Federated Machine Learning: Concept and Applications

QIANG YANG, Hong Kong University of Science and Technology, Hong Kong
YANG LIU, Webank, China
TIANJIAN CHEN, Webank, China
YONGXIN TONG, Beihang University, China

Today's AI still faces two major challenges. One is that in most industries, data exists in the form of isolated islands. The other is the strengthening of data privacy and security. We propose a possible solution to these challenges: secure federated learning. Beyond the federated learning framework first proposed by Google in 2016, we introduce a comprehensive secure federated learning framework, which includes horizontal federated learning, vertical federated learning and federated transfer learning. We provide definitions, architectures and applications for the federated learning framework, and provide a comprehensive survey of existing works on this subject. In addition, we propose building data networks among organizations based on federated mechanisms as an effective solution to allow knowledge to be shared without compromising user privacy.

CCS Concepts: • **Security and privacy**; • **Computing methodologies** → **Artificial intelligence**; **Machine learning**; *Supervised learning*;

Additional Key Words and Phrases: federated learning, GDPR, transfer learning

ACM Reference Format:
Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated Machine Learning: Concept and Applications. *ACM Trans. Intell. Syst. Technol.* 10, 2, Article 12 (February 2019), 19 pages. https://doi.org/0000001.0000001

1 INTRODUCTION

2016 is the year when artificial intelligence (AI) came of age. With AlphaGo[59] defeating the top human Go players, we have truly witnessed the huge potential in artificial intelligence (AI), and have began to expect more complex, cutting-edge AI technology in many applications, including driverless cars, medical care, finance, etc. Today, AI technology is showing its strengths in almost every industry and walks of life. However, when we look back at the development of AI, it is inevitable that the development of AI has experienced several ups and downs. Will there be a next down turn for AI? When will it appear and because of what factors? The current public interest in AI is partly driven by Big Data availability: AlphaGo in 2016 used a total of 300,000 games as training data to achieve the excellent results.

With AlphaGo's success, people naturally hope that the big data-driven AI like AlphaGo will be realized soon in all aspects of our lives. However, the real world situations are somewhat disappointing: with the exception of few industries, most fields have only limited data or poor

Authors' addresses: Qiang Yang, Hong Kong University of Science and Technology, Hong Kong, China; email: qyang@cse.ust.hk; Yang Liu, Webank, Shenzhen, China; email: yangliu@webank.com; Tianjian Chen, Webank, Shenzhen, China; email: tobychen@webank.com; Yongxin Tong (corresponding author), Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing, China; email: yxtong@buaa.edu.cn.

2157-6904/2019/2-ART12 $15.00
https://doi.org/0000001.0000001

quality data, making the realization of AI technology more difficult than we thought. Would it be possible to fuse the data together in a common site, by transporting the data across organizations? In fact, it is very difficult, if not impossible, in many situations to break the barriers between data sources. In general, the data required in any AI project involves multiple types. For example, in an AI-driven product recommendation service, the product seller has information about the product, data of the user's purchase, but not the data that describe user's purchasing ability and payment habits. In most industries, data exists in the form of isolated islands. Due to industry competition, privacy security, and complicated administrative procedures, even data integration between different departments of the same company faces heavy resistance. It is almost impossible to integrate the data scattered around the country and institutions, or the cost is prohibited.

At the same time, with the increasing awareness of large companies compromising on data security and user privacy, the emphasis on data privacy and security has become a worldwide major issue. News about leaks on public data are causing great concerns in public media and governments. For example, the recent data breach by Facebook has caused a wide range of protests [70]. In response, states across the world are strengthening laws in protection of data security and privacy. An example is the General Data Protection Regulation (GDPR)[19] enforced by the European Union on May 25, 2018. GDPR (Figure 1) aims to protect users' personal privacy and data security. It requires businesses to use clear and plain languages for their user agreement and grants users the "right to be forgotten", that is, users can have their personal data deleted or withdrawn. Companies violating the bill will face stiff fine. Similar acts of privacy and security are being enacted in the US and China. For example, China's Cyber Security Law and the General Principles of the Civil Law, enacted in 2017, require that Internet businesses must not leak or tamper with the personal information that they collect and that, when conducting data transactions with third parties, they need to ensure that the proposed contract follow legal data protection obligations. The establishment of these regulations will clearly help build a more civil society, but will also pose new challenges to the data transaction procedures commonly used today in AI.

To be more specific, traditional data processing models in AI often involves simple data transactions models, with one party collecting and transferring data to another party, and this other party will be responsible for cleaning and fusing the data. Finally a third party will take the integrated data and build models for still other parties to use. The models are usually the final products that are sold as a service. This traditional procedure face challenges with the above new data regulations and laws. As well, since users may be unclear about the future uses of the models, the transactions violate laws such as the GDPR. As a result, we face a dilemma that our data is in the form of isolated islands, but we are forbidden in many situations to collect, fuse and use the data to different places for AI processing. How to legally solve the problem of data fragmentation and isolation is a major challenge for AI researchers and practitioners today.

In this article, we give an overview of a new approach known as *federated learning*, which is a possible solution for these challenges. We survey existing works on federated learning, and propose definitions, categorizations and applications for a comprehensive secure federated learning framework. We discuss how the federated learning framework can be applied to various businesses successfully. In promoting federated learning, we hope to shift the focus of AI development from improving model performance, which is what most of the AI field is currently doing, to investigating methods for data integration that is compliant with data privacy and security laws.

2 AN OVERVIEW OF FEDERATED LEARNING

The concept of federated learning is proposed by Google recently [36, 37, 41]. Their main idea is to build machine learning models based on data sets that are distributed across multiple devices while preventing data leakage. Recent improvements have been focusing on overcoming the

Fig. 1. GDPR: EU regulation on data protection

statistical challenges [60, 77] and improving security [9, 23] in federated learning. There are also research efforts to make federated learning more personalizable [13, 60]. The above works all focus on on-device federated learning where distributed mobile user interactions are involved and communication cost in massive distribution, unbalanced data distribution and device reliability are some of the major factors for optimization. In addition, data are partitioned by user Ids or device Ids, therefore, *horizontally* in the data space. This line of work is very related to privacy-preserving machine learning such as [58] because it also considers data privacy in a decentralized collaborative learning setting. To extend the concept of federated learning to cover collaborative learning scenarios among organizations, we extend the original "federated learning" to a general concept for all privacy-preserving decentralized collaborative machine learning techniques. In [71], we have given a preliminary overview of the federated learning and federated transfer learning technique. In this article, we further survey the relevant security foundations and explore the relationship with several other related areas, such as multiagent theory and privacy-preserving data mining. In this section, we provide a more comprehensive definition of federated learning which considers data partitions, security and applications. We also describe a workflow and system architecture for the federated learning system.

2.1 Definition of Federated Learning

Define N data owners $\{\mathcal{F}_1, ...\mathcal{F}_N\}$, all of whom wish to train a machine learning model by consolidating their respective data $\{\mathcal{D}_1, ...\mathcal{D}_N\}$. A conventional method is to put all data together and use $\mathcal{D} = \mathcal{D}_1 \cup ... \cup \mathcal{D}_N$ to train a model $\mathcal{M}_{SUM}$. A federated learning system is a learning process in which the data owners collaboratively train a model $\mathcal{M}_{FED}$, in which process any data owner $\mathcal{F}_i$ does not expose its data $\mathcal{D}_i$ to others [1]. In addition, the accuracy of $\mathcal{M}_{FED}$, denoted as $\mathcal{V}_{FED}$ should be very close to the performance of $\mathcal{M}_{SUM}$, $\mathcal{V}_{SUM}$. Formally, let δ be a non-negative real number, if

$$| \mathcal{V}_{FED} - \mathcal{V}_{SUM} | < \delta \tag{1}$$

we say the federated learning algorithm has δ-accuracy loss.

