CONTRIBUTED PAPER

Trusted AI in Multiagent Systems: An Overview of Privacy and Security for Distributed Learning

This article provides an exhaustive overview of attacks and defensive mechanisms on privacy and security for distributed learning on four different levels, namely sharing data, sharing model, sharing knowledge, and sharing results.

By Chuan Ma, *Member IEEE*, Jun Li, *Senior Member IEEE*, Kang Wei, *Member IEEE*, Bo Liu, *Senior Member IEEE*, Ming Ding, *Senior Member IEEE*, Long Yuan, Zhu Han, *Fellow IEEE*, and H. Vincent Poor, *Life Fellow IEEE*

ABSTRACT | Motivated by the advancing computational capacity of distributed end-user equipment (UE), as well as the increasing concerns about sharing private data, there has been considerable recent interest in machine learning (ML) and artificial intelligence (AI) that can be processed on distributed UEs. Specifically, in this paradigm, parts of an ML process are outsourced to multiple distributed UEs. Then, the processed information is aggregated on a certain level at a central server, which turns a centralized ML process into a distributed one and brings about significant benefits. However, this new distributed ML paradigm raises new risks in terms of privacy and security issues. In this article, we provide a survey of the emerging security and privacy risks of distributed ML from a unique perspective of information exchange levels, which are defined according to the key steps of an ML process, i.e., we consider the following levels: 1) the level of preprocessed data; 2) the level of learning models; 3) the level of extracted knowledge; and 4) the level of intermediate results. We explore and analyze the potential of threats for each information exchange level based on an overview of current state-of-the-art attack mechanisms and then discuss the possible defense methods against such threats. Finally, we complete the survey by providing an outlook on the challenges and possible directions for future research in this critical area.

KEYWORDS | Distributed machine learning (ML); federated learning (FL); multiagent systems; privacy; security; trusted artificial intelligence (AI).

Manuscript received 17 December 2022; accepted 1 August 2023. Date of current version 15 September 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFF0712100; in part by the National Natural Science Foundation of China under Grant 62002170 and Grant 61902184; in part by the Fundamental Research Funds for the Central Universities under Grant 30921013104; in part by the Science and Technology on Information Systems Engineering Laboratory under Grant WDZC20205250411; in part by the Future Network Grant of Provincial Education Board in Jiangsu; in part by the Youth Foundation Project of Zhejiang Laboratory under Grant K2023PD0AA01; in part by the Research Initiation Project of Zhejiang Laboratory; and in part by the U.S. National Science Foundation, U.S. Department of Transportation, Toyota and Amazon, under Grant CNS-2107216, Grant CNS-2128368, Grant CNS-2128448, Grant CMMI-2222810, Grant ECCS-2302469, and Grant ECCS-2335876.
(Corresponding authors: Jun Li; Kang Wei.)

Chuan Ma is with the Zhejiang Laboratory, Hangzhou 311121, China, and also with the Key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, Nanjing 211189, China (e-mail: chuan.ma@zhejianglab.edu.cn).

Jun Li is with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210096, China (e-mail: jun.li@njust.edu.cn).

Kang Wei is with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210096, China, and also with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China (e-mail: kang.wei@njust.edu.cn).

Bo Liu is with the School of Computer Science, University of Technology Sydney, Sydney, NSW 2007, Australia (e-mail: bo.liu@uts.edu.au).

Ming Ding is with Data61, CSIRO, Sydney, NSW 2015, Australia (e-mail: Ming.Ding@data61.csiro.au).

Long Yuan is with the School of Computer Science, Nanjing University of Science and Technology, Nanjing 210096, China (e-mail: longyuan@njust.edu.cn).

Zhu Han is with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77004 USA, and also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul 446-701, South Korea (e-mail: hanzhu22@gmail.com).

H. Vincent Poor is with the Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: poor@princeton.edu).

Digital Object Identifier 10.1109/JPROC.2023.3306773

NOMENCLATURE

Abbr.	Definition
ML	Machine learning.
DL	Deep learning.
RL	Reinforcement learning.

DQN	Deep Q network.
AC	Actor critic.
A3C	Asynchronous advantage actor critic.
TRPO	Trust region policy optimization.
PG	Policy gradient.
PPO	Proximal policy optimization.
DP	Differential privacy.
HE	Homomorphic encryption.
SMPC	Secure multiparty computation.
SGD	Stochastic gradient descent.
FL	Federated learning.
NN	Neural network.

I. INTRODUCTION

An explosive growth in data availability arising from proliferating Internet of Things (IoT) and 5G/6G technologies, combined with the availability of increasing computational resources through cloud and data servers, promotes the applications of ML in many domains (e.g., finance, health care, industry, and smart city). ML technologies, e.g., DL, have revolutionized the ways that information is extracted with ground-breaking successes in various areas [1]. Meanwhile, owing to the advent of IoT, the number of intelligent applications with edge computing, such as smart manufacturing, intelligent transportation, and intelligent logistics, is growing dramatically.

As such, conventional centralized DL can no longer efficiently process the dramatically increasing amount of data from the massive numbers of IoT or edge devices. For example, as shown in Fig. 1, it is expected that the volume of data will be 181 ZB in 2025.[1] In addition, the long runtime of training models steers solution designers toward using distributed systems for an increase of parallelization and the total amount of wireless bandwidth, as the training data required for sophisticated applications can easily be on the order of terabytes [2]. Examples include transaction processing for larger enterprises on data that is stored in different locations [3] or astronomical data that is too large to move and collect [4].

To address this challenge, distributed learning frameworks have emerged. A typical distributed learning setting involves the cooperation of multiple clients and servers, which, thus, involves a decentralization and aggregation process along with the ML process [5]. With the increasing capability of edge devices, distributed clients are able to execute simple ML tasks. For example, FL [6], [7], [8] enables the decoupling of data provisioning by distributed clients and aggregating ML models at a centralized server. In certain ML tasks, the model sometimes can be so large that it cannot be trained in a reasonable amount of time and cannot run completely on a single machine. Therefore, large-scale distributed ML is proposed in [9] where datasets in each client will be reanalyzed and pretrained locally, and the knowledge is aggregated by a central server. In addition, aggregating learning results [10] by the server is another part of distributed ML technology.

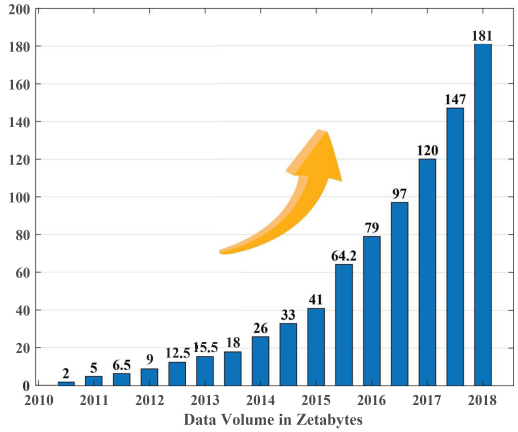


Fig. 1. *Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2025.*

To complete an ML task successfully, we need to preserve the integrity and security of the system, along with the privacy of participating clients. As the manufacturers can potentially fail to implement a robust security system in distributed devices, experts on security have warned of potential risks of large numbers of unsecured devices connecting to the Internet [11]. Security and privacy are very significant issues for distributed ML, which introduce a new level of emergent concerns for participants. This is because these devices collect not only personal and sensitive information, e.g., names and telephone numbers, but also monitor daily activities. Due to the regular stream of news stories about privacy leakage through major data breaches, users are wary of using personal data in public or private ML tasks for good reason [12].

There are some related surveys on security and privacy issues in distributed ML. For example, the challenges and opportunities of distributed learning over conventional (centralized) ML were discussed in [13] and [14], which elaborated on limited privacy and security issues. De Cristofaro [15] and Liu et al. [16] focused on the adversarial models related to private information leakage and corresponding defensive mechanisms in ML, and the work [17] investigated privacy issues in distributed ML. Moreover, DP-based protection methods were introduced in [18]. In addition, to protect the privacy of the IoT data, the work [19] surveyed ML-based methods to address privacy issues, including scalability, interoperability, and limitations on resources, such as computation and energy. The works [20], [21], [22] addressed security and privacy issues in FL, together with related solutions. The summary of related surveys on security and privacy issues in ML is listed in Table 1.

Different from the abovementioned surveys, in this work, we consider the following.

