SURVEY ON FEDERATED LEARNING TOWARDS PRIVACY PRESERVING AI

Sheela Raju Kurupathi[1] and Wolfgang Maass[1, 2]

[1]German Research Center for Artificial Intelligence, Saarbrücken, Germany
[2]Saarland University, Saarbrücken, Germany

ABSTRACT

One of the significant challenges of Artificial Intelligence (AI) and Machine learning models is to preserve data privacy and to ensure data security. Addressing this problem lead to the application of Federated Learning (FL) mechanism towards preserving data privacy. Preserving user privacy in the European Union (EU) has to abide by the General Data Protection Regulation (GDPR). Therefore, exploring the machine learning models for preserving data privacy has to take into consideration of GDPR. In this paper, we present in detail understanding of Federated Machine Learning, various federated architectures along with different privacy-preserving mechanisms. The main goal of this survey work is to highlight the existing privacy techniques and also propose applications of Federated Learning in Industries. Finally, we also depict how Federated Learning is an emerging area of future research that would bring a new era in AI and Machine learning.

KEYWORDS

Federated Learning, Artificial Intelligence, Machine Learning, Privacy, Security, Distributed Learning.

1. INTRODUCTION

Due to the emergence of AI and Machine learning over the past few decades, there has been significant progress in various domains like Robotics, Computer Vision and Gaming Applications. One of the major concerns is to preserve data privacy. Preserving the data privacy is of utmost importance in these days as the data is created in abundance every day. Data leaks on publicly available data and the private data of the companies lead to alarming increase towards data privacy. Utilizing the data which is isolated as data islands by maintaining specific privacy standards is very crucial for better data security. Misusing the personal data of the user may cause overhead to the user forcing him not to enclose his personal details. Even in the companies and industries, it is essential to protect data from data leaks as it would lead to grave consequences for the company. The data leaks, in turn, would affect the financial and commercial aspects of the company on a large scale leading to huge losses. One of the well-known standards for ensuring data privacy is the General Data Protection Regulation (GDPR) [1, 2] in the European Union. The GDPR was proposed in 2018 to ensure data privacy of every user, which in turn motivates to use AI and machine learning frameworks adapting to this standard while using data.

Many machine learning and AI models need sufficient data for training and to produce high-quality models. Although the models need to use user data if they need to build good prediction models for the user, there should be a way to ensure user privacy. Few organizations need to exchange data for working collaboratively for better performance of the companies, in turn,

Dhinaharan Nagamalai et al. (Eds): CSEIT, WiMoNe, NCS, CIoT, CMLA, DMSE, NLPD - 2020
pp. 235-253, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101120

ensuring the data privacy and confidentiality. In edge devices where users interact with different applications like in mobile phones, there is an ample amount of private data related to the user which is being exposed every day. To solve the problem of using data to train models, ensuring data privacy, we have a new approach known as Federated Learning (FL) [3]. The term Federated Learning (FL) was introduced by McMahan et al. [4] in 2016. Federated Learning is a collaborative Machine learning technique where the machine learning models are trained on edge devices (like mobiles) instead of a central server to ensure data privacy. The data is not exchanged between the devices. However, only the model updates (gradient updates) are sent to the server to build a global model using the aggregated gradients from all the computing edge devices. Thus, the server has no information about the raw data that the edge devices have been trained on, maximizing the data privacy of the users. Federated Learning has been evolving over the past few years due to the increasing demand for data privacy and security. It mitigates the risk of data privacy in comparison to centralized machine learning approaches. It also reduces the cost involved in traditional and centralized machine learning approaches.

The rest of the survey work is organized as follows: Section 2 details various related works in the area of Federated Learning. Section 3 details about Federated Learning, its working principle, training process, categorization of Federated Learning architectures along with various implementation frameworks, and Section 4 elaborates more about the privacy- preserving mechanisms in FL. Section 5 describes the application of Federated Learning in Industries along with its drawbacks, and in Section 6, we discuss in detail about the privacy- preserving aspect of Federated Learning. Section 7 concludes the survey work and suggests a few possible directions for the future area of research.

2. RELATED LITERATURE

As the data is vastly distributed over many devices, it is crucial for machine learning and AI models to access the data reliably for building efficient models. The goal of many research communities in the fields of Machine Learning, Artificial Intelligence, Cryptography and Distributed Systems has been to learn from the massively distributed data ensuring data security and privacy. Federated Learning is the recent trends for training the machine learning models in a decentralized way without having any information about the raw data except for the updated gradients from the client models. Federated Learning focusses on the edge and mobile computing [4, 5] devices and then extended its application to large scale production systems. Now industries are extensively using Federated Learning as part of their production systems for better product and profit generation on a large scale.

The data scattered everywhere as data islands need to be integrated on a large scale for useful application of AI models. It is challenging to integrate the data from these islands as it gives a cost overhead. Federated Learning has been the saviour for reducing the cost for data integration through execution of AI models on the data available on edge computing devices. Federated Learning is being used by Google in its Gboard mobile keyboard [6, 7, 8, 9, 10]. They also implemented a few of the features using Federated Learning in Android Messages [11] and Pixel phones [12]. Even Apple is using Federated Learning in iOS 13 [13], for various applications like the vocal classifier for “Hey Siri” [14] and QuickType keyboard. Other applications include Federated Learning for medical research [15] and the detection of hot words [16].

Currently, much of the research work is being focussed in FL, due to the privacy-preserving aspect of Federated Learning. Clifton and Vaidya proposed secure k-means [17], secure association mining rules [18], and a naive Bayes classifier [19] for vertically partitioned data. The authors of [20] implemented a privacy-preserving protocol using homomorphic encryption for

applying linear regression on horizontally partitioned data. The authors of [21, 22] have proposed a linear regression approach for vertically partitioned data. FL directly solved the linear regression problem. The authors of [23] have approached the problem with Stochastic Gradient Descent (SGD) and also proposed privacy-preserving protocols for neural networks and logistic regression. The authors of [24] proposed a novel algorithm for association rules on horizontally partitioned data. Secure Support Vector Machines (SVM) algorithms have been implemented for horizontally partitioned data [25] and vertically partitioned data [26]. The authors of [27] proposed various secure protocols for multi-party linear regression and classification. The authors of [28] proposed efficient, secure multi-party gradient descent methods. All these works used Secure Multiparty Computation (SMC) [29, 30] for preserving privacy and ensuring security.
The authors of [31] proposed a secure logistic regression protocol based on homomorphic encryption. Shokri and Shmatikov [32] proposed training of neural networks on horizontally partitioned data with exchanges of updated parameters. The authors of [33] used homomorphic encryption to enhance the security of the entire system and preserve the privacy of gradients. With recent trends in machine learning and AI, privacy-preserving neural networks are also one of the research interest [34, 35, 36, 37]. Therefore, building a decentralized system with collaborative machine learning models and ensuring data privacy is one of the crucial aspects of many industries.

3. FEDERATED MACHINE LEARNING

The term Federated Learning refers to a decentralized machine learning setting where all the participating clients train a shared global model without exposing the data to the central server. Only the model updates from each participating device are sent to the server. The model updates are then aggregated based on Federated Averaging mechanism [5] to obtain an efficient global model. Therefore, it is a collaborative machine learning where all the clients contribute the model updates to achieve a common learning objective. FL allows for smarter models, lower latency, and less power consumption, all while ensuring privacy. It is also used in distributed architectures where machine learning needs to be integrated into them.

3.1. Definitions

To understand the term Federated Learning, it is essential to know the terms distributed learning [38, 39], centralized and decentralized Federated Learning [5, 40, 41].