2.2 Privacy of Federated Learning

Privacy is one of the essential properties of federated learning. This requires security models and analysis to provide meaningful privacy guarantees. In this section, we briefly review and compare different privacy techniques for federated learning, and identify approaches and potential challenges for preventing indirect leakage.

[1]Definition of data security may differ in different scenarios, but is required to provide meaning privacy guarantees. We demonstrate examples of security definitions in section 2.3

Secure Multi-party Computation (SMC). SMC security models naturally involve multiple parties, and provide security proof in a well-defined simulation framework to guarantee complete zero knowledge, that is, each party knows nothing except its input and output. Zero knowledge is very desirable, but this desired property usually requires complicated computation protocols and may not be achieved efficiently. In certain scenarios, partial knowledge disclosure may be considered acceptable if security guarantees are provided. It is possible to build a security model with SMC under lower security requirement in exchange for efficiency [16]. Recently, studies [46] used SMC framework for training machine learning models with two servers and semi-honest assumptions. Ref [33] uses MPC protocols for model training and verification without users revealing sensitive data. One of the state-of-the-art SMC framework is Sharemind [8]. Ref [44] proposed a 3PC model [5, 21, 45] with an honest majority and consider security in both semi-honest and malicious assumptions. These works require participants' data to be secretly-shared among non-colluding servers.

Differential Privacy. Another line of work use techniques Differential Privacy [18] or k-Anonymity [63] for data privacy protection [1, 12, 42, 61]. The methods of differential privacy, k-anonymity, and diversification [3] involve in adding noise to the data, or using generalization methods to obscure certain sensitive attributes until the third party cannot distinguish the individual, thereby making the data impossible to be restore to protect user privacy. However, the root of these methods still require that the data are transmitted elsewhere and these work usually involve a trade-off between accuracy and privacy. In [23], authors introduced a differential privacy approach to federated learning in order to add protection to client-side data by hiding client's contributions during training.

Homomorphic Encryption. Homomorphic Encryption [53] is also adopted to protect user data privacy through parameter exchange under the encryption mechanism during machine learning [24, 26, 48]. Unlike differential privacy protection, the data and the model itself are not transmitted, nor can they be guessed by the other party's data. Therefore, there is little possibility of leakage at the raw data level. Recent works adopted homomorphic encryption for centralizing and training data on cloud [75, 76]. In practice, Additively Homomorphic Encryption [2] are widely used and polynomial approximations need to be made to evaluate non-linear functions in machine learn algorithms, resulting in the trade-offs between accuracy and privacy [4, 35].

2.2.1 Indirect information leakage. Pioneer works of federated learning exposes intermediate results such as parameter updates from an optimization algorithm like Stochastic Gradient Descent (SGD) [41, 58], however no security guarantee is provided and the leakage of these gradients may actually leak important data information [51] when exposed together with data structure such as in the case of image pixels. Researchers have considered the situation when one of the members of a federated learning system maliciously attacks others by allowing a backdoor to be inserted to learn others' data. In [6], the authors demonstrate that it is possible to insert hidden backdoors into a joint global model and propose a new "constrain-and-scale" model-poisoning methodology to reduce the data poisoning. In [43], researchers identified potential loopholes in collaborative machine learning systems, where the training data used by different parties in collaborative learning is vulnerable to inference attacks. They showed that an adversarial participant can infer membership as well as properties associated with a subset of the training data. They also discussed possible defenses against these attacks.In [62], authors expose a potential security issue associated with gradient exchanges between different parties, and propose a secured variant of the gradient descent method and show that it tolerates up to a constant fraction of Byzantine workers.

Researchers have also started to consider blockchain as a platform for facilitating federated learning. In [34], researchers have considered a block-chained federated learning (BlockFL) architecture, where mobile devices' local learning model updates are exchanged and verified by leveraging blockchain. They have considered an optimal block generation, network scalability and robustness issues.

2.3 A Categorization of Federated Learning

In this section we discuss how to categorize federated learning based on the distribution characteristics of the data.

Let matrix $\mathcal{D}_i$ denotes the data held by each data owner i. Each row of the matrix represents a sample, and each column represents a feature. At the same time, some data sets may also contain label data. We denote the features space as $\mathcal{X}$, the label space as $\mathcal{Y}$ and we use $\mathcal{I}$ to denote the sample ID space. For example, in the financial field labels may be users' credit; in the marketing field labels may be the user's purchase desire; in the education field, $\mathcal{Y}$ may be the degree of the students. The feature $\mathcal{X}$, label $\mathcal{Y}$ and sample Ids $\mathcal{I}$ constitutes the complete training dataset $(\mathcal{I}, \mathcal{X}, \mathcal{Y})$. The feature and sample space of the data parties may not be identical, and we classify federated learning into horizontally federated learning, vertically federated learning and federated transfer learning based on how data is distributed among various parties in the feature and sample ID space. Figure 2 shows the various federated learning frameworks for a two-party scenario .

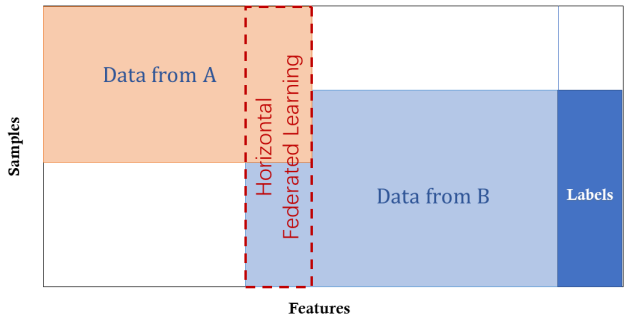
2.3.1 Horizontal Federated Learning. Horizontal federated learning, or sample-based federated learning, is introduced in the scenarios that data sets share the same feature space but different in samples (Figure 2a). For example, two regional banks may have very different user groups from their respective regions, and the intersection set of their users is very small. However, their business is very similar, so the feature spaces are the same. Ref [58] proposed a collaboratively deep learning scheme where participants train independently and share only subsets of updates of parameters. In 2017, Google proposed a horizontal federated learning solution for Android phone model updates [41]. In that framework, a single user using an Android phone updates the model parameters locally and uploads the parameters to the Android cloud, thus jointly training the centralized model together with other data owners. A secure aggregation scheme to protect the privacy of aggregated user updates under their federated learning framework is also introduced [9]. Ref [51] uses additively homomorphic encryption for model paramter aggregation to provide security against the central server.

In [60], a multi-task style federated learning system is proposed to allow multiple sites to complete separate tasks, while sharing knowledge and preserving security. Their proposed multi-task learning model can in addition address high communication costs, stragglers, and fault tolerance issues. In [41], the authors proposed to build a secure client-server structure where the federated learning system partitions data by users, and allow models built at client devices to collaborate at the server site to build a global federated model. The process of model building ensures that there is no data leakage. Likewise, in [36], the authors proposed methods to improve the communication cost to facilitate the training of centralized models based on data distributed over mobile clients. Recently, a compression approach called Deep Gradient Compression [39] is proposed to greatly reduce the communication bandwidth in large-scale distributed training.