[1]https://www.statista.com/statistics/871513/worldwide-data-created/

Table 1 Existing Surveys on Private and Secure ML

Related Survey	Topic	Key contributions
[20]	Privacy preservation in federated learning for IoT data	This work mainly focused on the survey on the use of federated learning for private data analysis in IoT, i.e., highly skewed non-IID data with high temporal variability, to address privacy concerns, bandwidth limitations and power/compute limitations.
[19]	Machine learning-based solutions to protect privacy in IoT	This work surveyed works that leverage machine learning as a strategy to address the privacy issues in IoT including scalability, inter-operability, and resource limitation such as computation and energy.
[23]	Data security issues in deep learning	This survey investigated the potential threats of deep learning with respect to black and white box attacks and presented related countermeasures on offense and defense.
[18]	Differentially private machine learning	This survey investigated existing differentially private machine learning technologies and categorized them as the Laplace/Gaussian/exponential mechanism and the output /objective perturbation mechanism
[13], [14]	Machine learning in distributed systems	These articles provided an overview by outlining the challenges and opportunities of distributed machine learning over conventional (centralized) machine learning, and discussing available techniques.
[20]–[22]	Attacks and defensive strategies on federated deep learning	These works investigated existing vulnerabilities of FL and subsequently provided a literature study of defensive strategies and algorithms for FL aimed to overcome these attacks.
[15], [16], [24]	Privacy in machine learning	These surveys focused on machine learning and algorithms related to private information leakage and corresponding defensive mechanisms.
[17]	Privacy in distributed machine learning	This work focused on privacy leakage issues in distributed learning and studied benefits, limitations, and trade-offs for defensive algorithms.
Our paper	Privacy and Security in distributed learning	Our work is different from the above survey articles in the following aspects: 1. our work first develops a distributed framework into four levels; 2. the state-of-the-art of privacy and security issues at each level are investigated and summarized; 3. the characteristics of the adversary at each level are further discussed.

1) We first give a clear and fresh definition of distributed learning and develop the distributed learning framework in four levels in terms of sharing different information, namely, sharing data, sharing models, sharing knowledge, and sharing results.
2) We then provide an extensive overview of the current state of the art related to the attacks and defensive mechanisms related to privacy and security for each level. Real examples are also listed for each level.
3) In addition, learned lessons from each aspect are described, which it is hoped can help readers to avoid potential mistakes.
4) Several research challenges and future directions are further discussed, which can provide insights into the design of advanced learning paradigms.

II. BACKGROUND OF DISTRIBUTED ML AND THIS ARTICLE STRUCTURE

In Section II, we first describe the detailed process of how an ML task is executed and then transit the centralized learning to distributed paradigms and develop a decentralized learning framework. In addition, we provide descriptions of several widely studied distributed learning frameworks.

A. Background of ML

Generally speaking, the core idea of ML algorithms can be summarized as training a machine to learn rules or patterns underlying some phenomenon using data, and then making decisions or inferences based on new data using the learned rules or patterns. Many ML algorithms fall into the category of pattern recognition (PR), including face recognition, voice recognition, character recognition, and so on [25]. Since humans cannot easily program machines to follow all detailed rules and judgments, ML can be used to help machines learn hidden and even implied rules by themselves. This process is described simply as follows.

Suppose we are going to train a machine to classify whether a fruit is an apple or a banana (a classification task). We first collect some samples that can be labeled and learned by the machine (dataset). So, some apples and bananas from this dataset along with their features, including shape, color, weight, size, and so on, are recorded. Now, a labeled fruit (apple or banana) with a set of ground-truth features together builds up a sample, and these labeled samples constitute the training dataset. The goal of this ML task is to make the machine learn features from the training dataset and output good predictions given new samples without labels (test dataset). This learning process can be expressed as fitting a function that takes the features as inputs and outputs a value that is as close as possible to the true label. Fig. 2 illustrates the procedure of ML with four main steps listed as follows.

1) *Data Collection:* The quantity and quality of the collected data dictate how accurate the model is, and the dataset can be divided into training, validation, and test datasets [26].
2) *Model Training:* For different ML tasks, an appropriate model should be chosen wisely first. Then, the training dataset with the right labels is fed as inputs to the model to start training.
3) *Knowledge Extraction:* During training, features of the input samples are extracted using some metrics or combinations of metrics (e.g., linear or nonlinear combinations), and this knowledge helps the model update its weights in structures.
4) *Result Prediction:* The test dataset, which has been withheld from the model building process, is used and outputs the prediction results, such as labels, values,

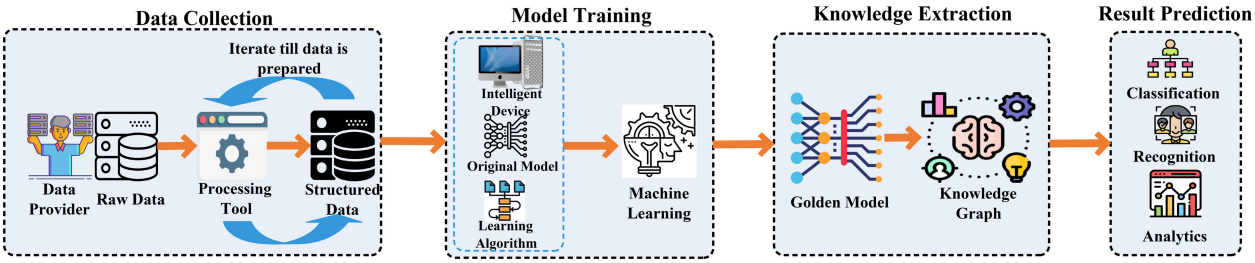


Fig. 2. *Process of ML in four key steps: data collection, model training, knowledge extraction, and result prediction.*

vectors (e.g., generative time series), and matrices (e.g., generative images).

B. Background of Distributed ML

Distributed ML systems and algorithms have been extensively studied in recent years to scale up ML in the presence of big data. The existing work focuses either on the theoretical convergence speed of proposed algorithms or on the practical system aspects to reduce the overall model training time [27]. Bulk synchronous parallel (BSP) algorithms [28] are among the first distributed ML algorithms. Due to hash constraints on the computation and communication procedures, these schemes share a convergence speed that is similar to traditional synchronous and centralized gradient-like algorithms. The stale synchronous parallel (SSP) algorithm [29] is a more practical alternative that abandons strict iteration barriers and allows distributed workers to be off synchrony up to a certain bounded delay. The convergence results have been developed for both gradient descent and SGD [29], [30], [31] as well as proximal gradient methods [32] under different assumptions of loss functions. In fact, SSP has become central to various types of current distributed parameter server architectures [33], [34], [35], [36]. Depending on how the workload is partitioned [27], distributed ML systems can be categorized into four levels.

1) *Level 0 (Sharing Data):* After collecting and preprocessing data locally, each user equipment (UE) will upload its private/anonymized data to a central server, and then, the server will use this aggregated data to complete the learning task.
2) *Level 1 (Sharing Models):* Different from uploading data directly, each UE can train a local ML model using its own data and share the trained model with the server. Then, the server will aggregate the collected model and retransmit the global model to UEs for the next round of learning.
3) *Level 2 (Sharing Knowledge):* Different from sharing ML models, the extracted knowledge from training local data, such as the relationship between different attributes, is further shared.
4) *Level 3 (Sharing Results):* The task training is completely processed locally, and each UE only shares ML results/outputs to the central server.

The detailed framework of the four-level distributed ML is illustrated in Fig. 3, which is composed of a local and global plane. In the local plane, different information, i.e., data or models, is processed and generated in local devices and then transmitted to a centralized server for aggregation. Four levels of the proposed distributed learning framework are described in detail, i.e., sharing data, sharing models, sharing knowledge, and sharing results, which are exemplified by representative ML techniques.

C. Existing Distributed Learning Frameworks

In this section, we will introduce some widely used distributed learning models, which include federated learning, split learning, SGD-based collaborative learning, and multiagent RL (MARL).

1) Federated Learning: FL is a collaborative ML technique [37], [38], [39] developed by Google, which allows decoupling of data provision at UEs, and ML model aggregation, such as network parameters of DL, at a centralized server. A structure of FL is illustrated in Fig. 4. The purpose of FL is to cooperatively learn a global model without directly sharing data. In particular, FL has distinct privacy advantages compared with data center training on a dataset. At a server, holding even an anonymized dataset can put client privacy at risk via linkage to other datasets. In contrast, the information transmitted for FL consists of minimal updates to improve a particular ML model. The updates can be ephemeral and will not contain more information than the raw training data (by the data processing inequality). Furthermore, the source of the updates is not needed by the aggregation algorithm, and so, updates can be transmitted without identifying metadata over a mixed network, such as Tor [40] or via a trusted third party. General categories are distributed horizontal FL, where clients have different sample spaces with the same feature space and share models during aggregation, distributed vertical FL with the same sample space and different feature spaces, sharing models or knowledge to the central server, and distributed transfer learning with various sample and feature spaces when uploading model or knowledge in aggregation [41].