- **Distributed Machine Learning:** In distributed machine learning, we train a model on a large dataset. Here, the clients are computing nodes in a single cluster or datacenter. All the clients can access any part of the dataset. The data is distributed onto multiple computing nodes in a datacentre. Distributed learning aims at parallelization of computing power through the distribution of data or model.
- **Centralized Federated Learning:** It has a central server which is used to orchestrate the entire training process and coordinate all the participating nodes during the learning process. The central server is responsible for the selection of nodes initially before the training starts and the aggregation of the received model updates. The server may become a bottleneck here as all the selected nodes have to send updates to a single entity.
- **Fully Decentralized/ Peer-to-Peer Learning:** It has a peer-to-peer topology [42], where every participating client can talk to the other participating clients. It has a possibly dynamic connectivity graph structure without any central orchestration.
- **Decentralized Federated Learning:** In this Federated Learning setting, the computing nodes can coordinate between themselves to compute the global model. As the model

updates are exchanged only between interconnected nodes without the orchestration of the central server, this setting prevents single-point failures.

3.2. Federated Learning Life Cycle

Federated Learning ensures secure collaborative machine learning with the decentralization of data. It uses hub-and-spoke topology, with the hub representing a coordinating service provider and the spokes connecting to clients. Despite preserving privacy, it has many challenges like if the server fails, then the global update of the model would be difficult. For better performance of the FL systems, the federated system must be resilient to failures.

The clients participating in the FL process can be multiple clients or multiple organizations. Based on the participating client, we have two Federated Learning settings, namely Cross-device and Cross-silo Federated Learning [43]. In Cross-device Federated Learning, multiple clients like mobile devices are stateless and highly unreliable. Only a few of the clients would be available at any point in time, thus making the computation and communication difficult. In Cross-silo Federated Learning, the clients are different organizations (medical or financial domains) participate in the FL process. This setting is typically limited to a hundred organizations while Cross-device setting can have an extensively large number of clients. In both Cross-device and Cross-silo setting, the data is decentralized and local to each client. The server acts as a central authority for organizing the training process, and it never sees the raw data of the participating clients. In this paper, we mainly consider the cross-device federated setting for explaining the Federated Learning life cycle and the training process.

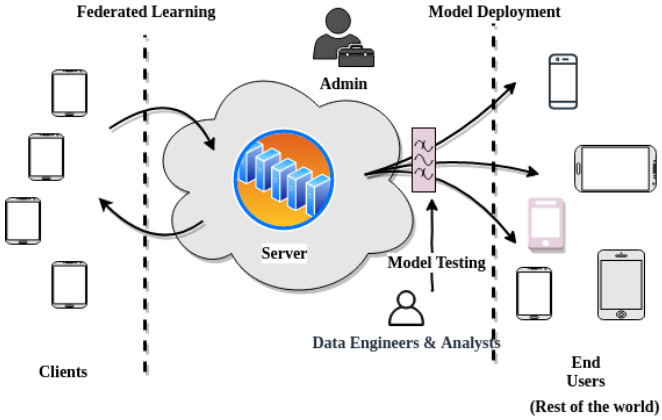

Figure 1. Life cycle of a Federated Learning (FL) model.

Initially, in the FL process, a model engineer develops a model for a particular application. In natural language processing, a domain expert may develop a prediction model for next word prediction to use in a virtual keyboard application. Figure 1 depicts the primary components and actors involved in the FL process. A typical workflow of the FL model can be realized, as shown in Figure 2. The life cycle of a Federated Learning (FL) model consists of six stages, as shown in Figure 2.

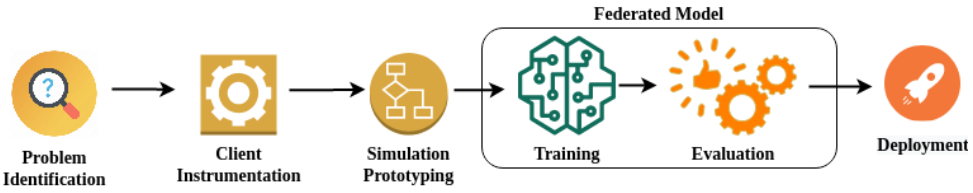

Figure 2. Stages in Life cycle of a Federated Learning (FL) model.

1. **Problem identification:** In this stage, the model engineer first identifies a problem that needs to be solved with Federated Learning.

2. **Client instrumentation:** The clients store the necessary training data locally. In a few cases, additional data or metadata might need to be maintained; for example, the labels for a supervised learning task.

3. **Simulation prototyping:** The model architectures are prototyped by the model engineer and then test's learning hyperparameters in a Federated Learning simulation using a proxy dataset.

4. **Federated model training:** Usually, all the federated training tasks are initiated to train different variations of the model. We could also use different optimization hyperparameters for further training.

5. **Federated model evaluation:** Once the Federated Learning tasks have been trained sufficiently, the models are then analyzed, and the best candidates are selected. The analysis depends on various metrics computed on standard datasets in the datacenter. Federated evaluation is carried out on local client data wherein the models are pushed to held-out clients.

6. **Deployment:** Once a good model is selected, it then goes through a standard model launch process, including live A/B testing, manual quality assurance and a staged rollout. The application owner sets the specific launch process for the selected model and is independent of how the model is trained.

3.3. Federated Learning Training Process

Federated Learning decouples the ability to do machine learning from the need to store the data in the central server or cloud. We could make use of local models to make predictions on mobile devices by bringing model training to the device as well. From Figure 3, the device first downloads the current model, improves it by learning from data on the phone, and then summarizes the changes as a small, focused update. Only this focused update is sent to the server through encrypted communication. Then immediately averaged with other user updates to improve the shared global model. Since all the training data remains on the device, no individual updates are stored in the server.

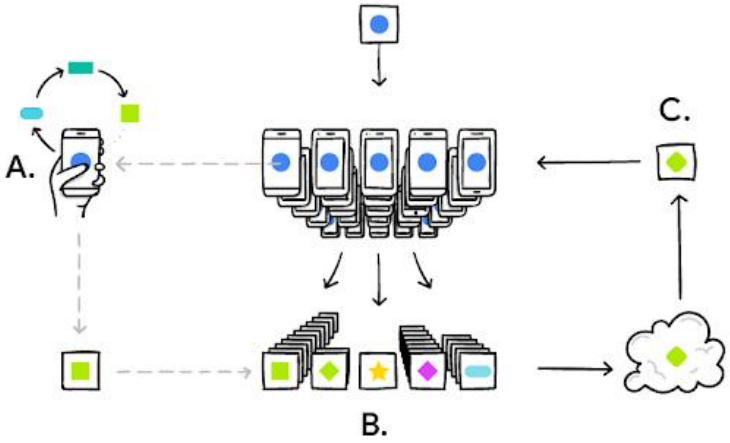

Figure 3. Process of Federated Learning (FL) model training.
(A) represents many users' updates are aggregated
(B) to form a consensus change (C) to the shared model, after which the procedure is repeated. [3]

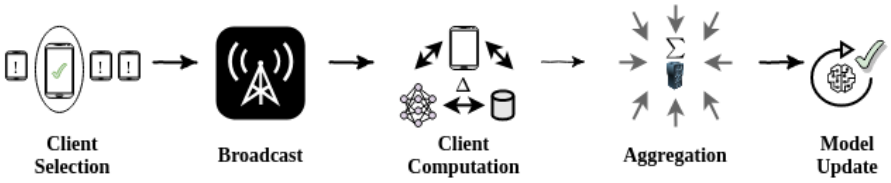

Figure 4. Stages in Federated Learning (FL) training process.

Federated Learning (FL) training process consists of five steps, as shown in Figure 4. A central server orchestrates the training process in FL setting, by iterative execution of the five steps shown in Figure 4 until the training process is stopped.