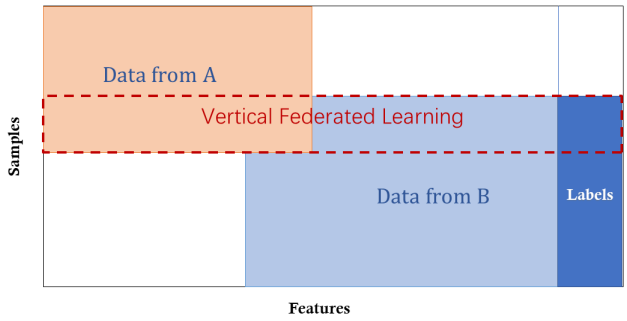
We summarize horizontal federated learning as:

$$\mathcal{X}_i = \mathcal{X}_j, \quad \mathcal{Y}_i = \mathcal{Y}_j, \quad I_i \neq I_j, \quad \forall \mathcal{D}_i, \mathcal{D}_j, i \neq j \tag{2}$$

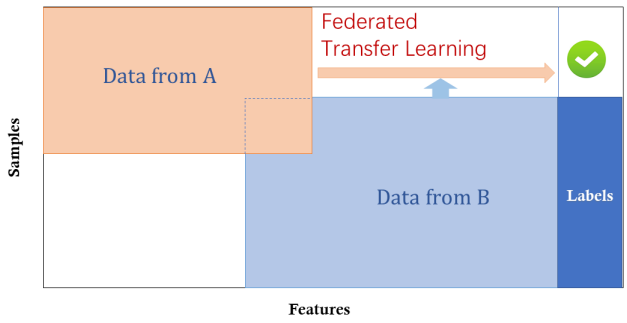
Security Definition. A horizontal federated learning system typically assumes honest participants and security against a honest-but-curious server [9, 51]. That is, only the server can compromise



(a) Horizontal Federated Learning



(b) Vertical Federated Learning



(c) Federated Transfer Learning

Fig. 2. Categorization of Federated Learning

the privacy of data participants. Security proof has been provided in these works. Recently another security model considering malicious user [29] is also proposed, posing additional privacy challenges. At the end of the training, the universal model and the entire model parameters are exposed to all participants.

2.3.2 Vertical Federated Learning. Privacy-preserving machine learning algorithms have been proposed for vertically partitioned data, including Cooperative Statistical Analysis [15], association rule mining [65], secure linear regression [22, 32, 55], classification [16] and gradient descent [68]. Recently, Ref [27, 49] proposed a vertical federated learning scheme to train a privacy-preserving logistic regression model. The authors studied the effect of entity resolution on the learning performance and applied Taylor approximation to the loss and gradient functions so that homomorphic encryption can be adopted for privacy-preserving computations.

Vertical federated learning or feature-based federated learning (Figure 2b) is applicable to the cases that two data sets share the same sample ID space but differ in feature space. For example, consider two different companies in the same city, one is a bank, and the other is an e-commerce company. Their user sets are likely to contain most of the residents of the area, so the intersection of their user space is large. However, since the bank records the user's revenue and expenditure behavior and credit rating, and the e-commerce retains the user's browsing and purchasing history, their feature spaces are very different. Suppose that we want both parties to have a prediction model for product purchase based on user and product information.

Vertically federated learning is the process of aggregating these different features and computing the training loss and gradients in a privacy-preserving manner to build a model with data from both parties collaboratively. Under such a federal mechanism, the identity and the status of each participating party is the same, and the federal system helps everyone establish a "common wealth" strategy, which is why this system is called "federated learning.". Therefore, in such a system, we have:

$$\mathcal{X}_i \neq \mathcal{X}_j, \quad \mathcal{Y}_i \neq \mathcal{Y}_j, \quad I_i = I_j \quad \forall \mathcal{D}_i, \mathcal{D}_j, i \neq j \tag{3}$$

Security Definition. A vertical federated learning system typically assumes honest-but-curious participants. In a two-party case, for example, the two parties are non-colluding and at most one of them are compromised by an adversary. The security definition is that the adversary can only learn data from the client that it corrupted but not data from the other client beyond what is revealed by the input and output. To facilitate the secure computations between the two parties, sometimes a Semi-honest Third Party (STP) is introduced, in which case it is assumed that STP does not collude with either party. SMC provides formal privacy proof for these protocols [25]. At the end of learning, each party only holds the model parameters associated to its own features, therefore at inference time, the two parties also need to collaborate to generate output.

2.3.3 Federated Transfer Learning (FTL). Federated Transfer Learning applies to the scenarios that the two data sets differ not only in samples but also in feature space. Consider two institutions, one is a bank located in China, and the other is an e-commerce company located in the United States. Due to geographical restrictions, the user groups of the two institutions have a small intersection. On the other hand, due to the different businesses, only a small portion of the feature space from both parties overlaps. In this case, transfer learning [50] techniques can be applied to provide solutions for the entire sample and feature space under a federation (Figure2c). Specially, a common representation between the two feature space is learned using the limited common sample sets and later applied to obtain predictions for samples with only one-side features. FTL is an important extension to the existing federated learning systems because it deals with problems exceeding the

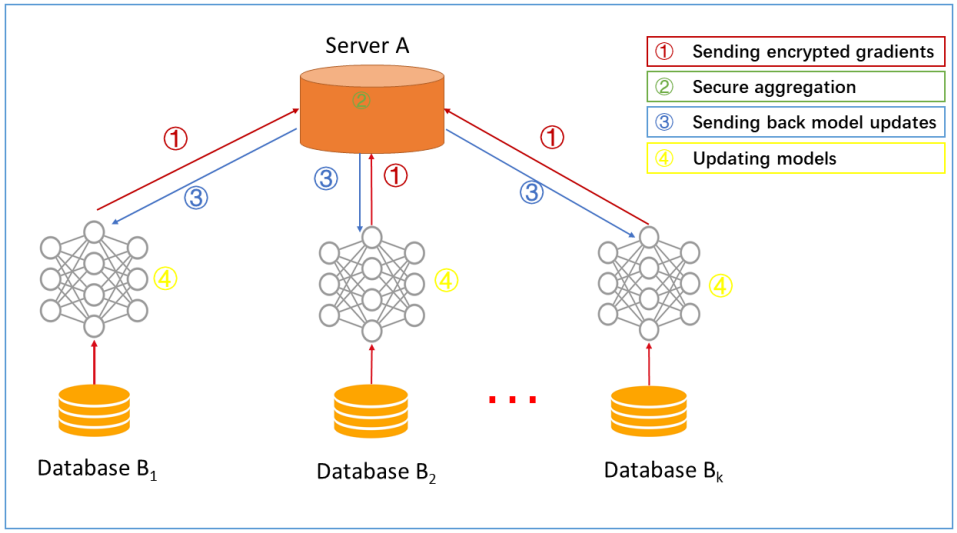


Fig. 3. Architecture for a horizontal federated learning system

scope of existing federated learning algorithms:

$$\mathcal{X}_i \neq \mathcal{X}_j, \quad \mathcal{Y}_i \neq \mathcal{Y}_j, \quad I_i \neq I_j \quad \forall \mathcal{D}_i, \mathcal{D}_j, i \neq j \tag{4}$$

Security Definition. A federated transfer learning system typically involves two parties. As will be shown in the next section, its protocols are similar to the ones in vertical federated learning, in which case the security definition for vertical federated learning can be extended here.

2.4 Architecture for a federated learning system

In this section, we illustrate examples of general architectures for a federated learning system. Note that the architectures of horizontal and vertical federated learning systems are quite different by design, and we will introduce them separately.

2.4.1 Horizontal Federated Learning. A typical architecture for a horizontal federated learning system is shown in Figure 3. In this system, k participants with the same data structure collaboratively learn a machine learning model with the help of a parameter or cloud server. A typical assumption is that the participants are honest whereas the server is honest-but-curious, therefore no leakage of information from any participants to the server is allowed [51]. The training process of such a system usually contain the following four steps:

- **Step 1**: participants locally compute training gradients, mask a selection of gradients with encryption [51], differential privacy [58] or secret sharing [9] techniques, and send masked results to server;
- **Step 2**: Server performs secure aggregation without learning information about any participant;
- **Step 3**: Server send back the aggregated results to participants;
- **Step 4**: Participants update their respective model with the decrypted gradients.