However, although the data are not explicitly shared in the original format, it is still possible for adversaries to reconstruct the raw data approximately,

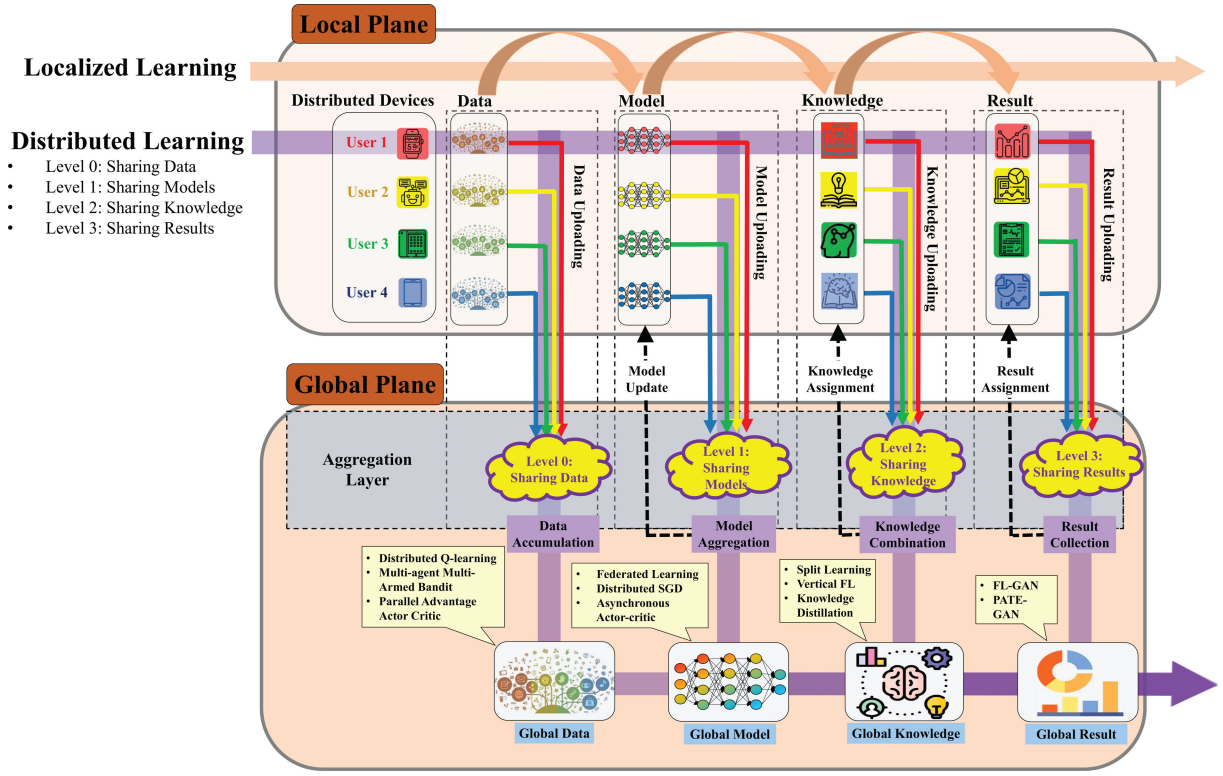


Fig. 3. ***Framework of distributed learning, which is composed of a local and global plane. In the local plane, different information, i.e., data or models, is processed and generated in local devices and then transmitted to a centralized server for aggregation. Four levels of the proposed distributed learning framework are described in detail, i.e., sharing data, sharing models, sharing knowledge, and sharing results, which are exemplified by representative ML techniques.***

especially when the architecture and parameters are not completely protected. In addition, FL can expose intermediate results, such as parameter updates, from an optimization algorithm, such as SGD, and the transmission of these gradients may actually leak private information when exposed together with a data structure, such as image pixels. Furthermore, the well-designed attacks, such as inference attacks (stealing membership information) [42], [43], [44] and poisoning attacks (polluting the quality of datasets or parameter models) [45], may induce further security issues.

2) Split Learning: Split learning, as a type of distributed DL [17], [47], [48], [49], is also called split NN (SplitNN). Similar to FL, split learning is effective when data uploading is not available because of privacy or legal restrictions. In SplitNN, each participant first trains an NN until a predefined layer, called the cut layer, and then transmits the output of the cut layer to the server. Upon receiving the outputs, a central server will continue training the remaining layers. Then, the loss function value is calculated and backpropagated to the participants. When receiving the feedback, the participants continue the backpropagation until the network finishes training. In Fig. 5, we show a combination of FL and split learning, where the logits are shared and aggregated at a centralized server.

The computational and communication costs on the client side are reduced in split learning, because only part of the network is processed locally. In addition, instead of transmitting the raw data, the activation function of the cut layer is uploaded to the server, which has a relatively smaller size. Some experimental results show that split learning has higher performances and fewer costs than FL over figure classification tasks, i.e., Canadian Institute For Advanced Research-100 (CIFAR-100) datasets, using Resnet-50 architectures for hundreds of clients-based setups [47]. However, it needs further explanations on

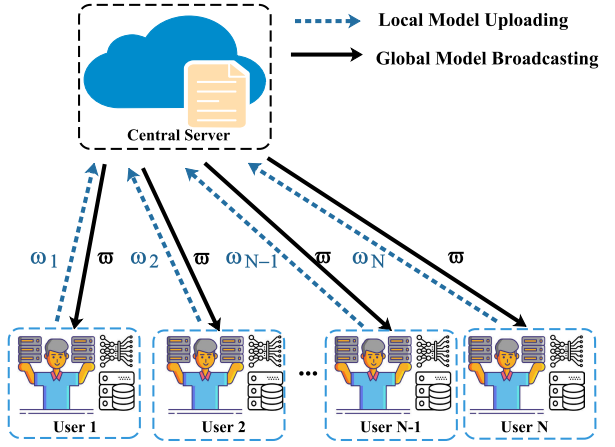


Fig. 4. ***Structure of FL, where users train an ML model using their local data and share the models to a centralized server.***

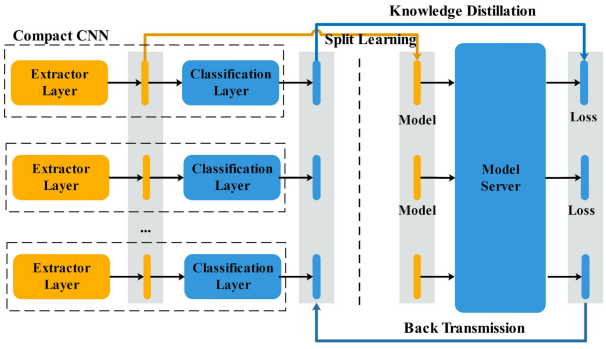


Fig. 5. ***Reformulation of FL assisted by the split learning and knowledge distillation [46].***

how split learning works and makes decisions, which is linked to the trust of distributed networks, especially in the healthcare area [47].

3) Large Batch Synchronous SGD: The difference between large batch synchronous SGD (LBS-SGD)-based collaborative learning and FL lies in that the updates in LBS-SGD are processed on each batch of training data, and multiple epochs of local training are required before uploading in FL. In LBS-SGD, model parallelism and data parallelism are two common ways to support updating, such as distributed large mini-batch SGD [50], distributed synchronous SGD with backups [17], [51], and selective SGD [52]. In [52], each participant is asked to choose a part of the model to update at each epoch and share it asynchronously with others. The work [50] considered synchronous SGDs by dividing local epochs into mini-batches over multiple clients and model aggregations. While the aggregated updates were performed synchronously in [50] that the aggregator will wait for all clients, stragglers may slow down the learning, and a synchronous optimization with backup participants have been considered in [51].

4) Multiagent RL: RL is trial-and-error learning by interacting directly with the environment, training according to feedback, and finally achieving the design goal. Specifically, RL defines a decision maker as an agent and the interaction with the environment, where three essential elements: the state, action, and reward, are used to describe the interaction. For each interaction, the client arrives at a certain state and processes a corresponding action and then obtains feedback that is used to alter the current state to the next state. However, a single RL framework cannot address complex real-world problems, and thus, MARL has attracted increasing attention. Within MARL, agents will cooperate and observe the complex environment more comprehensively. For example, as shown in Fig. 6, a three-agent RL system, where actions and rewards are shared between different users, is provided. By absorbing the learning experiences from the user-self and other participants, a faster convergence rate with better performance is always achieved. However, compared with the single-agent setting, controlling multiple agents poses several additional challenges, such as the heterogeneity of participants, the design of achieved goals, and a more serious malicious client problem. Although several methods have been proposed to address these challenges, e.g., approximate AC [53] and lenient DQN, limitations, such as nonseasonal communication among agents and privacy leakage, prevent the rapid development of MARL, and the existing methods cannot be extended to large-scale multiagent scenarios.

Following the discussed background of distributed ML, we present the structure of this survey work in Fig. 7. The rest of this article is structured as follows. In Section III, privacy and security issues are discussed, and several robust protection methods are provided in Section IV. Then, in Section V, we survey attacks and defenses in various paradigms in distributed ML. Several research challenges and future directions are shown in Section VI. Finally, conclusions are drawn in Section VII. In addition, a list of important abbreviations is provided in Nomenclature.

