. B. McMahan and D. Ramage. Federated learning: Collaborative machine learning without centralized training data, 2017.

[3] Badra Souhila Guendouzi, Samir Ouchani, Hiba EL Assaad, and Madeleine EL Zaher. A systematic review of federated learning: Challenges, aggregation methods, and development tools. *Journal of Network and Computer Applications*, 220:103714, 2023.

[4] Kallista A. Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloé Kiddon, Jakub Konečný, Stefano Mazzocchi, H. Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. Towards federated learning at scale: System design. *CoRR*, abs/1902.01046, 2019.

[5] Jakub Konečný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *CoRR*, abs/1610.02527, 2016.

[6] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction, 2019.

[7] Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Françoise Beaufays. Applied federated learning: Improving google keyboard query suggestions. *CoRR*, abs/1812.02903, 2018.

[8] Yi Liu, Li Zhang, Ning Ge, and Guanghao Li. A systematic literature review on federated learning: From a model quality perspective, 2020.

[9] Ognjen Rudovic, Nicolas Tobis, Sebastian Kaltwang, Björn W. Schuller, Daniel Rueckert, Jeffrey F. Cohn, and Rosalind W. Picard. Personalized federated deep learning for pain estimation from face images. *ArXiv*, abs/2101.04800, 2021.

[10] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista A. Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *CoRR*, abs/1912.04977, 2019.

[11] Shaoxiong Ji, Teemu Saravirta, Shirui Pan, Guodong Long, and Anwar Walid. Emerging trends in federated learning: From model fusion to federated x learning, 02 2021.

[12] Rémi Gosselin, Loïc Vieu, Faiza Loukil, and Alexandre Benoit. Privacy and security in federated learning: A survey. *Applied Sciences*, 12(19), 2022.

[13] Ahmed El Ouadrhiri and Ahmed Abdelhadi. Differential privacy for deep and federated learning: A survey. *IEEE Access*, 10:22359–22380, 2022.

[14] Xin Gu, Fariza Sabrina, Zongwen Fan, and Shaleeza Sohail. A review of privacy enhancement methods for federated learning in healthcare systems. *International Journal of Environmental Research and Public Health*, 20:6539, 08 2023.

[15] Nuria Rodríguez-Barroso, Daniel Jiménez-López, M. Victoria Luzón, Francisco Herrera, and Eugenio Martínez-Cámara. Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges. *Information Fusion*, 90:148–173, February 2023.

[16] Yashothara Shanmugarasa, Hye-young Paik, Salil S. Kanhere, and Liming Zhu. A systematic review of federated learning from clients'perspective: challenges and solutions. *Artificial Intelligence Review*, 56(2):1773–1827, 2023.

[17] Ziyao Liu, Jiale Guo, Wenzhuo Yang, Jiani Fan, Kwok-Yan Lam, and Jun Zhao. Privacy-preserving aggregation in federated learning: A survey. *IEEE Transactions on Big Data*, pages 1–20, 2022.

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

[19] Wael Issa, Nour Moustafa, Benjamin Turnbull, Nasrin Sohrabi, and Zahir Tari. Blockchain-based federated learning for securing internet of things: A comprehensive survey. *ACM Comput. Surv.*, 55(9), jan 2023.

[20] Poongodi Manoharan, Ranjan Walia, Celestine Iwendi, Tariq Ahamed Ahanger, S. T. Suganthi, M. M. Kamruzzaman, Sami Bourouis, Wajdi Alhakami, and Mounir Hamdi. Svm-based generative adverserial networks for federated learning and edge computing attack model and outpoising. *Expert Systems*, 40(5):e13072, 2023.

[21] Franziska Boenisch, Adam Dziedzic, Roei Schuster, Ali Shahin Shamsabadi, Ilia Shumailov, and Nicolas Papernot. When the curious abandon honesty: Federated learning is not private, 2023.

[22] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications, 2019.

[23] Lei Fu, Huanle Zhang, Ge Gao, Mi Zhang, and Xin Liu. Client selection in federated learning: Principles, challenges, and opportunities, 2023.

[24] Joel Wolfrath, Nikhil Sreekumar, Dhruv Kumar, Yuanli Wang, and Abhishek Chandra. Haccs: Heterogeneity-aware clustered client selection for accelerated federated learning. In *2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 985–995, 2022.