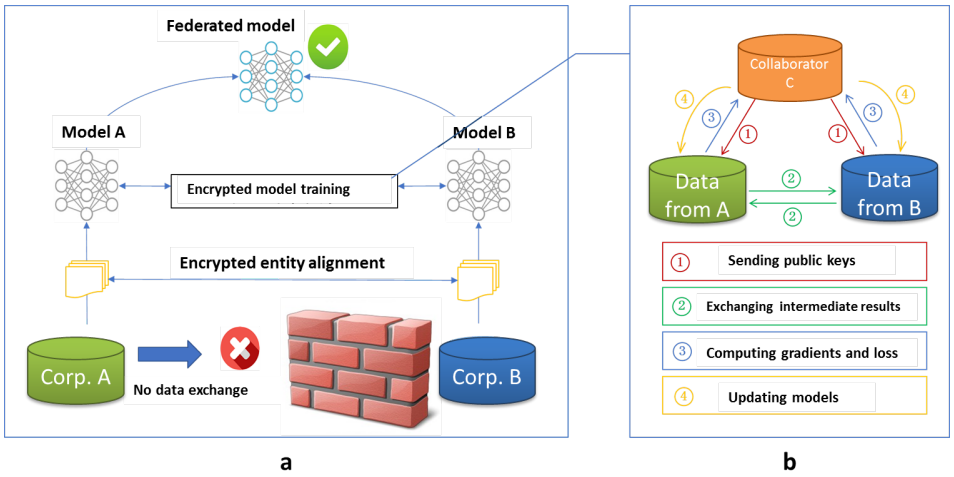


Fig. 4. Architecture for a vertical federated learning system

Iterations through the above steps continue until the loss function converges, thus completing the entire training process. This architecture is independent of specific machine learning algorithms (logistic regression, DNN etc) and all participants will share the final model parameters.

Security Analysis. The above architecture is proved to protect data leakage against the semi-honest server, if gradients aggregation is done with SMC [9] or Homomorphic Encryption [51]. But it may be subject to attack in another security model by a malicious participant training a Generative Adversarial Network (GAN) in the collaborative learning process [29].

2.4.2 Vertical Federated Learning. Suppose that companies A and B would like to jointly train a machine learning model, and their business systems each have their own data. In addition, Company B also has label data that the model needs to predict. For data privacy and security reasons, A and B cannot directly exchange data. In order to ensure the confidentiality of the data during the training process, a third-party collaborator C is involved. Here we assume the collaborator C is honest and does not collude with A or B, but party A and B are honest-but-curious to each other. A trusted third party C a reasonable assumption since party C can be played by authorities such as governments or replaced by secure computing node such as Intel Software Guard Extensions (SGX) [7]. The federated learning system consists of two parts, as shown in Figure 4.

Part 1. Encrypted entity alignment. Since the user groups of the two companies are not the same, the system uses the encryption-based user ID alignment techniques such as [38, 56] to confirm the common users of both parties without A and B exposing their respective data. During the entity alignment, the system does not expose users that do not overlap with each other.

Part 2. Encrypted model training. After determining the common entities, we can use these common entities' data to train the machine learning model. The training process can be divided into the following four steps (as shown in Figure 4):

- **Step 1**: collaborator C creates encryption pairs, send public key to A and B;
- **Step 2**: A and B encrypt and exchange the intermediate results for gradient and loss calculations;

- **Step 3**: A and B computes encrypted gradients and adds *additional mask*, respectively,and B also computes encrypted loss; A and B send encrypted values to C;
- **Step 4**: C decrypts and send the decrypted gradients and loss back to A and B; A and B unmask the gradients, update the model parameters accordingly.

Here we illustrate the training process using linear regression and homomorphic encryption as an example. To train a linear regression model with gradient descent methods, we need secure computations of its loss and gradients. Assuming learning rate η, regularization parameter λ, data set $\{x_i^A\}_{i\in\mathcal{D}_A}$, $\{x_i^B, y_i\}_{i\in\mathcal{D}_B}$, and model paramters Θ_A, Θ_B corresponding to the feature space of x_i^A, x_i^B respectively, the training objective is:

$$\min_{\Theta_A,\Theta_B} \sum_i ||\Theta_A x_i^A + \Theta_B x_i^B - y_i||^2 + \frac{\lambda}{2}(||\Theta_A||^2 + ||\Theta_B||^2) \tag{5}$$

let $u_i^A = \Theta_A x_i^A$, $u_i^B = \Theta_B x_i^B$, the encrypted loss is:

$$[[\mathcal{L}]] = [[\sum_i ((u_i^A + u_i^B - y_i))^2 + \frac{\lambda}{2}(||\Theta_A||^2 + ||\Theta_B||^2)]] \tag{6}$$

where additive homomorphic encryption is denoted as $[[\cdot]]$. Let $[[\mathcal{L}_A]] = [[\sum_i ((u_i^A)^2) + \frac{\lambda}{2}\Theta_A^2]]$, $[[\mathcal{L}_B]] = [[\sum_i ((u_i^B - y_i)^2) + \frac{\lambda}{2}\Theta_B^2]]$, and $[[\mathcal{L}_{AB}]] = 2\sum_i ([[u_i^A]](u_i^B - y_i))$, then

$$[[\mathcal{L}]] = [[\mathcal{L}_A]] + [[\mathcal{L}_B]] + [[\mathcal{L}_{AB}]] \tag{7}$$

Similarly, let $[[d_i]] = [[u_i^A]] + [[u_i^B - y_i]]$, then gradients are:

$$[[\frac{\partial \mathcal{L}}{\partial \Theta_A}]] = \sum_i [[d_i]] x_i^A + [[\lambda \Theta_A]] \tag{8}$$

$$[[\frac{\partial \mathcal{L}}{\partial \Theta_B}]] = \sum_i [[d_i]] x_i^B + [[\lambda \Theta_B]] \tag{9}$$

Table 1. Training Steps for Vertical Federated Learning : Linear Regression

	party A	party B	party C
step 1	initialize Θ_A	initialize Θ_B	create an encryption key pair, send public key to A and B;
step 2	compute $[[u_i^A]]$,$[[\mathcal{L}_A]]$ and send to B;	compute $[[u_i^B]]$,$[[d_i^B]]$,$[[\mathcal{L}]]$, send $[[d_i^B]]$ to A, send $[[\mathcal{L}]]$ to C;	
step 3	initialize R_A, compute $[[\frac{\partial \mathcal{L}}{\partial \Theta_A}]]$+$[[R_A]]$ and send to C;	initialize R_B, compute $[[\frac{\partial \mathcal{L}}{\partial \Theta_B}]]$+$[[R_B]]$ and send to C;	C decrypt $\mathcal{L}$, send $\frac{\partial \mathcal{L}}{\partial \Theta_A} + R_A$ to A, $\frac{\partial \mathcal{L}}{\partial \Theta_B} + R_B$ to B;
step 4	update Θ_A	update Θ_B	
what is obtained	Θ_A	Θ_B	

Table 2. Evaluation Steps for Vertical Federated Learning : Linear Regression

	party A	party B	inquisitor C
step 0			send user ID i to A and B;
step 1	compute u_i^A and send to C	compute u_i^B and send to C;	get result $u_i^A + u_i^B$;

See Table 1 and 2 for the detailed steps. During entity alignment and model training, the data of A and B are kept locally, and the data interaction in training does not lead to data privacy leakage. Note potential information leakage to C may or may not be considered to be privacy violation. To further prevent C to learn information from A or B in this case, A and B can further hide their gradients from C by adding encrypted random masks. Therefore, the two parties achieve training a common model cooperatively with the help of federated learning. Because during the training, the loss and gradients each party receives are exactly the same as the loss and gradients they would receive if jointly building a model with data gathered at one place without privacy constraints, that is, this model is lossless. The efficiency of the model depends on the communication cost and computation cost of encrypted data. In each iteration the information sent between A and B scales with the number of overlapping samples. Therefore the efficiency of this algorithm can be further improved by adopting distributed parallel computing techniques.