III. PRIVACY AND SECURITY RISKS IN DISTRIBUTED ML

In Section III, we will introduce the potential risks of privacy and security, which comprise several factors, including threat models, adversarial models, and attack methods.

A. Threat Models

1) Malicious/Curious Participant: Participants in distributed ML can be malicious or curious. For example, a car insurance company with limited user attributes might want to improve its risk evaluation model by incorporating more attributes from other businesses, e.g., a bank, a taxation office, and so on. The role of the other participants is simply to provide additional feature information without directly disclosing their data to other participants and, in return, obtain financial and/or reputation rewards.

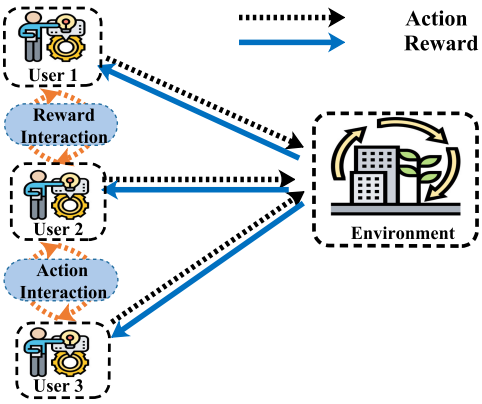


Fig. 6. ***Framework of MARL, where multiple users communicate and interact to change information and also process actions to obtain feedback from the environment.***

However, competitors may be disguised as collaborators and then damage the training process or steal the ML model.

2) External Attackers: In terms of exchanged information, eavesdropping, modification, or deletion can occur during communication in distributed ML as well. The exchanged information usually contains the updated direction and extracted features from private data, and thus, it is crucial to ensure its correctness, especially for the client–server framework. An external attacker may control the final output by modifying or deleting the exchanged information in the communication. In addition, via eavesdropping on the extracted features from private data, an external attacker can further infer sensitive information [54].

B. Adversarial Models

In this section, we will discuss adversarial goals related to leaking information from the training data or destroying models during learning.

1) Access:

a) *White Box:* The adversary is assumed to acknowledge certain information about the training data or the learning model, e.g., model parameters, network structures, or part of/the whole training dataset.

b) *Black Box:* The adversary does not have any knowledge about the ML model, but the adversary can further explore the model by injecting some designed inputs and observing related outputs [55].

2) Training Versus Inference: The second factor is the place where the attack happens.

a) *Training Stage:* The adversary attempts to learn the model by accessing a part or all of the training data and creating a substitute model, i.e., a shadow model.

b) *Inference Stage:* The adversary observes the outputs from the learning and sums up the model characteristics [54].

3) Passive Versus Active: A third factor is to distinguish between passive and active attacks.

a) *Passive Attack:* The adversary can passively observe and obtain the updates but change nothing during the training process.

b) *Active Attack:* The adversary actively performs and adjusts the learning operation. For example, the adversary can upload unreasonable parameters to degrade the aggregated model in FL [56].

C. Attack Methods

In this section, several attack methods are investigated as follows.

1) Poisoning Attack: The goal of a poisoning attack is to degrade the model quality, which misleads the learning into taking an incorrect direction by carefully crafting poisoning samples during training, also called adversarial examples [57]. In the black-box attack, the attacker can only inject a relatively small amount of crafted/poisoned data into the training model, where the amount and the undiscovered capability of these poisoning data are two basic metrics to estimate the attacking performance. For example, Jagielski et al. [58] have first investigated poisoning attacks against linear regression models and proposed a fast optimization algorithm with limited crafting samples to perturb outputs. Furthermore, Suciu et al. [59] have investigated the minimum information required by the attacker to achieve various attacking goals. In the white-box attack, the adversaries have full knowledge of the training model and can take advantage of it to reconstruct a powerful poisoning attack. For example, Yuan et al. [60] have proposed a white-box attack with perfect knowledge under different goals. Although the mentioned method might be unrealistic in practical settings, it can achieve almost five more than the black-box attack in success rate.

2) Evasion Attack: An evasion attack often happens in the prediction process, which aims to mislead the outputs. In detail, the evasion attack is to change real data from one category to a determined or random one and destroy the integrity of the original dataset. From a black-box attack angle, the adversary only knows the type of the training dataset and observes the outputs. Based on this assumption, Kwon et al. [61] have realized this kind of attack in a speech recognition system. The generated adversarial samples achieve a 91.67% success rate on moving one data from one category to another. Alternatively in a white-box attack, the adversary is able to acknowledge more useful information, such as the network structure and the type of training samples, rather than the predictive interface. For example, Eykholt et al. [62] have shown the weakness of DNNs when random noises are added to the inputs, and an advanced robust physical perturbations-based method has been proposed.

3) Model Inversion Attack: The model inversion attack proposed in [63] works in a black-box fashion, in which the adversary can choose inputs and observe the corresponding outputs. This information is then used to detect correlations between unknown inputs and respective outputs. A follow-up work has presented a combination with a black-and-white box attack [43]. The proposed attack aims to predict the highest probability of one input for a given label, by way of which the adversary is able to reconstruct the input for a known label, i.e., a figure from a specific class. However, the proposed model inversion attack only works in linear models for most cases, and a major weakness is that the complexity grows exponentially with the input size, since it relies on searching all linear combinations by brute force.

4) Membership Inference Attack: The membership inference attack (MIA) is mainly focused on privacy attacks. A previous attack targeting distributed recommender systems [64] intended to infer which input will lead to a

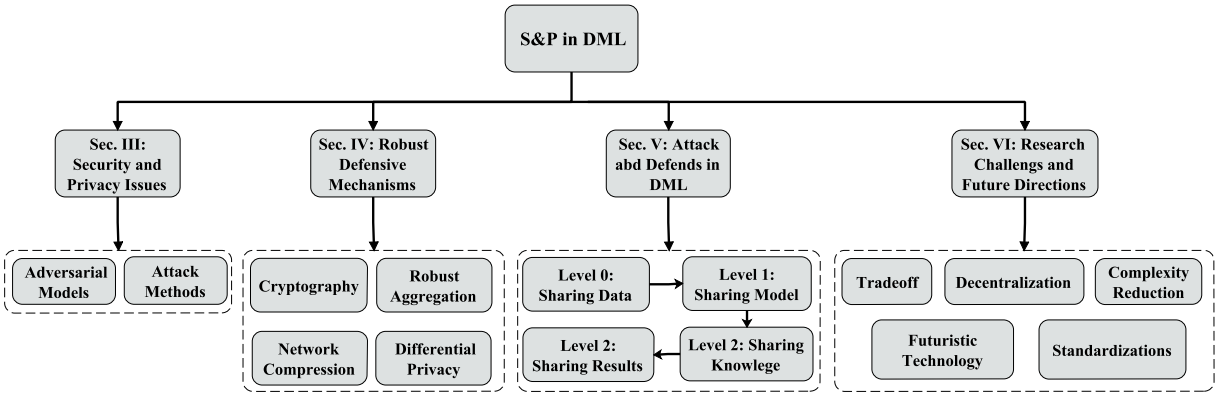


Fig. 7. ***Structure of the survey with key elements.***

change in the output by observing temporal patterns: from the learning model. Shokri et al. [54] have investigated the differences between the models to infer whether an input exists in the training dataset for the supervised model. In particular, a shadow model constructs a similar structure to the targeted model in a black-box fashion. Following [54], Song et al. [65] attempted to record the training data with black-box access. Subsequently, Ateniese et al. [66] have exploited the knowledge of learning models to hide the an underlying Markov model and attack support vector machine (SVM) in classification tasks. Also, related works [44], [67], [68] presented inference attacks against distributed DL [37], [52]. In particular, Phong et al. [67] aimed to attack the privacy-preserving learning framework proposed in [52] and revealed that partial data samples can be revealed by an honest-but-curious server. However, operation with a single-point batch size limits its effectiveness. Also, a white-box attack against [52] has been proposed in [44], which used generative adversarial networks (GANs) to produce similar samples with a targeted training dataset; however, the proposed algorithm loses effectiveness in black-box access. Finally, Truex et al. [69] have shown that MIAs are usually data-driven, and Melis et al. [68] have demonstrated the way that a malicious participant can infer sensitive properties in distributed learning. Other MIAs focused on genomic research studies [70], [71], in which the attack is designed to infer the presence of specific information of individuals within an aggregated genomic dataset [71], locations [72], and noisy statistics in general [73].