1. **Client selection:** First, the server samples from a set of participating clients meeting eligibility requirements. Mobile phones would only check in to the server if they are plugged in, and idle, to avoid impact on the device user.

2. **Broadcast:** In this step, the selected clients download the current model weights and a training program from the central server. For example, a training program can be a TensorFlow graph [44].

3. **Client computation:** In this step, each selected device locally computes a focused update to the model by executing the training program, like running SGD on the local data as in Federated Averaging algorithm.

4. **Aggregation:** The central server collects an aggregate of the device focused updates for efficiency. This step also includes other techniques like secure aggregation for added privacy, noise addition, a lossy compression of aggregates for communication efficiency and update clipping for differential privacy.

5. **Model update:** Finally, in this step, the server locally updates the shared model based on the aggregated update computed from all the participating clients in the current round. For

better performance of the central global machine learning model, FL relies on an iterative process of model updates.

3.4. Federated Learning Categorization

The data used for training the Federated Learning (FL) is non-identical as the data is on multiple devices. Based on how the data is distributed across multiple participating devices in the Federated Learning (FL) process, we classify FL into three different categories as Horizontal FL, Vertical FL and Federated Transfer Learning. The central authority to execute the final global update of the model based on the model updates from the clients plays a vital role in FL. Based on whether there is a central authority or not for coordination, FL can be classified as Centralized FL and Decentralized FL as already discussed. The clients participating in the FL process may be mobile devices or can be organizations. Few organizations need to collaborate to implement effective practical solutions which are profitable to the organizations involving as a whole. Therefore, we can classify FL into Cross-silo and Cross-device Federated Learning when the participating client is organization and mobile device, respectively. We will discuss in detail about FL categorization based on data partitioning.

Horizontal Federated Learning (HFL): In a Horizontal Federated Learning (HFL) system, only the central server can compromise the privacy of data participants. Horizontal Federated Learning (HFL), also known as sample-based Federated Learning (FL), is used in the scenarios in which datasets share the same feature space but different space in samples, as shown in Figure 5.

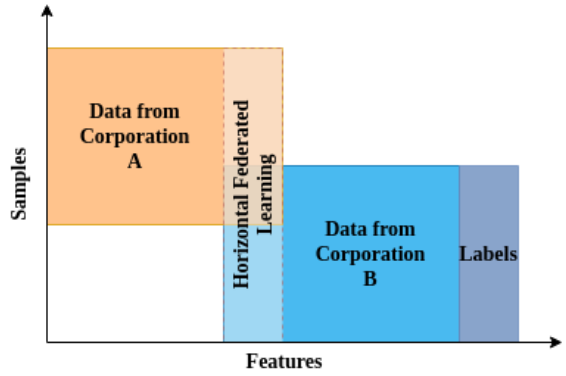

Figure 5. Horizontal Federated Learning (HFL) data partitioning.

For example, two regional banks that differ in user groups have a small intersection of users. However, as the business is very similar, they have the same feature spaces. The authors of [45] proposed a collaboratively deep-learning setting wherein participants train independently and share only subsets of parameter updates. Google proposed a Horizontal Federated Learning (HFL) solution for Android phone model updates [46]. In this framework, a single user using an Android phone updates the model parameters locally and then uploads the parameters to the Android cloud. Thus, jointly training the centralized model together with other data owners. A secure aggregation was used to protect the privacy of aggregated user updates, as shown in [47]. The authors of [48] use homomorphic encryption for model parameter aggregation to provide security.

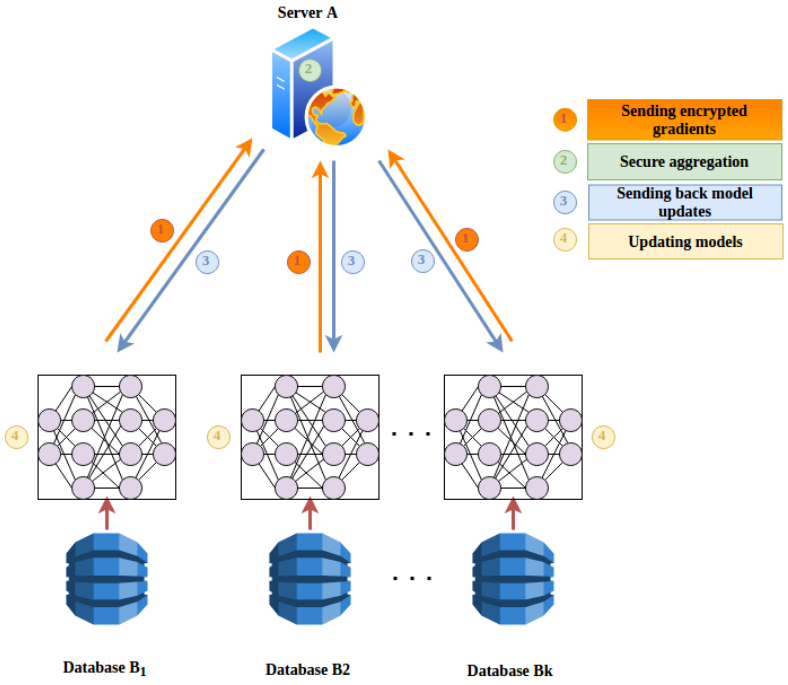

Figure 6. Horizontal Federated Learning (HFL) architecture.

Sample architecture for a horizontal Federated Learning (FL) system is shown in Figure 6. In this system, k participants with the same data structure collaboratively learn a machine-learning model using a parameter or cloud server. It assumes that there is no leakage of information from any participants to the server [48]. The training process of the HFL system usually contains the following four steps.

• **Step 1:** Initially, all the participants locally compute training gradients and then mask selected gradients with differential privacy [49], encryption [48], or secret sharing [47] techniques. Later these masked results are sent to the server.
• **Step 2:** The server then performs secure aggregation without learning any information about any participating client.
• **Step 3:** The server sends the aggregated results to all the participants.
• **Step 4:** Participants update their respective model with the decrypted gradients.
All the steps go through iterations until the loss function converges, thus completing the entire training process.

Vertical Federated Learning (VFL): Vertical Federated Learning (VFL) or feature-based Federated Learning (FL) is applicable to the cases in which two datasets share the same sample ID space but differ in feature space. For example, two different companies, like the bank and the other is an e-commerce company in the same city. Their user sets contain most of the residents of the area, and the intersection of their user space is enormous. However, their feature spaces are very different. Vertically Federated Learning (VFL) aggregates these different features and computes the gradients and training loss in a privacy-preserving manner. It finally builds a model with data from both parties collaboratively. At the end of the learning phase, each party holds only those model parameters associated with its features. Finally, at inference time, the two parties need to collaborate to generate output.

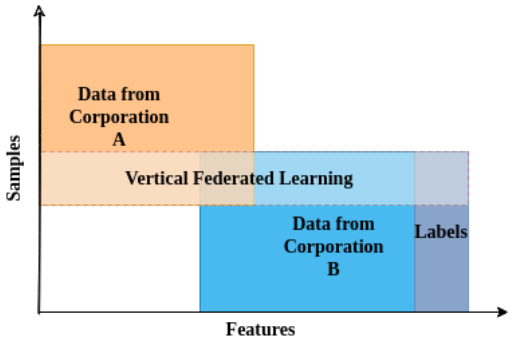

Figure 7. Vertical Federated Learning (VFL) data partitioning.

Many privacy-preserving machine learning algorithms have been proposed for vertically partitioned data, including association rule mining [50], cooperative statistical analysis [51], secure linear regression [52, 53], gradient descent [54] and classification [55]. Two companies A and B would like to train a machine-learning model jointly, and their business systems each have their data. For data privacy and security reasons, companies A and B cannot directly exchange data. During the training process, a third-party collaborator C is involved to ensure the privacy and confidentiality of the data. This Federated Learning (FL) system consists of two parts, as shown in Figure 8.