Security Analysis. The training protocol shown in Table 1 does not reveal any information to C, because all C learns are the masked gradients and the randomness and secrecy of the masked matrix are guaranteed [16]. In the above protocol, party A learns its gradient at each step, but this is not enough for A to learn any information from B according to equation 8, because the security of scalar product protocol is well-established based on the inability of solving n equations in more than n unknowns [16, 65]. Here we assume the number of samples N_A is much greater than n_A, where n_A is the number of features. Similarly, party B can not learn any information from A. Therefore the security of the protocol is proved. Note we have assumed that both parties are semi-honest. If a party is malicious and cheats the system by faking its input, for example, party A submits only one non-zero input with only one non-zero feature, it can tell the value of u_i^B for that feature of that sample. It still can not tell x_i^B or Θ_B though, and the deviation will distort results for the next iteration, alarming the other party who will terminate the learning process. At the end of the training process, each party (A or B) remains oblivious to the data structure of the other party, and it obtains the model parameters associated only with its own features. At inference time, the two parties need to collaboratively compute the prediction results, with the steps shown in Table 2, which still do not lead to information leakage.

2.4.3 Federated Transfer Learning. Suppose in the above vertical federated learning example, party A and B only have a very small set of overlapping samples and we are interested in learning the labels for all the data set in party A. The architecture described in the above section so far only works for the overlapping data set. To extend its coverage to the entire sample space, we introduce transfer learning. This does not change the overall architecture shown in Figure 4 but the details of the intermediate results that are exchanged between party A and party B. Specifically, transfer learning typically involves in learning a common representation between the features of party A and B, and minimizing the errors in predicting the labels for the target-domain party by leveraging the labels in the source-domain party (B in this case). Therefore the gradient computations for

party A and party B are different from that in the vertical federated learning scenario. At inference time, it still requires both parties to compute the prediction results.

2.4.4 Incentives Mechanism. In order to fully commercialize federated learning among different organizations, a fair platform and incentive mechanisms needs to be developed [20]. After the model is built, the performance of the model will be manifested in the actual applications and this performance can be recorded in a permanent data recording mechanism (such as Blockchain). Organizations that provide more data will be better off, and the model's effectiveness depends on the data provider's contribution to the system. The effectiveness of these models are distributed to parties based on federated mechanisms and continue to motivate more organizations to join the data federation.

The implementation of the above architecture not only considers the privacy protection and effectiveness of collaboratively-modeling among multiple organizations, but also considers how to reward organizations that contribute more data, and how to implement incentives with a consensus mechanism. Therefore, federated learning is a "closed-loop" learning mechanism.

3 RELATED WORKS

Federated learning enables multiple parties to collaboratively construct a machine learning model while keeping their private training data private. As a novel technology, federated learning has several threads of originality, some of which are rooted on existing fields. Below we explain the relationship between federated learning and other related concepts from multiple perspectives.

3.1 Privacy-preserving machine learning

Federated learning can be considered as privacy-preserving decentralized collaborative machine learning, therefore it is tightly related to multi-party privacy-preserving machine learning. Many research efforts have been devoted to this area in the past. For example, Ref [17, 67] proposed algorithms for secure multi-party decision tree for vertically partitioned data. Vaidya and Clifton proposed secure association mining rules [65], secure k-means [66], Naive Bayes classifier [64] for vertically partitioned data. Ref [31] proposed an algorithm for association rules on horizontally partitioned data. Secure Support Vector Machines algorithms are developed for vertically partitioned data [73] and horizontally partitioned data [74]. Ref [16] proposed secure protocols for multi-party linear regression and classification. Ref [68] proposed secure multi-party gradient descent methods. The above works all used secure multi-party computation (SMC) [25, 72] for privacy guarantees.

Nikolaenko et al.[48] implemented a privacy-preserving protocol for linear regression on horizontally partitioned data using homomorphic encryption and Yao's garbled circuits and Ref [22, 24] proposed a linear regression approach for vertically partitioned data. These systems solved the linear regression problem directly. Ref [47] approached the problem with Stochastic Gradient Descent (SGD) and they also proposed privacy-preserving protocols for logistic regression and neural networks. Recently, a follow-up work with a three-server model is proposed [44]. Aono et al.[4] proposed a secure logistic regression protocol using homomorphic encryption. Shokri and Shmatikov [58] proposed training of neural networks for horizontally partitioned data with exchanges of updated parameters. Ref [51] used the additively homomorphic encryption to preserve the privacy of gradients and enhance the security of the system. With the recent advances in deep learning, privacy-preserving neural networks inference is also receiving a lot of research interests[10, 11, 14, 28, 40, 52, 54].

3.2 Federated Learning vs Distributed Machine Learning

Horizontal federated learning at first sight is somewhat similar to Distributed Machine Learning. Distributed machine learning covers many aspects, including distributed storage of training data, distributed operation of computing tasks, distributed distribution of model results, etc. Parameter Server [30] is a typical element in distributed machine learning. As a tool to accelerate the training process, the parameter server stores data on distributed working nodes, allocates data and computing resources through a central scheduling node, so as to train the model more efficiently. For horizontally federated learning, the working node represents the data owner. It has full autonomy for the local data, and can decide when and how to join the federated learning. In the parameter server, the central node always takes the control, so federated learning is faced with a more complex learning environment. Secondly, federated learning emphasizes the data privacy protection of the data owner during the model training process. Effective measures to protect data privacy can better cope with the increasingly stringent data privacy and data security regulatory environment in the future.

Like in distributed machine learning settings, federated learning will also need to address Non-IID data. In [77] showed that with non-iid local data, performance can be greatly reduced for federated learning. The authors in response supplied a new method to address the issue similar to transfer learning.

3.3 Federated Learning vs Edge Computing

Federated learning can be seen as an operating system for edge computing, as it provides the learning protocol for coordination and security. In [69], authors considered generic class of machine learning models that are trained using gradient-descent based approaches. They analyze the convergence bound of distributed gradient descent from a theoretical point of view, based on which they propose a control algorithm that determines the best trade-off between local update and global parameter aggregation to minimize the loss function under a given resource budget.

3.4 Federated Learning vs Federated Database Systems

Federated Database Systems [57] are systems that integrate multiple database units and manage the integrated system as a whole. The federated database concept is proposed to achieve interoperability with multiple independent databases. A federated database system often uses distributed storage for database units, and in practice the data in each database unit is heterogeneous. Therefore, it has many similarities with federated learning in terms of the type and storage of data. However, the federated database system does not involve any privacy protection mechanism in the process of interacting with each other, and all database units are completely visible to the management system. In addition, the focus of the federated database system is on the basic operations of data including inserting, deleting, searching, and merging, etc., while the purpose of federated learning is to establish a joint model for each data owner under the premise of protecting data privacy, so that the various values and laws the data contain serve us better.

4 APPLICATIONS

As an innovative modeling mechanism that could train a united model on data from multiple parties without compromising privacy and security of those data, federated learning has a promising application in sales, financial, and many other industries, in which data cannot be directly aggregated for training machine learning models due to factors such as intellectual property rights, privacy protection, and data security.