5) Model and Functionality Stealing:

a) *Model Extraction:* The aim of model extraction is first proposed in [74], in which the authors proposed to infer the parameters from a trained classifier in a black-box fashion; however, it only works when the adversary has access to the predictions, i.e., the probabilities for each class in a classification task. In follow-up works, other researchers went a step further to steal hyperparameters [75], which are external configurations. These values cannot be estimated from data samples, architecture extraction [76] that infers the deep model structure as well as the updating tools [e.g., SGD or alternating direction method of multipliers (ADMM)], and so on.

b) *Functionality Extraction:* The concept of functionality extraction is, rather than stealing the model, to create knock-off models. Orekondy et al. [77] have created such an attack based only on design inputs and relative outputs to observe correlation from ML as a service (MLaaS) queries. In particular, the adversary uses the input–output pairs, e.g., image–prediction pairs in a figure classification task, to train a knock-off model, and compares it with one of the victims for the same task. In addition, Papernot et al. [55] have trained a shadow model to replace a DNN, which directly uses inputs generated by the attacker and labeled by the attacking DNN.

D. Section Summary

In summary, the attack target can be regarded as a clue to distinguish the privacy and security risks from the adversary aspect. A common aim for the privacy attack is to infer membership of participants without degrading the learning performance, i.e., MIA, and model and functionality stealing, while malicious clients usually aim to destroy the integrity of the learning system, i.e., model poisoning, evasion, and inversion attack.

IV. ROBUST DEFENSIVE MECHANISMS

In Section IV, we will present an overview of several robust defensive mechanisms that include cryptography, robust aggregation, network compression, and DP to reduce information leakage and address security issues.

A. Cryptography

Cryptography is a vital part of distributed ML, as it has the ability to support confidential secure computing scenarios. Many algorithms and protoypes have been

proposed in the literature, which allow participants to obtain learning outputs without uploading their raw data to the server. For instance, in the supervised ML task, SMPC and HE-based privacy-enhancing tools have been proposed to enable secure computing. Typical examples are NNs [78], [79], [80], matrix factorization [81], linear regressions [82], decision trees [83], and linear classifiers [84], [85].

Specifically, SMPC allows two or more participants to jointly complete an ML task over the shared data without revealing it to others. Popular SMPC prototypes are usually developed for two parties, such as [80], [82], [86], and [87], designed for distributed ML tasks. For more than two parties, algorithms based on three-party communication have been provided in [88], [89], [90], and [91], which all rely on the majority of semi-honest or honest participants. For example, Bonawitz et al. [78] have proposed a mixture of several communicating schemes to enable secure computing of participants in FL by blurring the aggregation from the server.

With regard to HE, it mainly uses the encryption and decryption protocol to transform the original message by certain mathematical operations, and there are three common forms for HE: 1) partially HE (PHE) supports one type of mathematical operation; 2) somewhat HE (SWHE) that uses a number of mathematical operations for limited use cases; and 3) fully HE (FHE) supports unlimited numbers of mathematical operations with no other limits [92]. For example, Phong et al. [67] have developed a novel homomorphic scheme based on additive operations for FL with no performance degradation [67]. Other distributed learning strategies, such as [93] and [94], used HE to encrypt data, and the central server can train a learning model based on the encrypted one. However, the drawbacks of HE are obvious. First, it is usually hard or even impractical to implement HE, since this will generate a huge computation overhead [87], [92], [95]. Second, with the increasing number of homomorphic operations, the size of the encrypted models grows exponentially, especially in the SWHE [92], which usually largely surpasses the original model. Third, extra communications between the client and server are required to facilitate key-sharing protocols, which will increase communication costs.

B. Robust Aggregation

The robust aggregation protection methods are designed for distributed ML in which a server needs to aggregate something from clients. To prevent malicious clients, or a group of collusive malicious clients, such as the Byzantine attack in FL [96], Blanchard et al. [97] have proposed Krum, a robust aggregation scheme. By minimizing the sum of squared Euclidean distances over the aggregated models, Krum can effectively recognize and remove these outliers. Several follow-ups [98], [99], [100] aimed to recognize malicious clients. In addition, Chang et al. [101] have developed a knowledge-sharing-based algorithm to preserve privacy. The proposed Cronus algorithm relies on a public dataset that is available to all clients. Instead of transmitting parameters, clients will upload the predicted results from this public dataset, and a mean estimation algorithm [102] was used to aggregate these high-dimensional label samples. Although Cronus has been proven to defend against basic model poisoning attacks with an acceptable performance loss, sharing labels will lead to privacy leakage to a certain extent.

C. Network Compression

The main purpose of compressing the network is to reduce information transmission, which saves communication resources and accelerates learning. As well, it can also reduce the information exposed to the adversary. Typical methods include quantization [103], [104], [105], network sparsification [106], [107], knowledge distillation [108], [109], network pruning [110], [111], and sketch [112], [113], [114]. Specifically, an initial work [52] proposed the idea of transmitting a subset of all gradients in distributed SGD, and based on it, Yoon et al. [115] have proposed a novel gradient subset scheme that uploads sparse gradients, and the chosen gradients can improve the prediction accuracy in non-independent and identically distributed (non-IID) settings. However, as the gradients keep their own form, recent works [42], [116] have shown that such methods cannot prevent a specific adversary from inferring available information from these frameworks [42], [116].

Another approach is using lossy compression techniques to decrease the transmitted bits, and it may facilitate certain forms of information security. Reisizadeh et al. [117] quantized the updates using the low-precision quantizer proposed in [103] and provided a smooth trade-off between compression rate and the convergence performance in convex and non-convex settings. In [118], a count sketch method with momentum and error accumulation was provided for FL while achieving a high compression rate with good convergence. On the basis of it, Li et al. [114] have proved such a quantization method can provide a certain DP guarantee. Moreover, a sketch-based method was proposed in [113], which sorts gradient values into buckets and encodes them with bucket indexes. In addition, a stochastic-sign-based gradient compressor was used and analyzed to enable communication efficiency [119], and an autoencoder compressor was proposed in [120] in which the autoencoder is trained based on dummy gradients, and the server will release the coded part to clients while keeping the decoder part secretive.

Different from the above methods, a technique called dropout can also be used to defend [121], although it is usually used to prevent overfitting problems in training [122]. By applying dropout, there will be no deterministic outputs (e.g., the updating gradients) on the same training dataset, which can reduce the exploitable attack surface [42].

Table 2 Taxonomy of Attacks in Level-0 Distributed ML With Sharing Data

Issue	Ref.	Attacker's knowledge	Learning Model	Effectiveness
Adversarial examples	[133]	White-box, black-box	Inception v2, Inception v3, Inception v4, Resnet v2-152	Attacking a white-box model with a near 100% success rate and more than 50% for black-box models
	[135]	White-box, black-box	DQN, A3C, TRPO	Physically interfering with the observations of the victim
	[136]	Black-box	AC	Directly attacking actions to achieve the designated purposes
	[137]	Black-box	AC	Taking actions to induce natural observations (environment dynamic) that are adversarial to the victim
Feature identification	[138]	A small amount of information about an individual subscriber	-	Identifying the Netflix records of known users, uncovering users' preferences and other sensitive information

D. Differential Privacy

DP is a standard definition for privacy estimation [123]. A query mechanism is first defined as a property to a dataset, and DP-based analytical methods are then extended for ML models on private training data, such as SVM [124], linear regression [125], and DL [52], [126]. On NNs, differentially private SGD [126] is the most widely applied method that adds random noises on the updating gradients to achieve DP guarantee.

DP sets up a game where the adversary is trying to determine whether a training model has an input D or D', which are adjacent datasets and only differ in one sample. If the adversary can distinguish which dataset (D or D') is used to train by observing the outputs, we can say this training model leaks private information. A formal definition of (ϵ, δ)-DP is expressed as follows.

Definition 1: ((ϵ, δ)-DP). A randomized mechanism $f : \mathbf{D} \mapsto \mathcal{R}$ offers (ϵ, δ)-DP if for any adjacent inputs $d, d' \in \mathbf{D}$ and $S \subset \mathcal{R}$

$$\Pr\left[f\left(d\right) \in S\right] \leq e^{\epsilon} \Pr\left[f\left(d'\right) \in S\right] + \delta \tag{1}$$

where $f(d)$ denotes a random function of d.

To estimate the accumulated privacy budget in multiple learning iterations, the composition theory in [123] shows the effectiveness, and other variants on DP [127], [128] use slightly different formulations with (1) and can achieve a tighter privacy delimitation. Recently, Nasr et al. [129] have derived a lower bound on DP from the adversary perspective, and the Monte Carlo-based method is the first trial of obtaining the privacy level empirically. In addition, the concept of local DP (LDP) was proposed first in [130] and [131] and has gradually become accepted.

E. Section Summary

In summary, general defensive schemes, such as cryptography, robust aggregation, and network compression, can provide thorough protection on security and preserve privacy, where the application of DP is particularly useful for privacy issues.

V. ATTACKS AND DEFENSES AT VARIOUS LEVELS OF DISTRIBUTED LEARNING

In Section V, we will provide a detailed discussion of the state of the art of attacks and defenses at each level of distributed ML.