Part 1. Encrypted entity alignment: As the user groups of the two companies, A and B are different; the system uses the encryption-based user ID alignment techniques [56, 57] to confirm the standard users of both parties without A and B exposing their data. During the entity alignment, the system does not expose users that do not overlap with each other.

Part 2. Encrypted model training: Once the common entities are determined, we can use these common entities' data to train the machine-learning model.

The training process of VFL can be divided into the following four steps, as shown in Figure 8.

- **Step 1:** Initially, Collaborator C creates encryption pairs and sends a public key to A and B.
- **Step 2:** Both A and B encrypt and exchange the intermediate results for gradient and loss calculations.
- **Step 3:** Companies A and B compute the encrypted gradients and add a mask, respectively. Company B also computes an encrypted loss. Both A and B send encrypted values to C.
- **Step 4:** C decrypts and send the decrypted gradients and loss back to A and B. Then A and B unmask the gradients and update the model parameters accordingly.

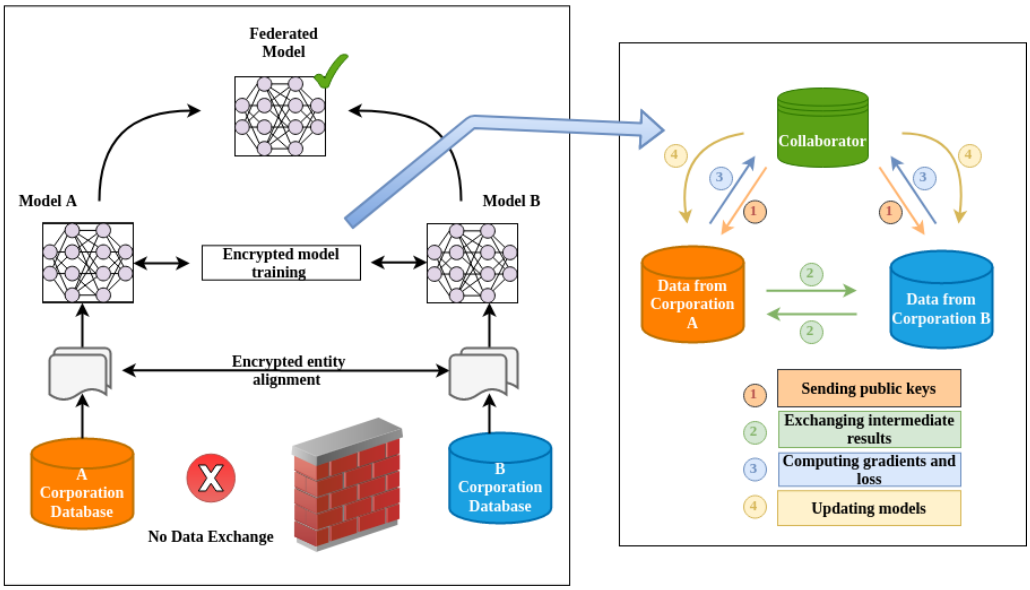

Figure 8. Vertical Federated Learning (VFL) architecture.

Federated Transfer Learning (FTL): Federated transfer learning is used in scenarios in which two datasets differ in both sample and also in feature space. FTL is a vital extension to the existing Federated Learning (FL) systems as it deals with the problems outside the scope of existing Federated Learning (FL) algorithms. For example, if one is a bank located in Russia and the other is an e-commerce company located in Ireland. A small portion of the feature space overlaps from both parties due to geographical restrictions. The architecture of VFL works only for the overlapping dataset. To extend it to the entire sample space, we introduce transfer learning. Typically, transfer learning involves learning a common representation between the features of parties A and B. It minimizes the errors in predicting the labels for the target-domain. At inference time, it still requires both parties to compute the prediction results. Thus, transfer-learning [58] techniques can be applied to provide solutions for the entire sample and feature space under a federated setting. Generally, a common representation is learnt between the two feature spaces using limited common sample sets and then later applied to obtain predictions for only one-side feature samples.

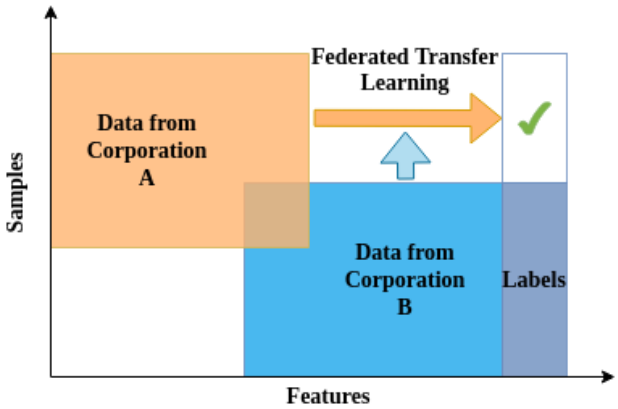

Figure 9. Federated Transfer Learning (FTL) data partitioning.

3.5. Federated Learning Implementation Frameworks

Federated Learning is difficult to implement and deploy in real life due to the heterogeneity in edge computing devices. These devices may have different programming languages, frameworks, and hardware configurations. There are many federated frameworks available to simulate FL algorithms. Few of the available tools and frameworks are TensorFlow Federated [59], PySyft [60], Federated AI Technology Enabler [61], PaddleFL [62], Leaf [63] and Clara Training Framework [64]. TensorFlow Federated (TFF) introduced by Google is an extensible, powerful framework for implementing Federated Learning (FL) research by simulating Federated Learning (FL) computations on realistic proxy datasets. It has Federated Core (FC) API is used for expressing new algorithms, and Federated Learning (FL) API can be used for implemented federated models. PySyft is another open-source library built for Federate Learning (FL) and preserving privacy. It was developed by the OpenMined community which combines these different tools for building secure and private machine learning models. It is built as an extension of well known DL libraries, such as PyTorch, Keras and Tensorflow. Using these popular deep learning frameworks, we can immediately begin to build privacy-preserving applications without having to learn a new Deep Learning framework. Thus, Federated Learning (FL) and other tools could be easily adopted in any application domain for preserving privacy. Therefore, these frameworks are designed to simulate FL in a server environment. However, they do not allow experimentation in a distributed mobile setting for a large number of clients. Another framework Leaf includes a set of open-source federated datasets, an evaluation framework, and a set of reference implementations using for practical federated environments.

4. PRIVACY MECHANISMS IN FEDERATED LEARNING

Privacy is one of the crucial properties of Federated Learning (FL). Therefore, it requires analysis and security models to provide privacy guarantees. In this section, we briefly review various privacy techniques for Federated Learning (FL).

Secure Multiparty Computation (SMC): SMC security models involve multiple parties and provide security proof in a well-defined simulation framework to guarantee that each party knows nothing except its input and output. Here, the parties have zero knowledge about other parties. Zero-knowledge is highly desirable, but this desired property usually requires highly complicated computation protocols and may not be achieved efficiently. In certain exceptional scenarios, disclosure of partial knowledge can be considered acceptable if security guarantees are provided. Therefore, it is possible to build a security model with SMC under lower security requirements in exchange for efficiency.

Differential Privacy: Differential Privacy involves adding noise to the data, or using generalisation methods to hide certain sensitive attributes until the third party cannot distinguish the individual, thereby making the data impossible to be restored to protect user privacy. The DP method is lossy as machine learning models are built after noise is injected, which can reduce much performance in prediction accuracy.