Take the smart retail as an example. Its purpose is to use machine learning techniques to provide customers with personalized services, mainly including product recommendation and sales services. The data features involved in the smart retail business mainly include user purchasing power, user personal preference, and product characteristics. In practical applications, these three data features are likely to be scattered among three different departments or enterprises. For example, a user's purchasing power can be inferred from her bank savings and her personal preference can be analyzed from her social networks, while the characteristics of products are recorded by an e-shop. In this scenario, we are facing two problems. First, for the protection of data privacy and data security, data barriers between banks, social networking sites, and e-shopping sites are difficult to break. As a result, data cannot be directly aggregated to train a model. Second, the data stored in the three parties are usually heterogeneous, and traditional machine learning models cannot directly work on heterogeneous data. For now, these problems have not been effectively solved with traditional machine learning methods, which hinder the popularization and application of artificial intelligence in more fields.

Federated learning and transfer learning are the key to solving these problems. First, by exploiting the characteristics of federated learning, we can build a machine learning model for the three parties without exporting the enterprise data, which not only fully protects data privacy and data security, but also provides customers with personalized and targeted services and thereby achieves mutual benefits. Meanwhile, we can leverage transfer learning to address the data heterogeneity problem and break through the limitations of traditional artificial intelligence techniques. Therefore federated learning provides a good technical support for us to build a cross-enterprise, cross-data, and cross-domain ecosphere for big data and artificial intelligence.

One can use federated learning framework for multi-party database querying without exposing the data. For example, supposed in a finance application we are interested in detecting multi-party borrowing, which has been a major risk factor in the banking industry. This happens when certain users maliciously borrows from one bank to pay for the loan at another bank. Multi-party borrowing is a threat to financial stability as a large number of such illegal actions may cause the entire financial system to collapse. To find such users without exposing the user list to each other between banks A and B, we can exploit a federated learning framework. In particular, we can use the encryption mechanism of federated learning and encrypt the user list at each party, and then take the intersection of the encrypted list in the federation. The decryption of the final result gives the list of multi-party borrowers, without exposing the other "good" users to the other party. As we will see below, this operation corresponds to the vertical federated learning framework.

Smart healthcare is another domain which we expect will greatly benefit from the rising of federated learning techniques. Medical data such as disease symptoms, gene sequences, medical reports are very sensitive and private, yet medical data are difficult to collect and they exist in isolated medical centers and hospitals. The insufficiency of data sources and the lack of labels have led to an unsatisfactory performance of machine learning models, which becomes the bottleneck of current smart healthcare. We envisage that if all medical institutions are united and share their data to form a large medical dataset, then the performance of machine learning models trained on that large medical dataset would be significantly improved. Federated learning combining with transfer learning is the main way to achieve this vision. Transfer learning could be applied to fill the missing labels thereby expanding the scale of the available data and further improving the performance of a trained model. Therefore, federated transfer learning would play a pivotal role in the development of smart healthcare and it may be able to take human health care to a whole new level.

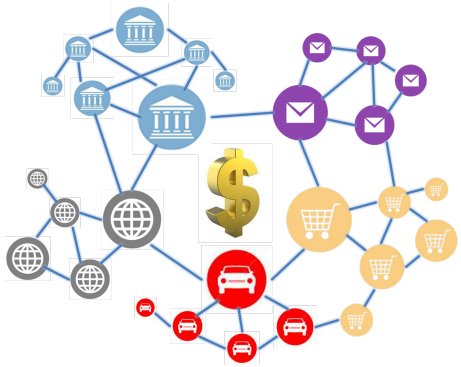


Fig. 5. Data Alliance Allocates the Benefits on Blockchain

5 FEDERATED LEARNING AND DATA ALLIANCE OF ENTERPRISES

Federated learning is not only a technology standard but also a business model. When people realize the effects of big data, the first thought that occurs to them is to aggregate the data together, compute the models through a remote processor and then download the results for further use. Cloud computing comes into being under such demands. However, with the increasing importance of data privacy and data security and a closer relationship between a company's profits and its data, the cloud computing model has been challenged. However, the business model of federated learning has provided a new paradigm for applications of big data. When the isolated data occupied by each institution fails to produce an ideal model, the mechanism of federated learning makes it possible for institutions and enterprises to share a united model without data exchange. Furthermore, federated learning could make equitable rules for profits allocation with the help of consensus mechanism from blockchain techniques. The data possessors, regardless of the scale of data they have, will be motivated to join in the data alliance and make their own profits. We believe that the establishment of the business model for data alliance and the technical mechanism for federated learning should be carried out together. We would also make standards for federated learning in various fields to put it into use as soon as possible.

6 CONCLUSIONS AND PROSPECTS

In recent years, the isolation of data and the emphasis on data privacy are becoming the next challenges for artificial intelligence, but federated learning has brought us new hope. It could establish a united model for multiple enterprises while the local data is protected, so that enterprises could win together taking the data security as premise. This article generally introduces the basic concept, architecture and techniques of federated learning, and discusses its potential in various applications. It is expected that in the near future, federated learning would break the barriers between industries and establish a community where data and knowledge could be shared together with safety, and the benefits would be fairly distributed according to the contribution of each participant. The bonus of artificial intelligence would finally be brought to every corner of our lives.

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*. ACM, New York, NY, USA, 308–318. https://doi.org/10.1145/2976749.2978318

[2] Abbas Acar, Hidayet Aksu, A. Selcuk Uluagac, and Mauro Conti. 2018. A Survey on Homomorphic Encryption Schemes: Theory and Implementation. *ACM Comput. Surv.* 51, 4, Article 79 (July 2018), 35 pages. https://doi.org/10.1145/3214303

[3] Rakesh Agrawal and Ramakrishnan Srikant. 2000. Privacy-preserving Data Mining. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (SIGMOD '00)*. ACM, New York, NY, USA, 439–450. https://doi.org/10.1145/342009.335438

[4] Yoshinori Aono, Takuya Hayashi, Le Trieu Phong, and Lihua Wang. 2016. Scalable and Secure Logistic Regression via Homomorphic Encryption. In *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy (CODASPY '16)*. ACM, New York, NY, USA, 142–144. https://doi.org/10.1145/2857705.2857731

[5] Toshinori Araki, Jun Furukawa, Yehuda Lindell, Ariel Nof, and 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 (CCS '16)*. ACM, New York, NY, USA, 805–817. https://doi.org/10.1145/2976749.2978331

[6] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. 2018. How To Backdoor Federated Learning. arXiv:cs.CR/1807.00459

[7] Raad Bahmani, Manuel Barbosa, Ferdinand Brasser, Bernardo Portela, Ahmad-Reza Sadeghi, Guillaume Scerri, and Bogdan Warinschi. 2017. Secure Multiparty Computation from SGX. In *Financial Cryptography and Data Security - 21st International Conference, FC 2017, Sliema, Malta, April 3-7, 2017, Revised Selected Papers*. 477–497. https://doi.org/10.1007/978-3-319-70972-7_27

[8] Dan Bogdanov, Sven Laur, and Jan Willemson. 2008. Sharemind: A Framework for Fast Privacy-Preserving Computations. In *Proceedings of the 13th European Symposium on Research in Computer Security: Computer Security (ESORICS '08)*. Springer-Verlag, Berlin, Heidelberg, 192–206. https://doi.org/10.1007/978-3-540-88313-5_13

[9] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17)*. ACM, New York, NY, USA, 1175–1191. https://doi.org/10.1145/3133956.3133982

[10] Florian Bourse, Michele Minelli, Matthias Minihold, and Pascal Paillier. 2017. Fast Homomorphic Evaluation of Deep Discretized Neural Networks. *IACR Cryptology ePrint Archive* 2017 (2017), 1114.

[11] Hervé Chabanne, Amaury de Wargny, Jonathan Milgram, Constance Morel, and Emmanuel Prouff. 2017. Privacy-Preserving Classification on Deep Neural Network. *IACR Cryptology ePrint Archive* 2017 (2017), 35.