A. Level 0: Sharing Data

Data collection plays an important role in various data-governed distributed ML algorithms. However, original data usually contain sensitive information, such as medical records, salaries, and locations, and thus, a straightforward release of data is not appropriate. Correspondingly, research on protecting the privacy of individuals and the confidentiality of data with an acceptable performance loss has received increasing attention from many fields, such as computer science, statistics, economics, and social science.

1) Threat Models: Although the existing works have proposed a number of mechanisms to hide identifiers of the raw data, it is also possible for attackers to steal privacy by analyzing hidden features [132]. Moreover, deep NNs have been proven vulnerable to adversarial examples, which poses security concerns due to the potentially severe consequences [133]. This means that if some adversaries successfully make adversarial examples participate in system training, the training performance will be unacceptable.

2) Taxonomy of Attacks: Attacks on data publishing models can be mainly categorized as adversarial examples and feature identification based on their goals. As shown in Table 2, we summarize possible attacks as follows.

a) *Adversarial Examples (Data Poisoning):* The work in [133] integrated the momentum term into the iterative process for attacks and generated more transferable adversarial examples by stabilizing update directions and escaping from poor local maxima during the generating iterations. The research on this area is faced with an "arms race" between attacks and defenses, i.e., a defense method

Table 3 Taxonomy of Defenses in Level-0 Distributed ML With Data Sharing

Method	Ref.	Use case	Key idea	Effectiveness
Adversarial training	[139]	Against adversarial examples	Formulating a minimax optimization problem, Parameterizing the adversarial distributions	Improving model security and robustness
Anonymization	[141]	Removing unique identifiers of spatiotemporal trajectory datasets	Clustering the trajectories using a variation k-means algorithm	Enhancing the k-anonymity metric of privacy
	[142]	Motion data	A multi-objective loss function involving an information-theoretic approach	Concealing user's private identity
	[143]	Image and video	Conditional generative adversarial networks	Removing the identifying characteristics of faces and bodies for privacy
Dummy	[144]–[147]	Tabular dataset	Generating fake samples to hide real one	Realizing k-anonymity or similar metrics for privacy
	[149]	Balance MIT-BIH arrhythmia dataset	Generative adversarial networks (GANs)	Generating high quality dummy samples for privacy
DP	[130], [131], [151], [152]	Localized or tabular dataset	Using random response to perturb the value of local data	Achieving LDP for privacy
	[154]	PAC-learning from distributed data	General upper and lower bounds for quantities such as the teaching-dimension	Achieving DP without incurring any additional communication penalty for privacy
	[155]	Communication bandwidth limitation and security concerns of data upload	Training autoencoder, Transmitting latent vectors	Reducing the communications overhead, and protecting the data of the end users
Encryption	[159]	Enforcement of access policies, Support of policies updates	Defining their own access policies over user attributes and enforce the policies on the distributed data	Securely manage the data distributed
	[156]	Complete ML workflow by enabling the execution of a cooperative GD	Multiparty homomorphic encryption	Preserving data and model confidentiality with up to $N-1$ colluding parties
	[157]	Distributed training data, a large volume of the shared data portion.	Data locality property of Apache Hadoop architecture, a limited number of cryptographic operations	Achieving privacy-preservation with an affordable computation overhead
Others	[158]	A learner with a distributed set of nodes	Establishing a game-theoretic framework to capture the conflicting interests between the adversary and data processing units	Obtaining the network topology with a strong relation to the resiliency

proposed to prevent the existing attacks will be soon evaded by new attacks.

b) *Feature Identification:* Although many works have proposed efficient methods to process original data in order to preserve sensitive information, many feature identification attacks are emerging to expose hidden information. As one of the feature identification attacks, structure-based de-anonymization attacks on graph data have been proposed, which aim to de-anonymize the private users in terms of their uniquely distinguishable structural characteristics [134].

3) Taxonomy of Defenses: Many defensive mechanisms have been designed against the aforementioned attacks, as shown in Table 3, and we will discuss various defenses as follows.

a) *Adversarial Training:* Adversarial training is among the most effective techniques to improve model robustness by augmenting training data with adversarial examples. The work in [139] has proposed an adversarial distributional training (ADT) framework, which is formulated as a min–max optimization problem and improves the model robustness. In this framework, the inner maximization aims to learn an adversarial distribution to characterize the potential adversarial examples around a natural one under an entropic regularizer, and the outer minimization aims to train robust models by minimizing the expected loss over the worst-case adversarial distributions.

b) *Anonymization:* An anonymization operation comes in several flavors: generalization, suppression, anatomization, permutation, and perturbation [140], [141]. These techniques aim to remove or hide identifying characteristics from raw data while guaranteeing the data utility. An information-theoretic approach has been formulated and proposed a new multiobjective loss function for training deep autoencoders [142], which helps to minimize user-identity information as well as data distortion to preserve the application-specific utility. The work in [143] has proposed the conditional identity anonymization GANs (CIA-GANs) model, which can remove

the identifying characteristics of faces and bodies while producing high-quality images and videos that can be used for various computer vision tasks, such as detection or tracking. Unlike previous methods, CIA-GAN has full control over the de-identification (anonymization) procedure, ensuring both anonymizations as well as diversity. In summary, the choice of anonymization operations has an implication for the search space of anonymous tables and data distortion. The full-domain generalization has the smallest search space with the largest distortion, and the local recording scheme has the largest search space but the least distortion.

c) *Dummy Data:* The existing methods to protect data privacy mainly focus on the protection of the users' identities through anonymity. User attributes can be classified into identity information, quasi-identifier, and sensitive information. Given an anonymity table, if the attributes in the table have not been properly treated, an adversary may deduce the relationship between the user's identity and sensitive information according to the user's quasi-identifier, such as age and gender. A popular approach for data anonymity is k anonymity, and any record in a k-anonymized dataset has a maximum probability $1/k$ of being reidentified [144], [145], [146]. The privacy models l diversity and t closeness in [147] further refine the concept of diversity and require that the distribution of the sensitive values of each equivalent class should be as close as to the overall distribution of the dataset. The common rules for these algorithms are basically to produce dummy records to hide the real ones. In addition, the dummy-based methods also work for location privacy protection. Dummy data along with the true one will be sent to the server from users, which may hide the client's contribution during training [148]. Because the collection is processed on the server, the system performance can still be guaranteed. As an efficient method to generate realistic datasets, GANs provide an alternative to balance user privacy and training performance. The work in [149] has proposed a novel data augmentation technique based on the combination of real and synthetic heartbeats using GAN to improve the classification of electrocardiogram (ECG) heartbeats of 15 different classes from the Massachusetts Institute of Technology-Boston's Beth Israel Hospital (MIT-BIH) arrhythmia dataset.[2]

d) *DP:* As a promising solution, a mechanism is said to be differentially private [123] if the computation result of a dataset is robust to any change of an individual sample. Several differentially private ML algorithms [150] have been developed in the community, where a trusted data curator is introduced to gather data from individual owners and honestly runs the private algorithms. Compared with DP, LDP [130], [131] eliminates the need for a trusted data curator and is more suitable for distributed ML. Randomized aggregatable privacy-preserving ordinal response (RAPPOR) [151], which applies LDP by Google, is designed to collect the perturbed data samples from multiple data owners. Besides simple counting, a follow-up paper [152] shows that RAPPOR can also compute other types of statistics, such as joint-distribution estimation and association testing. Besides RAPPOR, an alternative way that achieves DP is to add random noise to the sample value before publishing it [130], [153]. To process this method, a numerical sample is always normalized, and a categorical one is transformed to the same range by one-hot coding. In addition, Balcan et al. [154] adopted the DP algorithm to handle the privacy concern in a communication problem in which each distributed client needs to transmit data to an aggregation center to learn a model. The work [155] has proposed distributed edge computing for image classification, where each edge will upload its raw data after coding to latent data to protect privacy.

e) *Encryption:* The work in [156] has instantiated scalable privacy-preserving distributed learning (SPINDLE), an operational distributed system that supports the privacy-preserving training, and evaluation of generalized linear models on distributed datasets. Moreover, it relies on a multiparty HE scheme to execute high-depth computations on encrypted data without significant overhead. The work in [157] has proposed a distributed algorithm for distributed data, where privacy is achieved by the data locality property of the Apache Hadoop architecture, and only a limited number of cryptographic operations are required.

f) *Others:* The work in [158] has aimed to develop secure, resilient, and distributed ML algorithms under adversarial environments. This work has established a game-theoretic framework to capture the conflicting interests between the adversary and a set of distributed data processing units. The Nash equilibrium of the game allows for predicting the outcome of learning algorithms in adversarial environments and enhancing the resilience of the ML through dynamic distributed learning algorithms.