- ***Local Differential Privacy:*** Differential privacy can be achieved without requiring trust in a centralised server by having each client apply a differentially private transformation to their data before sharing it with the server.
- ***Distributed Differential Privacy:*** Here, the clients first compute and encode a minimal, focused report, and then send the encoded reports to a secure computation function, whose output is available to the central server. The output already satisfies differential privacy requirements by the time the central server can inspect it. The encoding is done

to help maintain privacy on the clients. This privacy-preserving technique can be implemented via secure aggregations and secure shuffling.

- ***Hybrid Differential Privacy:*** This combines multiple trust models by partitioning users by their trust model preferences. There are two options before the advent of HDP like most-trusting and the least trusting model.

Homomorphic Encryption (HE): Homomorphic encryption is adopted to protect user data privacy through an exchange of parameters under the encryption mechanism. Unlike differential privacy protection, the data and the model itself are not transmitted, nor can they be guessed by the other party's data. Homomorphic encryption (HE) schemes allow certain mathematical operations to be performed directly on ciphertexts, without any prior decryption. Homomorphic encryption is a powerful tool for enabling Multiparty Computation (MPC) by enabling a participant to compute functions on values, keeping the values hidden. Different variations of HE exist, ranging from general Fully Homomorphic Encryption (FHE) [65] to the more efficient levelled variants [66, 67, 68, 69]. There are also partially homomorphic schemes allowing either homomorphic multiplication or addition.

Secure Aggregations: Secure aggregation is functionality for n number of clients and a server. It enables each client to submit a tensor value such that the server learns just an aggregate function of the clients' values, generally the sum. The server learns just an unordered collection of the messages from all clients. The server cannot link any message to its sender beyond the information in the message itself. There are many research literature exploring secure aggregation in both the single-server setting using threshold homomorphic encryption [70, 71, 72], pairwise additive masking [73, 74, 75], and generic secure multi-party computation [76]. It is also used in the multiple non-colluding servers setting [77, 78, 79]. Secure aggregation can also be implemented using trusted execution environments as in [80].

Secure Shuffling: Secure shuffling can be considered as an instance of Secure Aggregation where the values are multiset-singletons, and the aggregation operation is multiset-sum. It is mostly the case that very different implementations provide the best performance for secure shuffling and secure aggregation. Secure shufflers have been studied in the context of secure multi-party computation [81, 82] and also in trusted computing [83].

SecureBoost: SecureBoost is a novel gradient-tree boosting algorithm in the setting of Federated Learning (FL). It consists of two main steps. First, it aligns the data under the privacy constraint. Second, it collaboratively learns a shared gradient-tree boosting model while keeping all the training data secure over multiple private parties. SecureBoost is beneficial as it provides the same level of accuracy in comparison to non-privacy-preserving approach while at the same time, reveal zero information of each private data provider. The SecureBoost framework is as accurate as other non-federated gradient tree-boosting algorithms that bring the data into one place and is theoretically proven.

Private Information Retrieval (PIR): PIR is functionality for one client and one server. It enables the client to download an entry from a server-hosted database such that the server gains no information about which entry the client has requested. MPC approaches to PIR can be put into two main categories: computational PIR (cPIR), in which a single party can execute the entire server-side of the protocol [84], and information theoretic PIR (itPIR), in which multiple non-colluding parties are required to execute the server-side of the protocol [85]. Computational PIR has a very high computational cost [86], while the non-colluding parties setting has been complex to achieve in industrial scenarios. Recently, the results on PIR have shown dramatic reductions in the computational cost through the use of lattice-based cryptosystems [87, 88, 89].

It shows how to construct communication-efficient PIR on a single-server by leveraging side information available at the user [90]. Research works propose to leverage local client state to speed up PIR. Patel et al. [91] showed a practical hybrid PIR scheme on a single server was implemented and validated. Corrigan-Gibbs and Kogan [92] present protocols for PIR with sublinear online time by working in an offline/online model. During an offline phase, clients fetch information from the server(s) independent on the future query to be executed.

5. APPLICATIONS IN INDUSTRIES AND LIMITATIONS

Federated Learning (FL) is not only a technology standard but also a business model for many industries. When we consider the effects of big data, the first thing is to aggregate the data, compute the models through a remote processor, and then download the results for further use. In such cases, cloud computing comes into demand. With the increasing importance for data privacy and data security and a high relationship between a company's profits and its data, the cloud computing model has been challenged. However, the business model of Federated Learning (FL) has provided a new paradigm for many applications of big data. When the isolated data by each institution fails to produce an ideal model, the mechanism of Federated Learning (FL) makes it possible for many institutions and enterprises to share a united global model without the exchange of data.

However, Federated Learning (FL) could make equitable rules for profits allocation using blockchain techniques. We believe that the establishment of the business model for data alliance and the technical mechanism for Federated Learning (FL) should be implemented together. Various standards for Federated Learning (FL) in many fields need to be put into use for the betterment of industries and enterprises. Industries could use Federated Learning mechanism for the resilience management where the computing or manufacturing devices are likely to fail due to the quality fails or manufacturing defects at later stages. It could affect the profits of industries on a large scale, especially in the pandemic situations wherein there should be manual intervention for the devices for parameter settings. It could be an area of application for Federated mechanism where the devices run efficiently even in case of failures and thus optimizing the profits on a global scale. Cross-silo Federated Learning applications can be seen in various domains including finance risk prediction for reinsurance [93], electronic health records mining [94], pharmaceuticals discovery [95], medical data segmentation [96], and smart manufacturing [97]. Commercial data platforms incorporating Federated Learning (FL) are in progress in various technology companies and smaller start-up companies.

Even though there are significant practical privacy improvements of Federated Learning over centralizing all the training data, there is still no formal guarantee of privacy in this baseline Federated Learning (FL) model. The significant challenges of the Federated Learning (FL) setting are non-Independent and Identically Distributed (IID) data, unbalanced, massively distributed, and limited communication. Each user generates quite different data, and thus the data is non-IID data. Due to the massive number of participating clients in the federated process, some of the users produce significantly more data than others, making it unbalanced. It is massively distributed, and therefore there are more mobile device owners than the average training samples on each device. Due to unstable, unreliable and asymmetric mobile network connections between the clients and the server, there is limited communication.

6. DISCUSSION

Federated Learning (FL) embodies basic principles of focused data collection and minimization and can reduce many of the systemic privacy risks. Although there are many existing privacy-

preserving techniques in a Federated Learning (FL) setting, they still do not offer much support to complete privacy of the system. An actively malicious adversary controlling the central server could lead to a large number of fake client devices as a "Sybil attack" [98]. Adversarial attacks include data poisoning [99], model evasion attacks [99, 100] and model update poisoning [101, 102], which degrade the model performance. Hitaj et al. [103] devised an attack based on GANs, which shows that record-level differential privacy is generally ineffective in Federated Learning (FL) systems. Balcan et al. [104] introduced a concept to add statistical noise only to a subset of data, but the resulting privacy in this scenario is dependent on the number of statistical queries required to learn the dataset. It is interesting to investigate various approaches to facilitate federated privacy mechanisms, which could then be integrated into many business models for scaling the profits. Therefore, Federated Learning (FL) mechanism motivates the development of novel and highly trusted models taking Federated Learning's unique computational model as the baseline.

7. CONCLUSIONS AND FUTURE SCOPE

The emphasis on data privacy and security with the isolation of data has become the next challenges for AI, but Federated Learning (FL) has emerged with the solution. It could establish a united model for multiple enterprises and institutions while local data is protected so that enterprises could work together on data security. Thus, Federated Learning (FL) provides a platform to build a cross-enterprise, cross-data, and cross-domain ecosphere for AI, Machine learning and big data. This paper generally introduces the basic working of Federated Learning (FL), various architectures, privacy-preserving techniques of Federated Learning (FL), and discusses its potential in industrial applications. In the near future, Federated Learning (FL) would break the barriers between industries and establish a new community, wherein the data and knowledge could be shared. It ensures the safety and the benefits would be equally distributed based on the contribution of each participant. Finally, the essence and need of AI would be brought to every corner of our lives through Federated Learning (FL).