[12] Kamalika Chaudhuri and Claire Monteleoni. 2009. Privacy-preserving logistic regression. In *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou (Eds.). Curran Associates, Inc., 289–296. http://papers.nips.cc/paper/3486-privacy-preserving-logistic-regression.pdf

[13] Fei Chen, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. 2018. Federated Meta-Learning for Recommendation. *CoRR* abs/1802.07876 (2018). arXiv:1802.07876 http://arxiv.org/abs/1802.07876

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

[15] W. Du and M. Atallah. 2001. Privacy-Preserving Cooperative Statistical Analysis. In *Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC '01)*. IEEE Computer Society, Washington, DC, USA, 102–. http://dl.acm.org/citation.cfm?id=872016.872181

[16] Wenliang Du, Yunghsiang Sam Han, and Shigang Chen. 2004. Privacy-Preserving Multivariate Statistical Analysis: Linear Regression and Classification. In *SDM*.

[17] Wenliang Du and Zhijun Zhan. 2002. Building Decision Tree Classifier on Private Data. In *Proceedings of the IEEE International Conference on Privacy, Security and Data Mining - Volume 14 (CRPIT '14)*. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 1–8. http://dl.acm.org/citation.cfm?id=850782.850784

[18] Cynthia Dwork. 2008. Differential Privacy: A Survey of Results. In *Proceedings of the 5th International Conference on Theory and Applications of Models of Computation (TAMC'08)*. Springer-Verlag, Berlin, Heidelberg, 1–19. http://dl.acm.org/citation.cfm?id=1791834.1791836

[19] EU. 2016. REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). *Available at: https://eur-lex. europa. eu/legal-content/EN/TXT* (2016).

[20] Boi Faltings, Goran Radanovic, and Ronald Brachman. 2017. *Game Theory for Data Science: Eliciting Truthful Information*. Morgan & Claypool Publishers.

[21] Jun Furukawa, Yehuda Lindell, Ariel Nof, and Or Weinstein. 2016. High-Throughput Secure Three-Party Computation for Malicious Adversaries and an Honest Majority. Cryptology ePrint Archive, Report 2016/944. https://eprint.iacr.

org/2016/944.
[22] Adrià Gascón, Phillipp Schoppmann, Borja Balle, Mariana Raykova, Jack Doerner, Samee Zahur, and David Evans. 2016. Secure Linear Regression on Vertically Partitioned Datasets. *IACR Cryptology ePrint Archive* 2016 (2016), 892.
[23] Robin C. Geyer, Tassilo Klein, and Moin Nabi. 2017. Differentially Private Federated Learning: A Client Level Perspective. *CoRR* abs/1712.07557 (2017). arXiv:1712.07557 http://arxiv.org/abs/1712.07557
[24] Irene Giacomelli, Somesh Jha, Marc Joye, C. David Page, and Kyonghwan Yoon. 2017. Privacy-Preserving Ridge Regression with only Linearly-Homomorphic Encryption. Cryptology ePrint Archive, Report 2017/979. https://eprint.iacr.org/2017/979.
[25] O. Goldreich, S. Micali, and A. Wigderson. 1987. How to Play ANY Mental Game. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing (STOC '87)*. ACM, New York, NY, USA, 218–229. https://doi.org/10.1145/28395.28420
[26] Rob Hall, Stephen E. Fienberg, and Yuval Nardi. 2011. Secure multiple linear regression based on homomorphic encryption. *Journal of Official Statistics* 27, 4 (2011), 669–691.
[27] Stephen Hardy, Wilko Henecka, Hamish Ivey-Law, Richard Nock, Giorgio Patrini, Guillaume Smith, and Brian Thorne. 2017. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. *CoRR* abs/1711.10677 (2017).
[28] Ehsan Hesamifard, Hassan Takabi, and Mehdi Ghasemi. 2017. CryptoDL: Deep Neural Networks over Encrypted Data. *CoRR* abs/1711.05189 (2017). arXiv:1711.05189 http://arxiv.org/abs/1711.05189
[29] Briland Hitaj, Giuseppe Ateniese, and Fernando Pérez-Cruz. 2017. Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning. *CoRR* abs/1702.07464 (2017).
[30] Qirong Ho, James Cipar, Henggang Cui, Jin Kyu Kim, Seunghak Lee, Phillip B. Gibbons, Garth A. Gibson, Gregory R. Ganger, and Eric P. Xing. 2013. More Effective Distributed ML via a Stale Synchronous Parallel Parameter Server. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1 (NIPS'13)*. Curran Associates Inc., USA, 1223–1231. http://dl.acm.org/citation.cfm?id=2999611.2999748
[31] Murat Kantarcioglu and Chris Clifton. 2004. Privacy-Preserving Distributed Mining of Association Rules on Horizontally Partitioned Data. *IEEE Trans. on Knowl. and Data Eng.* 16, 9 (Sept. 2004), 1026–1037. https://doi.org/10.1109/TKDE.2004.45
[32] Alan F. Karr, X. Sheldon Lin, Ashish P. Sanil, and Jerome P. Reiter. 2004. Privacy-Preserving Analysis of Vertically Partitioned Data Using Secure Matrix Products.
[33] Niki Kilbertus, Adria Gascon, Matt Kusner, Michael Veale, Krishna Gummadi, and Adrian Weller. 2018. Blind Justice: Fairness with Encrypted Sensitive Attributes. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Jennifer Dy and Andreas Krause (Eds.), Vol. 80. PMLR, Stockholmsmässan, Stockholm Sweden, 2630–2639. http://proceedings.mlr.press/v80/kilbertus18a.html
[34] Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. 2018. On-Device Federated Learning via Blockchain and its Latency Analysis. arXiv:cs.IT/1808.03949
[35] Miran Kim, Yongsoo Song, Shuang Wang, Yuhou Xia, and Xiaoqian Jiang. 2018. Secure Logistic Regression Based on Homomorphic Encryption: Design and Evaluation. *JMIR Med Inform* 6, 2 (17 Apr 2018), e19. https://doi.org/10.2196/medinform.8805
[36] Jakub Konecný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. 2016. Federated Optimization: Distributed Machine Learning for On-Device Intelligence. *CoRR* abs/1610.02527 (2016). arXiv:1610.02527 http://arxiv.org/abs/1610.02527
[37] Jakub Konecný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated Learning: Strategies for Improving Communication Efficiency. *CoRR* abs/1610.05492 (2016). arXiv:1610.05492 http://arxiv.org/abs/1610.05492
[38] Gang Liang and Sudarshan S Chawathe. 2004. Privacy-preserving inter-database operations. In *International Conference on Intelligence and Security Informatics*. Springer, 66–82.
[39] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William J. Dally. 2017. Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training. *CoRR* abs/1712.01887 (2017). arXiv:1712.01887 http://arxiv.org/abs/1712.01887
[40] Jian Liu, Mika Juuti, Yao Lu, and N. Asokan. 2017. Oblivious Neural Network Predictions via MiniONN Transformations. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17)*. ACM, New York, NY, USA, 619–631. https://doi.org/10.1145/3133956.3134056
[41] H. Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. 2016. Federated Learning of Deep Networks using Model Averaging. *CoRR* abs/1602.05629 (2016). arXiv:1602.05629 http://arxiv.org/abs/1602.05629
[42] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2017. Learning Differentially Private Language Models Without Losing Accuracy. *CoRR* abs/1710.06963 (2017).