4) *Real Examples for Level-0 Distributed ML:*

a) *RAPPOR:* This provides a privacy-preserving way to learn software statistics to better safeguard users security, find bugs, and improve the overall user experience. Building on the concept of randomized response, RAPPOR enables learning statistics about the behavior of users software while guaranteeing client privacy [151]. The guarantees of DP, which are widely accepted as the strongest form of privacy, have almost never been used in practice despite intense academic research. RAPPOR

[2]https://www.physionet.org/content/mitdb/1.0.0/

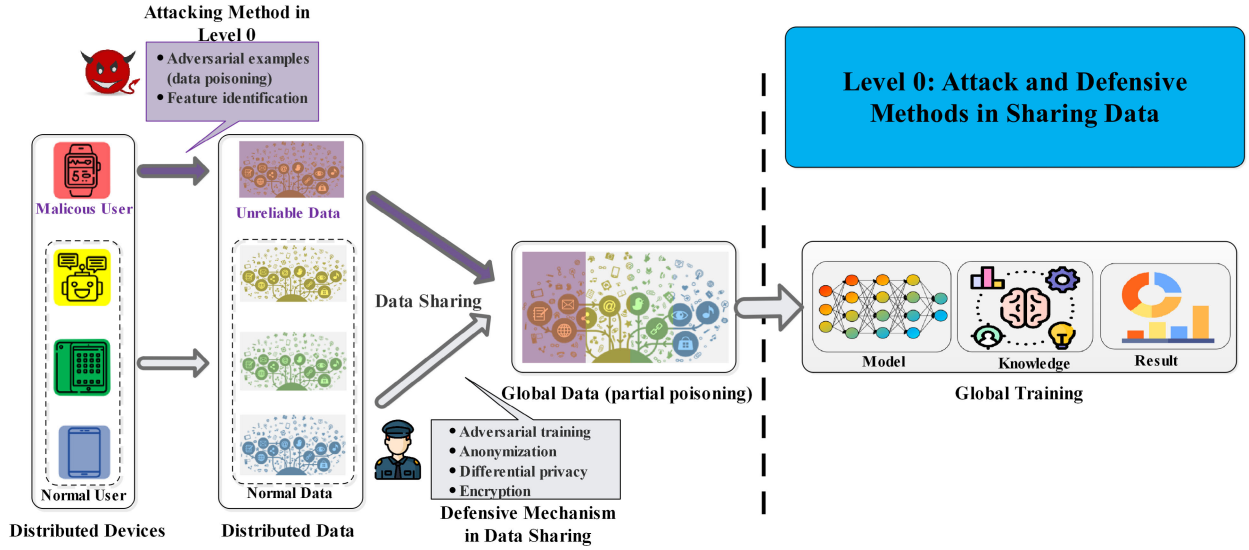


Fig. 8. ***Breakout figure from Fig. 3: an illustration of privacy and security issues in Level-0 distributed learning with data sharing.***

introduces a practical method to achieve those guarantees. In detail, the core of RAPPOR is a randomized response mechanism [160] for a user to answer a yes/no query to the record aggregator. A classic example is to collect statistics about a sensitive group, in which the aggregator asks each individual: "Are you a doctor?" To answer this question, each individual tosses a coin, gives the true answer if it is a head, and a random yes/or answer otherwise. This randomized approach provides plausible deniability to the individuals. Meanwhile, it is shown to satisfy ϵ-LDP, and the strength of privacy protection (i.e., ϵ) can be controlled by using a biased coin. Based on the collected randomized answers, the aggregator estimates the percentage of users whose true answer is "yes" (resp. "no"). RAPPOR allows the software to send reports that are effectively indistinguishable and are free of any unique identifiers. RAPPOR is currently an available implementation in Chrome, which learns statistics about how unwanted software is hijacking users settings.

b) *DP in the IOS System:* Apple has adopted and further developed LDP to enable Apple to learn about the user community while avoiding learning about individuals [161]. DP perturbs the information shared with random noise before it ever leaves the user's device, such that Apple can never reproduce the raw data. The power of additive noise that has been added can be reduced without exposing raw data from users by averaging out over large numbers of data points, and meaningful information emerges. DP is utilized as the first step of a system for data analysis that consists of robust privacy protections at every stage. The system is optional and developed to provide transparency to users. Device identifiers are removed from the data, and it is transmitted to Apple over an encrypted channel. The Apple analysis system ingests the differentially private contributions, dropping IP addresses and other metadata. The final stage is aggregation, where the private records are processed to compute the relevant statistics, and the aggregate statistics are then shared with relevant Apple teams. Since both the ingestion and aggregation stages are performed in a restricted access environment, the raw data are not broadly accessible to the public.

5) Brief Summary: The guarantee of privacy and security in terms of data sharing models relies on the preprocessing of the raw data, such as perturbation, dummy data, anonymization, and encryption. As shown in Fig. 8, data preprocessing happens at the first stage of an ML task, and thus, these preprocessing techniques are usually harmful to the utility of systems or involve extra computations. Therefore, it is more practical to select a proper mechanism to hide sensitive information from shared data while alleviating the negative influences on the system's utility.

B. Level 1: Sharing Model

In model sharing systems, all distributed nodes must share their training models with the central server or other participants. Via the interaction between independent data training and local model aggregation, model sharing systems can capture a required learning model over data that resides at the associated nodes.

1) Threat Models: Although data are not required to be uploaded in model sharing systems, private information can still be divulged by analyzing uploaded model parameters, e.g., weights trained in deep NNs. Moreover, adversarial participants may degrade or even destroy the training systems by uploading unreliable models. Attacks can be carried out by the following three aspects.

a) *Insiders Versus Outsiders:* Insider attacks include those launched by the server and the participants in

the model sharing systems. Outsider attacks include those launched by eavesdroppers in the wireless transmission environment between participants and the server and by users of the final model when it is deployed as a service. Insider attacks are generally stronger than outsider attacks, as it strictly enhances the capability of the adversary.

b) *Semi-Honest Versus Malicious:* Under the semi-honest setting, adversaries are considered passive or honest but curious. They try to learn the private states of other participants without deviating from the model sharing protocol. Passive adversaries are assumed to only observe the aggregated or averaged gradient, but not the training data or gradient from other honest participants. Under the malicious setting, an active or malicious adversary tries to learn the private states of honest participants and deviates arbitrarily from the model sharing protocol by modifying, replaying, or removing messages. This strong adversary model allows the adversary to conduct particularly devastating attacks.

c) *Poisoning Versus Inference:* Attacks at the poisoning phase attempt to learn, influence, or corrupt the model sharing itself. During the poisoning phase, the attacker can run data poisoning attacks to compromise the integrity of training dataset collection or launch model poisoning attacks to compromise the integrity of the learning process. The attacker can also launch a range of inference attacks on an individual participant's update or on the aggregation of updates from all participants.

2) Taxonomy of Attacks: Attacks to model sharing models can be categorized as poisoning attacks, inference attacks, and model inversion based on their various goals, as shown in Table 4. We also summarize them as follows.

a) *Poisoning Attack:* Client compromised by attackers always have opportunities to poison the global model in model sharing systems, in which local models are continuously updated by clients throughout their deployments. Moreover, the existence of compromised clients may induce further security issues, such as bugs in preprocessing pipelines, noisy training labels, and explicit attacks that target training and deployment pipelines [186]. In order to destroy ML models, poisoning attackers may control a subset of clients and manipulate their outputs sent to the server. For example, the compromised clients can upload noisy and reversed models to the server at each communication round [176], [187], which has the advantage of low complexity to mount attacks. Other attackers may manipulate the outputs of the compromised clients carefully to achieve the evasion of defenses and downgrade the performance of ML models. Furthermore, Fang et al. [163] and Baruch et al. [188] have formulated the local model poisoning attack as an optimization problem and then apply this attack against Byzantine-robust FL methods. In this way, attackers can improve the success rate of attacks, dominate the cluster and change the judgment boundary of the global model, or make the global model deviate from the right direction. Besides, attackers may hope to craft the ML model to minimize this specific objective function instead of destroying it. Via using multiple local triggers and model-dependent triggers (i.e., generated based on local models of attackers), the collusive attackers can conduct backdoor attacks successfully [189]. Bagdasaryan et al. [45] have developed and evaluated a generic constrain-and-scale technique that incorporates the evasion of defenses into the attacker's loss function during training. The work in [162] has explored the threat of model poisoning attacks on FL initiated by a single, non-colluding malicious client where the adversarial objective is to cause the model to misclassify a set of chosen inputs with high confidence.

b) *Inference Attack:* The work in [188] has presented a new attack paradigm, in which a malicious opponent may interfere with or backdoor the process of distributed learning by applying limited changes to the uploaded parameters. The work in [45] has proposed a new model-replacement method that demonstrated its efficacy on poisoning models of standard FL tasks. Inferring privacy information about clients for attackers is also possibly achievable in ML models. A generic attacking framework mGAN-artificial intelligence (AI) that incorporates a multitask GAN has been proposed in [190], which conducted novel discrimination on client identity, achieving attack to clients' privacy, i.e., discriminating a participating party's feature values, such as category, reality, and client identity.

c) *Model Inversion:* By casting the model inversion task as an optimization problem, which finds the input that maximizes the returned confidence, the work in [43] has recovered recognizable images of people's faces given only their names and accesses to the ML model. In order to identify the presence of an individual's data, an attack model trained by the shadow training technique has been designed and can successfully distinguish the target model's outputs on members versus nonmembers of its training dataset [52].