ACKNOWLEDGEMENTS

We thank DFKI SPAICER project for giving an opportunity and encouraging to do this survey work.

REFERENCES

[1] Albrecht, J. P. (2016), How the gdpr will change the world. Eur. Data Prot. L. Rev. 2:287.
[2] Regulation, P. (2016), The general data protection regulation. European Commission. [Online]. Available: https://eur-lex. europa. eu/legal-content/EN/TXT
[3] H. B. McMahan & Daniel Ramage. (2017), "Federated learning: Collaborative machine learning without centralized training data", Google AI Blog. [Online]. Available: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html
[4] McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017), "Communication-efficient learning of deep networks from decentralized data", *In Artificial Intelligence and Statistics*, ppl. 1273-1282.
[5] H. B. McMahan, Eider Moore, Daniel Ramage, Seth Hampson, & Blaise Aguera y Arcas. (2017), "Communication-efficient learning of deep networks from decentralized data", *In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ppl 1273–1282.
[6] Sundar Pichai. (2019), Privacy Should Not Be a Luxury Good, *New York Times*.
[7] Swaroop Ramaswamy, Rajiv Mathews, Kanishka Rao, & Françoise Beaufays. (2019), "Federated learning for emoji prediction in a mobile keyboard", [Online]. Available: http://arxiv.org/abs/1906.04329

[8] Andrew Hard, Kanishka Rao, Rajiv Mathews, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, & Daniel Ramage. (2018), "Federated learning for mobile keyboard prediction", [Online]. Available: http://arxiv.org/abs/1811.03604

[9] Mingqing Chen, Rajiv Mathews, Tom Ouyang, & Françoise Beaufays. (2019), "Federated learning of out-of-vocabulary words". [Online]. Available: http://arxiv.org/abs/1903.10635

[10] Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, & Françoise Beaufays. (2018), "Applied federated learning: Improving Google keyboard query suggestions", [Online]. Available: http://arxiv.org/abs/1812.02903

[11] support.google. (2019), Your chats stay private while Messages improves suggestions. [Online]. Available: https://support.google.com/messages/answer/9327902.

[12] ai.google. (2018), Under the hood of the Pixel 2: How AI is supercharging hardware. [Online]. Available: https://ai.google/stories/ai-in-hardware/

[13] Apple. (2019), Private Federated Learning, *NeurIPS 2019 Expo Talk Abstract.* [Online]. Available: ExpoConferences/2019/schedule?talk_id=40.

[14] Apple.(2019), Designing for privacy, *Apple WWDC.* [Online]. Available: https://developer.apple.com/videos/play/wwdc2019/708

[15] Walter de Brouwer. (2019), The federated future is ready for shipping. [Online]. Available: https://medium.com/@_doc_ai/the-federated-future-is-ready-for-shipping-d17ff40f43e3

[16] David Leroy, Alice Coucke, Thibaut Lavril, Thibault Gisselbrecht, & Joseph Dureau. (2018), "Federated learning for keyword spotting", [Online]. Available: http://arxiv.org/abs/1810.05512

[17] Jaideep Vaidya & Chris Clifton. (2003), "Privacy-preserving K-means clustering over vertically partitioned data", *In Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03). ACM*, ppl 206–215. [Online]. Available: https://doi.org/10.1145/956750.956776

[18] Jaideep Vaidya & Chris Clifton. (2002), "Privacy preserving association rule mining in vertically partitioned data". *In Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02). ACM*, ppl 639–644. [Online]. Available: https://doi.org/10.1145/775047.775142

[19] Jaideep Vaidya & Chris Clifton. (2004), "Privacy preserving naïve Bayes classifier for vertically partitioned data", *In Proceedings of the 4th SIAM Conference on Data Mining*, ppl 330–334.

[20] Murat Kantarcioglu & Chris Clifton. (2004), "Privacy-preserving distributed mining of association rules on horizontally partitioned data", *IEEE Trans. on Knowl. and Data Eng*, ppl 1026–1037. [Online]. Available: https://doi.org/10.1109/TKDE.2004.45

[21] Adrià Gascón, Phillipp Schoppmann, Borja Balle, Mariana Raykova, Jack Doerner, Samee Zahur, & David Evans. (2016), "Secure linear regression on vertically partitioned datasets", *IACR Cryptology*, 892.

[22] Irene Giacomelli, Somesh Jha, Marc Joye, C. David Page, & Kyonghwan Yoon. (2017), "Privacy-preserving ridge regression with only linearly-homomorphic encryption", *IACR Cryptology*, 979. [Online]. Available: https://eprint.iacr.org/2017/979

[23] Payman Mohassel & Yupeng Zhang. (2017), "SecureML: A system for scalable privacy-preserving machine learning". *IACR Cryptology*.

[24] Murat Kantarcioglu & Chris Clifton. (2004), "Privacy-preserving distributed mining of association rules on horizontally partitioned data", *IEEE Trans. on Knowl. and Data Eng*, ppl 1026–1037. [Online]. Available: https://doi.org/10.1109/TKDE.2004.45

[25] Hwanjo Yu, Xiaoqian Jiang, & Jaideep Vaidya. (2006), "Privacy-preserving SVM using nonlinear kernels on horizontally partitioned data", *In Proceedings of the 2006 ACM Symposium on Applied Computing (SAC'06),* ppl 603–610. [Online]. Available: https://doi.org/10.1145/1141277.1141415

[26] Hwanjo Yu, Jaideep Vaidya, & Xiaoqian Jiang. (2006), "Privacy-preserving SVM classification on vertically partitioned data", *In Proceedings of the 10th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD'06),* ppl 647–656. [Online]. Available: https://doi.org/10.1007/11731139_74

[27] Wenliang Du, Yunghsiang Sam Han, & Shigang Chen. (2004), "Privacy-preserving multivariate statistical analysis: Linear regression and classification", *In SDM*, Vol. 4, ppl 222–233.

[28] Li Wan, Wee Keong Ng, Shuguo Han, & Vincent C. S. Lee. (2007), "Privacy-preservation for gradient descent methods", *In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'07)*, ppl 775–783. [Online]. Available: https://doi.org/10.1145/1281192.1281275

[29] O. Goldreich, S. Micali, & A. Wigderson. (1987), "How to play any mental game", *In Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC'87)*, ppl 218–229. [Online]. Available: https://doi.org/10.1145/28395.28420

[30] Andrew C. Yao. (1982), "Protocols for secure computations", *In Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (SFCS'82),* IEEE Computer Society, ppl 160–164. [Online]. Available: http://dl.acm.org/citation.cfm?id=1382436.1382751

[31] Yoshinori Aono, Takuya Hayashi, Le Trieu Phong, & Lihua Wang. (2016), "Scalable and secure logistic regression via homomorphic encryption", *In Proceedings of the 6th ACM Conference on Data and Application Security and Privacy (CODASPY'16)*, ppl 142–144. [Online]. Available: https://doi.org/10.1145/2857705.2857731

[32] Reza Shokri & Vitaly Shmatikov. (2015). "Privacy-preserving deep learning", *In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS'15)* , ppl 1310–1321. [Online]. Available: https://doi.org/10.1145/2810103.2813687

[33] Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, & Shiho Moriai. (2018). Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Trans. Information Forensics and Security* (2018), ppl 1333–1345.

[34] Florian Bourse, Michele Minelli, Matthias Minihold, & Pascal Paillier. (2017), "Fast homomorphic evaluation of deep discretized neural networks", *IACR Cryptology*, 1114.