[43] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. 2018. Inference Attacks Against Collaborative Learning. *CoRR* abs/1805.04049 (2018). arXiv:1805.04049 http://arxiv.org/abs/1805.04049

[44] Payman Mohassel and Peter Rindal. 2018. ABY3: A Mixed Protocol Framework for Machine Learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18)*. ACM, New York, NY, USA, 35–52. https://doi.org/10.1145/3243734.3243760

[45] Payman Mohassel, Mike Rosulek, and Ye Zhang. 2015. Fast and Secure Three-party Computation: The Garbled Circuit Approach. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15)*. ACM, New York, NY, USA, 591–602. https://doi.org/10.1145/2810103.2813705

[46] Payman Mohassel and Yupeng Zhang. 2017. SecureML: A System for Scalable Privacy-Preserving Machine Learning. In *IEEE Symposium on Security and Privacy*. IEEE Computer Society, 19–38.

[47] Payman Mohassel and Yupeng Zhang. 2017. SecureML: A System for Scalable Privacy-Preserving Machine Learning. *IACR Cryptology ePrint Archive* 2017 (2017), 396.

[48] Valeria Nikolaenko, Udi Weinsberg, Stratis Ioannidis, Marc Joye, Dan Boneh, and Nina Taft. 2013. Privacy-Preserving Ridge Regression on Hundreds of Millions of Records. In *Proceedings of the 2013 IEEE Symposium on Security and Privacy (SP '13)*. IEEE Computer Society, Washington, DC, USA, 334–348. https://doi.org/10.1109/SP.2013.30

[49] Richard Nock, Stephen Hardy, Wilko Henecka, Hamish Ivey-Law, Giorgio Patrini, Guillaume Smith, and Brian Thorne. 2018. Entity Resolution and Federated Learning get a Federated Resolution. *CoRR* abs/1803.04035 (2018). arXiv:1803.04035 http://arxiv.org/abs/1803.04035

[50] Sinno Jialin Pan and Qiang Yang. 2010. A Survey on Transfer Learning. *IEEE Trans. on Knowl. and Data Eng.* 22, 10 (Oct. 2010), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

[51] Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. 2018. Privacy-Preserving Deep Learning via Additively Homomorphic Encryption. *IEEE Trans. Information Forensics and Security* 13, 5 (2018), 1333–1345.

[52] M. Sadegh Riazi, Christian Weinert, Oleksandr Tkachenko, Ebrahim M. Songhori, Thomas Schneider, and Farinaz Koushanfar. 2018. Chameleon: A Hybrid Secure Computation Framework for Machine Learning Applications. *CoRR* abs/1801.03239 (2018).

[53] R L Rivest, L Adleman, and M L Dertouzos. 1978. On Data Banks and Privacy Homomorphisms. *Foundations of Secure Computation, Academia Press* (1978), 169–179.

[54] Bita Darvish Rouhani, M. Sadegh Riazi, and Farinaz Koushanfar. 2017. DeepSecure: Scalable Provably-Secure Deep Learning. *CoRR* abs/1705.08963 (2017). arXiv:1705.08963 http://arxiv.org/abs/1705.08963

[55] Ashish P. Sanil, Alan F. Karr, Xiaodong Lin, and Jerome P. Reiter. 2004. Privacy Preserving Regression Modelling via Distributed Computation. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04)*. ACM, New York, NY, USA, 677–682. https://doi.org/10.1145/1014052.1014139

[56] Monica Scannapieco, Ilya Figotin, Elisa Bertino, and Ahmed K. Elmagarmid. 2007. Privacy Preserving Schema and Data Matching. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data (SIGMOD '07)*. ACM, New York, NY, USA, 653–664. https://doi.org/10.1145/1247480.1247553

[57] Amit P. Sheth and James A. Larson. 1990. Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. *ACM Comput. Surv.* 22, 3 (Sept. 1990), 183–236. https://doi.org/10.1145/96602.96604

[58] Reza Shokri and Vitaly Shmatikov. 2015. Privacy-Preserving Deep Learning. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15)*. ACM, New York, NY, USA, 1310–1321. https://doi.org/10.1145/2810103.2813687

[59] David Silver, Aja Huang, Christopher J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature* 529 (2016), 484–503. http://www.nature.com/nature/journal/v529/n7587/full/nature16961.html

[60] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. 2017. Federated Multi-Task Learning. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 4424–4434. http://papers.nips.cc/paper/7029-federated-multi-task-learning.pdf

[61] Shuang Song, Kamalika Chaudhuri, and Anand D. Sarwate. 2013. Stochastic gradient descent with differentially private updates. *2013 IEEE Global Conference on Signal and Information Processing* (2013), 245–248.

[62] Lili Su and Jiaming Xu. 2018. Securing Distributed Machine Learning in High Dimensions. *CoRR* abs/1804.10140 (2018). arXiv:1804.10140 http://arxiv.org/abs/1804.10140

[63] Latanya Sweeney. 2002. K-anonymity: A Model for Protecting Privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* 10, 5 (Oct. 2002), 557–570. https://doi.org/10.1142/S0218488502001648

[64] Jaideep Vaidya and Chris Clifton. [n. d.]. Privacy Preserving Naive Bayes Classifier for Vertically Partitioned Data. In *in Proceedings of the fourth SIAM Conference on Data Mining, 2004*. 330–334.

[65] Jaideep Vaidya and Chris Clifton. 2002. Privacy Preserving Association Rule Mining in Vertically Partitioned Data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '02)*. ACM, New York, NY, USA, 639–644. https://doi.org/10.1145/775047.775142

[66] Jaideep Vaidya and Chris Clifton. 2003. Privacy-preserving K-means Clustering over Vertically Partitioned Data. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '03)*. ACM, New York, NY, USA, 206–215. https://doi.org/10.1145/956750.956776

[67] Jaideep Vaidya and Chris Clifton. 2005. Privacy-Preserving Decision Trees over Vertically Partitioned Data. In *Data and Applications Security XIX*, Sushil Jajodia and Duminda Wijesekera (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 139–152.

[68] Li Wan, Wee Keong Ng, Shuguo Han, and 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)*. ACM, New York, NY, USA, 775–783. https://doi.org/10.1145/1281192.1281275

[69] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K. Leung, Christian Makaya, Ting He, and Kevin Chan. 2018. When Edge Meets Learning: Adaptive Control for Resource-Constrained Distributed Machine Learning. *CoRR* abs/1804.05271 (2018). arXiv:1804.05271 http://arxiv.org/abs/1804.05271

[70] Wikipedia. 2018. https://en.wikipedia.org/wiki/Facebook-Cambridge_Analytica_data_scandal.

[71] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2018. Federated Learning. *Communications of The CCF* 14, 11 (2018), 49–55.

[72] 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, Washington, DC, USA, 160–164. http://dl.acm.org/citation.cfm?id=1382436.1382751

[73] Hwanjo Yu, Xiaoqian Jiang, and 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)*. ACM, New York, NY, USA, 603–610. https://doi.org/10.1145/1141277.1141415

[74] Hwanjo Yu, Jaideep Vaidya, and 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)*. Springer-Verlag, Berlin, Heidelberg, 647–656. https://doi.org/10.1007/11731139_74

[75] Jiawei Yuan and Shucheng Yu. 2014. Privacy Preserving Back-Propagation Neural Network Learning Made Practical with Cloud Computing. *IEEE Trans. Parallel Distrib. Syst.* 25, 1 (Jan. 2014), 212–221. https://doi.org/10.1109/TPDS.2013.18

[76] Qingchen Zhang, Laurence T. Yang, and Zhikui Chen. 2016. Privacy Preserving Deep Computation Model on Cloud for Big Data Feature Learning. *IEEE Trans. Comput.* 65, 5 (May 2016), 1351–1362. https://doi.org/10.1109/TC.2015.2470255

[77] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. 2018. Federated Learning with Non-IID Data. arXiv:cs.LG/1806.00582