[25] Rahul Atul Bhope, K. R. Jayaram, Nalini Venkatasubramanian, Ashish Verma, and Gegi Thomas. Flips: Federated learning using intelligent participant selection, 2023.

[26] Chenhao Xu, Youyang Qu, Yong Xiang, and Longxiang Gao. Asynchronous federated learning on heterogeneous devices: A survey, 2023.

[27] Mang Ye, Xiuwen Fang, Bo Du, Pong C. Yuen, and Dacheng Tao. Heterogeneous federated learning: State-of-the-art and research challenges, 2023.

[28] Lumin Liu, Jun Zhang, S.H. Song, and Khaled B. Letaief. Client-edge-cloud hierarchical federated learning. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pages 1–6, 2020.

[29] Edoardo Gabrielli, Giovanni Pica, and Gabriele Tolomei. A survey on decentralized federated learning, 2023.

[30] Shengyang Li, Qin Hu, and Zhilin Wang. Pofel: Energy-efficient consensus for blockchain-based hierarchical federated learning, 2023.

[31] Abhijit Guha Roy, Shayan Siddiqui, Sebastian Pölsterl, Nassir Navab, and Christian Wachinger. Braintorrent: A peer-to-peer environment for decentralized federated learning, 2019.

[32] Enrique Tomás Martínez Beltrán, Mario Quiles Pérez, Pedro Miguel Sánchez Sánchez, Sergio López Bernal, Gérôme Bovet,

Manuel Gil Pérez, Gregorio Martínez Pérez, and Alberto Huertas Celdrán. Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges. *IEEE Communications Surveys & Tutorials*, 25(4):2983–3013, 2023.

[33] Youyang Qu, Md Palash Uddin, Chenquan Gan, Yong Xiang, Longxiang Gao, and John Yearwood. Blockchain-enabled federated learning: A survey. *ACM Computing Surveys*, 55, 03 2022.

[34] Nguyen Truong, Kai Sun, Siyao Wang, Florian Guitton, and YiKe Guo. Privacy preservation in federated learning: An insightful survey from the gdpr perspective. *Computers & Security*, 110:102402, 2021.

[35] Nuria Rodríguez-Barroso, Daniel Jiménez-López, M. Victoria Luzón, Francisco Herrera, and Eugenio Martínez-Cámara. Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges. *Information Fusion*, 90:148–173, 2023.

[36] Andrew Chi-Chih Yao. How to generate and exchange secrets. In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*, pages 162–167, 1986.

[37] Vaikkunth Mugunthan and Antigoni Polychroniadou. Smpai: Secure multi-party computation for federated learning, 2019.

[38] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Transactions on Neural Networks and Learning Systems*, 32(8):3710–3722, 2021.

[39] Fan Mo, Hamed Haddadi, Kleomenis Katevas, Eduard Marin, Diego Perino, and Nicolas Kourtellis. Ppfl: Privacy-preserving federated learning with trusted execution environments. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '21, page 94–108, New York, NY, USA, 2021. Association for Computing Machinery.

[40] Zhipeng Wang, Nanqing Dong, Jiahao Sun, and William Knottenbelt. zkfl: Zero-knowledge proof-based gradient aggregation for federated learning. *arXiv preprint arXiv:2310.02554*, 2023.

[41] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, page 169–178, New York, NY, USA, 2009. Association for Computing Machinery.

[42] Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security*, 13(5):1333–1345, 2018.

[43] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, pages 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[44] Brendan Avent, Aleksandra Korolova, David Zeber, Torgeir Hovden, and Benjamin Livshits. BLENDER: Enabling local search with a hybrid differential privacy model. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 747–764, Vancouver, BC, August 2017. USENIX Association.

[45] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[46] Jean-Jacques Quisquater, Myriam Quisquater, Muriel Quisquater, Michaël Quisquater, Louis Guillou, Marie Annick Guillou, Gaïd Guillou, Anna Guillou, Gwenolé Guillou, and Soazig Guillou. How to explain zero-knowledge protocols to your children. In Gilles Brassard, editor, *Advances in Cryptology — CRYPTO' 89 Proceedings*, pages 628–631. Springer New York, 1990.

[47] Florian Tramèr and Dan Boneh. Slalom: Fast, verifiable and private execution of neural networks in trusted hardware, 2019.