[35] Nathan Dowlin, Ran Gilad-Bachrach, Kim Laine, Kristin Lauter, Michael Naehrig, & John Wernsing. (2016), "CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy". [Online]. Available: https://www.microsoft.com/en-us/research/publication/cryptonets-applying-neural-networks-to-encrypted-data-with-high-throughput-and-accuracy/

[36] Ehsan Hesamifard, Hassan Takabi, & Mehdi Ghasemi. (2017), "CryptoDL: Deep neural networks over encrypted data", *CoRR*, [Online]. Available: http://arxiv.org/abs/1711.05189

[37] Bita Darvish Rouhani, M. Sadegh Riazi, & Farinaz Koushanfar. (2017), "DeepSecure: Scalable provably-secure deep learning", *CoRR*. [Online]. Available: http://arxiv.org/abs/1705.08963

[38] Joost Verbraeken, Matthijs Wolting, Jonathan Katzy, & Jeroen Klop. (2020), " A survey on Distributed Machine Learning", ACM Computing Surveys, Vol. 53, [Online]. Available: https://doi.org/10.1145/3377454

[39] Diego Peteiro-Barral & Bertha Guijarro-Berdiñas. (2013), "A survey of methods for distributed machine learning", *Progress in Artificial Intelligence*, Vol. 2, No. 1, ppl 1–11.

[40] Abhijit Guha Roy, S. Siddiqui, S. Pölsterl, Nassir Navab, & Christian Wachinger. (2019), "BrainTorrent: A Peer-to-Peer Environment for Decentralized Federated Learning," arXiv: 1905.06731

[41] Chenghao Hu, Jingyan Jiang, & Zhi Wang, (2019), "Decentralized Federated Learning: A Segmented Gossip Approach", *1st International Workshop on Federated Machine Learning for User Privacy and Data Confidentiality (FML'19)*.

[42] Róbert Ormándi, István Hegedűs, & Márk Jelasity. (2013), Gossip learning with linear models on fully distributed data. Concurrency and Computation: Practice and Experience, Vol. 25, No. 4, ppl 556–571.

[43] Peter Kairouz et. al., (2019), "Advances and Open Problems in Federated Learning", [Online]. Available: http://arxiv.org/abs/1912.04977

[44] Martín Abadi et. al., (2015), Large-scale machine learning on heterogeneous systems. [Online]. Available: https://www.tensorflow.org/

[45] Reza Shokri & Vitaly Shmatikov. (2015), "Privacy-preserving deep learning", *In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS'15),* ppl 1310–1321. [Online]. Available: https://doi.org/10.1145/2810103.2813687

[46] H. Brendan McMahan, Eider Moore, Daniel Ramage, & Blaise Agüera y Arcas. (2016), "Federated learning of deep networks using model averaging", CoRR, [Online]. Available: http://arxiv.org/abs/1602.05629

[47] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, & Karn Seth. (2017), "Practical secure aggregation for privacy-preserving machine learning", *In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS'17)*, ppl 1175–1191. [Online]. Available: https://doi.org/10.1145/3133956.3133982

[48] Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, & Shiho Moriai. (2018), "Privacy-preserving deep learning via additively homomorphic encryption", *IEEE Trans. Information Forensics and Security,* Vol. 13, No. 5, ppl 1333–1345.

[49] Reza Shokri & Vitaly Shmatikov. (2015), "Privacy-preserving deep learning", *In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS'15),* ppl 1310–1321. [Online]. Available: https://doi.org/10.1145/2810103.2813687

[50] Jaideep Vaidya & Chris Clifton. (2002), "Privacy preserving association rule mining in vertically partitioned data", *In Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02),* ppl 639–644, [Online]. Available: https://doi.org/10.1145/775047.775142

[51] W. Du & M. Atallah. (2001), "Privacy-preserving cooperative statistical analysis", *In Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC'01),* IEEE 102. [Online]. Available: http://dl.acm.org/citation.cfm?id=872016.872181

[52] Adrià Gascón, Phillipp Schoppmann, Borja Balle, Mariana Raykova, Jack Doerner, Samee Zahur, & David Evans. (2016), "Secure linear regression on vertically partitioned datasets", *IACR Cryptology*, 892.

[53] Ashish P. Sanil, Alan F. Karr, Xiaodong Lin, & Jerome P. Reiter. (2004), "Privacy preserving regression modelling via distributed computation", *In Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04)*, ppl 677–682. [Online]. Available: https://doi.org/10.1145/1014052.1014139

[54] Li Wan, Wee Keong Ng, Shuguo Han, & Vincent C. S. Lee. (2007), "Privacy-preservation for gradient descent methods", I*n Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'07),* ppl 775–783. [Online]. Available: https://doi.org/10.1145/1281192.1281275

[55] Wenliang Du, Yunghsiang Sam Han, & Shigang Chen. (2004), Privacy-preserving multivariate statistical analysis: Linear regression and classification. *In SDM,* Vol. 4. ppl 222–233.

[56] Gang Liang & Sudarshan S. Chawathe. (2004), "Privacy-preserving inter-database operations", *In International Conference on Intelligence and Security Informatics,* ppl 66–82.

[57] Amit P. Sheth & James A. Larson. (1990). "Federated database systems for managing distributed, heterogeneous, and autonomous databases", *ACM Comput. Surv.*, Vol. 22, No. 3, ppl 183–236. [Online]. Available: https://doi.org/10.1145/96602.96604

[58] Sinno Jialin Pan & Qiang Yang. (2010), "A survey on transfer learning", *IEEE Trans. Knowl. Data Eng.*, Vol. 22, No. 10, ppl 1345–1359. [Online]. Available: https://doi.org/10.1109/TKDE.2009.191

[59] The TFF Authors. (2019), TensorFlow Federated. [Online]. Available: https://www.tensorflow.org/federated

[60] Theo Ryffel, Andrew Trask, Morten Dahl, Bobby Wagner, Jason Mancuso, Daniel Rueckert, & Jonathan Passerat-Palmbach. (2018), A generic framework for privacy preserving deep learning.

[61] The FATE Authors. (2019) Federated AI technology enabler, [Online]. Available: https://www.fedai.org/

[62] The PaddleFL Authors. (2019) PaddleFL. [Online]. Available: https://github.com/PaddlePaddle/PaddleFL

[63] The Leaf Authors. (2019), Leaf. [Online]. Available: https://leaf.cmu.edu/

[64] The Clara Training Framework Authors. (2019), NVIDIA Clara. [Online]. Available: https://developer.nvidia.com/clara.

[65] Craig Gentry et al. (2009), Fully homomorphic encryption using ideal lattices. (2009), *In Stoc*, Vol. 9, ppl 169–178.

[66] Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp. (2012), *CRYPTO*, Vol. 7417, ppl 868–886.

[67] Junfeng Fan & Frederik Vercauteren. (2012), Somewhat practical fully homomorphic encryption, *IACR Cryptology*.

[68] Zvika Brakerski, Craig Gentry, & Vinod Vaikuntanathan. (2012), "(leveled) fully homomorphic encryption without bootstrapping", *In ITCS*, ppl 309–325.

[69] Jean-Sébastien Coron, Tancrède Lepoint, & Mehdi Tibouchi. (2014), "Scale-invariant fully homomorphic encryption over the integers", *In Public Key Cryptography*, Vol. 8383, ppl 311–328.

[70] Elaine Shi, HTH Chan, Eleanor Rieffel, Richard Chow, & Dawn Song , (2011), "Privacy-preserving aggregation of time-series data", *In Annual Network & Distributed System Security Symposium (NDSS).*

[71] Shai Halevi, Yehuda Lindell, & Benny Pinkas. (2011), "Secure computation on the web: Computing without simultaneous interaction", *In Annual Cryptology Conference*, ppl 132–150.

[72] T-H Hubert Chan, Elaine Shi, & Dawn Song. (2012), "Privacy-preserving stream aggregation with fault tolerance", *In International Conference on Financial Cryptography and Data Security*, ppl 200–214.

[73] Gergely Ács & Claude Castelluccia. (2011), "I have a DREAM!: DIfferentially PrivatE smart Metering", *In Proceedings of the 13th International Conference on Information Hiding, IH'11,* ppl 118–132. [Online]. Available: http://dl.acm.org/citation.cfm?id=2042445.2042457

[74] Slawomir Goryczka & Li Xiong. (2017), "A comprehensive comparison of multiparty secure additions with differential privacy", *IEEE Trans. Dependable Sec. Comput.*, Vol. 14, No. 5, ppl 463–477. [Online]. Available: https://doi.org/10.1109/TDSC.2015.2484326

[75] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, & Karn Seth, (2017), "Practical secure aggregation for privacy-preserving machine learning. *In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security"*, ppl 1175–1191.

[76] Martin Burkhart, Mario Strasser, Dilip Many, & Xenofontas Dimitropoulos. (2010), "SEPIA: Privacy-preserving aggregation of multi-domain network events and statistics", Network, Vol. 1.

[77] Peter Bogetoft, Dan Lund Christensen, Ivan Damgård, Martin Geisler, Thomas P. Jakobsen, Mikkel Krøigaard, Janus Dam Nielsen, Jesper Buus Nielsen, Kurt Nielsen, Jakob Pagter, Michael I. Schwartzbach, & Tomas Toft. (2009), "Secure multiparty computation goes live", *In Financial Cryptography*, Vol. 5628, ppl 325–343.

[78] Toshinori Araki, Jun Furukawa, Yehuda Lindell, Ariel Nof, & Kazuma Ohara. (2016), "High-throughput semi-honest secure three-party computation with an honest majority", *In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ppl 805–817.

[79] Henry Corrigan-Gibbs & Dan Boneh. (2017), "Prio: Private, robust, and scalable computation of aggregate statistics", *In 14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17)*, ppl 259–282.

[80] David Lie & Petros Maniatis. (2017), "Glimmers: Resolving the privacy/trust quagmire", *In Proceedings of the 16th Workshop on Hot Topics in Operating Systems*, ppl 94–99.

[81] Albert Kwon, David Lazar, Srinivas Devadas, & Bryan Ford. Riffle. (2016), *Proceedings on Privacy Enhancing Technologies*, Vol. 2, ppl 115–134.

[82] David Chaum. (1981), "Untraceable electronic mail, return addresses, and digital pseudonyms", *Communications of the ACM*, Vol. 24, No. 2.

[83] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnes, & Bernhard Seefeld. Prochlo (2017), "Strong privacy for analytics in the crowd", *In Proceedings of the 26th Symposium on Operating Systems Principles, SOSP '17,* ppl 441–459, [Online]. Available: http://doi.acm.org/10.1145/3132747.3132769

[84] Eyal Kushilevitz & Rafail Ostrovsky. (1997), "Replication is not needed: Single database, computationally-private information retrieval", *In Proc. of the 38th Annu. IEEE Symp. on Foundations of Computer Science*, ppl 364–373.

[85] Benny Chor, Eyal Kushilevitz, Oded Goldreich, & Madhu Sudan. (1998), "Private information retrieval", J. ACM, Vol .45, No. 6, ppl 965–981. [Online]. Available: http://doi.acm.org/10.1145/293347.293350

[86] Radu Sion & Bogdan Carbunar. (2007), "On the computational practicality of private information retrieval", *In Proceedings of the Network and Distributed Systems Security Symposium.*

[87] Femi Olumofin & Ian Goldberg. (2011), "Revisiting the computational practicality of private information retrieval", *In International Conference on Financial Cryptography and Data Security*, ppl 158–172.

[88] Sebastian Angel, Hao Chen, Kim Laine, & Srinath T. V. Setty. (2018), "PIR with compressed queries and amortized query processing", *In IEEE Symposium on Security and Privacy*, ppl 962–979.

[89] Craig Gentry & Shai Halevi. (2019), "Compressible FHE with applications to PIR", *In TCC*, Vol. 11892, ppl 438–464.

[90] S. Kadhe, B. Garcia, A. Heidarzadeh, S. E. Rouayheb, & A. Sprintson. (2017), "Private information retrieval with side information: The single server case", *In 55th Annual Allerton Conference on*

Communication, Control, and Computing (Allerton), ppl 1099–1106. [Online]. Available: 10.1109/ALLERTON.2017.8262860

[91] Sarvar Patel, Giuseppe Persiano, & Kevin Yeo. (2018), "Private stateful information retrieval", *In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS '18*, ppl 1002–1019. [Online]. Available: http://doi.acm.org/10.1145/3243734.3243821

[92] Henry Corrigan-Gibbs & Dmitry Kogan. (2019), "Private information retrieval with sublinear online time", *IACR Cryptology*.

[93] WeBank. (2019), WeBank & Swiss re signed cooperation MOU, [Online]. Available: https://finance.yahoo.com/news/webank-swiss-signed-cooperation-mou-112300218.html.

[94] FeatureCloud. (2019), FeatureCloud: Our vision. [Online]. Available: https://featurecloud.eu/

[95] EU CORDIS. (2019), "Machine learning ledger orchestration for drug discovery", [Online]. Available: https://cordis.europa.eu/project/rcn/223634/factsheet/en?WT.mc_id=RSS-Feed&WT.rss_f=project&WT.rss_a=223634&WT.rss_ev=a

[96] ai.intel. (2019), Federated learning for medical imaging. [Online]. Available: https://www.intel.ai/federated-learning-for-medical-imaging/

[97] Musketeer. (2019), Musketeer: About, 2019. [Online]. Available: http://musketeer.eu/project/.

[98] John R. Douceur. (2002), "The sybil attack", *In Revised Papers from the First International Workshop on Peer-to-Peer Systems, IPTPS '01*, ppl 251–260, [Online]. Available: http://dl.acm.org/citation.cfm?id=646334.687813

[99] Battista Biggio, Blaine Nelson, & Pavel Laskov. (2012), "Poisoning attacks against support vector machines", *In Proceedings of the 29th International Coference on International Conference on Machine Learning, ICML'12*, ppl 1467–1474. [Online]. Available: http://dl.acm.org/citation.cfm?id=3042573.3042761

[100] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, & RobFergus. (2013), "Intriguing properties of neural networks", *ICLR*.

[101] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, & Vitaly Shmatikov. (2018), "How to backdoor federated learning", [Online]. Available: http://arxiv.org/abs/1807.00459

[102] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, & Seraphin Calo. (2019), "Analyzing federated learning through an adversarial lens", *In Proceedings of the 36th International Conference on Machine Learning*, ppl 634–643.

[103] Peter Bogetoft, Dan Lund Christensen, Ivan Damgård, Martin Geisler, Thomas P. Jakobsen, Mikkel Kroigaard, Janus Dam Nielsen, Jesper Buus Nielsen, Kurt Nielsen, Jakob Pagter, Michael I. Schwartzbach, & Tomas Toft. (2009), "Secure multiparty computation goes live", *In Financial Cryptography*, Vol. 5628, ppl 325–343.

[104] Nitin Agrawal, Ali Shahin Shamsabadi, Matt J. Kusner, & Adrià Gascón. (2019), "QUOTIENT: two-party secure neural network training and prediction", *In Proceedings of the ACM Conference on Computer and Communication Security (CCS)*.

AUTHORS

Sheela Raju Kurupathi, Deep Learning Researcher, DFKI, Germany.