Specifically, in distributed RL (DRL) systems, there is literature available on security vulnerabilities. We provide many characteristics of an adversary's capabilities and goals that can be studied as follows. First, we divide attacks based on what components in a Markov decision process (MDP) the attacker chooses to attack: the agent's observations, actions, and environment (transition) dynamics. Then, we discuss the practical scenarios where attacks happen on these components.

1) *Observations:* The existing work on attacking DRL systems with adversarial perturbations focuses on

Table 4 Taxonomy of Attacks in Level-1 Distributed ML With Model Sharing

Issue	Ref.	Attacker's knowledge	Learning Model	Effectiveness
Model poisoning	[45]	Black-box	LSTM, ResNet	Manipulating the RL to achieve the designated purposes
	[162]	Black-box	CNN	Manipulating the RL to achieve the designated purposes
	[163]	White-box, Black-box	LR, CNN	Destroying the system performance
Inference attacks (Snooping attack)	[42]	Black-box	CNN	Inferring certain sensitive characteristics of clients, such as locations and gender, etc.
	[164]	Black-box access to the trained policy, access to the state space, the action space, the initial state distribution, and the reward function	DQN, PG, PPO	Inferring certain sensitive characteristics of the training environment transition dynamics, such as dynamics coefficients, environment transition dynamics
	[165]	Black-box	DQN, A2C	Consistently predicting RL agents' future actions with high accuracy
Model inversion	[44]	Black-box	CNN	Reconstructing raw training data
	[166]	Black-box	CNN	Reconstructing the actual training samples without affecting the standard training

perturbing an agent's observations, i.e., states and rewards, that are communicated between the agent and the environment. This is the most appealing place to start, with seminal results already suggesting that recognition systems are vulnerable to adversarial examples [135], [191], [192], [193], [194], [195], [196], [197], [198], [199]. Huang et al. [135] have shown that adversarial attacks are also effective when targeting NN policies in RL adversarial examples. Based on this technique, some of the works enhance adversarial examples to attack DRL. To improve the attack efficiency, the strategically timed attack [191], consuming a small subset of time steps in an episode, has been explored. Via stamping a small percentage of inputs of the policy network with a Trojan trigger and manipulating the associated rewards, the work in [195] has proposed the TrojDRL attack, which can deteriorate drastically the policy network in both targeted and untargeted settings. Another idea for a reward-poisoning attack is to design an adaptive disturbing strategy [196], where the infinity norm constraint is adjusted on the DRL agent's learning process at different time steps. For the theoretical analysis, two standard victims with adversarial observations, i.e., tabular certainty equivalence learner in RL and linear quadratic regulator in control, have been analyzed in a convex optimization problem in which global optimality and the attack feasibility and attack cost have been provided [194]. In addition, the effectiveness of a universal adversarial attack against DRL interpretations (i.e., UADRLI) has been verified by the theoretical analysis [197], from which the attacker can add the crafted universal perturbation uniformly to the environment states in a maximum number of steps to incur minimal damage. In order to stealthily attack the DRL agents, the work in [198] has injected adversarial samples in a minimal set of critical moments while causing the most severe damage to the agent. Another work in [199] has formulated an optimization framework in a stealthy manner to find an optimal attack for different measures of attack cost and solved it with an offline and online setting.

2) *Actions:* Attacks applied on the action space usually aim to minimize the expected return or lure the agent to a designated state, e.g., the action outputs can be modified by installing a virus in the actuator executing process. This can be realistic in certain robotic control tasks where the control center sends some control signals to the actuator. A vulnerability in the implementation, i.e., the vulnerability in the blue-tooth signal transmission, may allow an attacker to modify that signal [200]. A training policy network to learn the attack has been developed, which treats the environment and the original policy together as a new environment and views attacks as actions [136]. However, the existing works only concentrate on the white-box scenario, i.e., knowing the victim's learning process and observations, which is not practical and is inaccessible to attackers.
3) *Environment Dynamics:* The environment (transition) dynamics can be defined as a probability mapping from state–action pairs to states, which is governed by the environmental conditions. For attacks applied on the environment dynamics, an attacker may infer environment dynamics [164] or perturb a DRL system's environment dynamics to make an agent fail in a specific way [136], [137], [199], [201]. In the autonomous driving case, the attacker can change the material surface characteristics of the road, such that the policy trained in one environment will fail in the perturbed environment. In a robot control task, the attacker can change the robot's mass distribution, so that the robot may lose balance when executing its original policy, because it has not been trained in that case.

Then, we categorize these attacks based on what knowledge the attacker needs. Broadly, this breaks attacks down into the already recognized white-box attacks, where the attacker has full knowledge of the DRL system, and black-box attacks, where the attacker has less or no knowledge.

1) *White Box:* If the adversary attacks the DRL system with the capability of accessing the architecture, weight parameters of the policy and Q networks, and querying the network, we can call it a white-box attack. Clearly, the attacker can formulate an optimization framework for the white-box setting and derive the optimal adversarial perturbation [135], [197]. Moreover, via the theoretical analysis of the attack feasibility and cost, the adversary can further decrease the efficiency and stealth of the learning [136], [194]. However, this setting is inaccessible for the adversary in most practical scenarios.
2) *Black Box:* In general, the trained RL models are kept private to avoid easy attacks by certain secure access control mechanisms. Therefore, the attacker cannot fully acknowledge the weight parameters of the policy network and Q networks and may or may not have access to query the policy network. In this case, the attacker can train a surrogate policy to imitate the victim policy and then use a white-box method on the surrogate policy to generate a perturbation and apply that perturbation to the victim policy [136]. The finite-difference (FD) method [202] in attacking classification models can be utilized to estimate the gradient on the input observations and then perform gradient descent to generate perturbations on the input observations [136]. In this black-box setting, it becomes difficult for the adversary to perturb a DRL system and needs to estimate the victim's information with large computation costs.

Based on the adversary's objective, adversarial attacks are divided into two types: poisoning attacks and snooping attacks.

1) *Poisoning Attack:* In particular, for poisoning attacks, there are at least two dimensions to potential attacks against learning systems, namely untargeted attacks [135] and targeted (induction) attacks [192]. In untargeted attacks, attackers focus on the integrity and availability of the DRL system, i.e., minimizing the expected return (cumulative rewards). Specifically, the work [135] has shown that the existing adversarial example crafting techniques can be used to significantly degrade the test-time performance of trained policies. However, in terms of defensive mechanisms, the attacker may control time steps [198] or solve an optimization framework in a stealthy manner [197]. Another attack of this category aims at maliciously luring an agent to a designated state more than decreasing the cumulative rewards [192]. Via combining a generative model and a planning algorithm, the generative model predicts the future states, and the planning algorithm generates a preferred sequence of actions for luring the agent [191]. Similar to untargeted attacks, by solving an optimization framework in a stealthy manner [199], the attacker can easily succeed in teaching any target policy.
2) *Snooping Attack:* Different from poisoning attacks, the attacker only aims to eavesdrop on environment dynamics, the action, and reward signals being exchanged between the agent and the environment. If the adversary can train a surrogate DRL model that closely resembles the target agent [164], [165], the desired information can be estimated by this model. Furthermore, the adversary only needs to train a proxy model to maximize reward, and adversarial examples crafted to fool the proxy will also fool the agent [203]. We can note that the snooping attacks can still launch devastating attacks against the target agent by training proxy models on related tasks and leveraging the transferability of adversarial examples.

3) Taxonomy of Defenses: Defensive mechanisms found in multiple works are grouped by their underlying defensive strategies, as shown in Table 5. We will discuss various defenses in model sharing frameworks as follows.

a) *DP:* DP tackles the privacy leakage from a single data change in a dataset when some information from the dataset is publicly available and is widely used due to its strong theoretical guarantees [204]. Common DP mechanisms will add an independent random noise component to access data, i.e., the shared models in this level, to provide privacy. DP preserving distributed learning systems have been studied from various paradigms, such as distributed principal component analysis (PCA) [167], distributed ADMM [168], distributed SGD [126], FL [169], [170], and MARL [171], [172]. In order to provide fine-tuned control over the trade-off between the estimation accuracy and the privacy preservation, a distributed privacy-preserving sparse PCA (DPS-PCA) algorithm that generates a min–max optimal sparse PCA estimator under DP constraints has been proposed in [167]. Similarly, for distributed ADMM, distributed SGD, FL, and MARL systems, all related works focus on improving the utility-privacy trade-off via two aspects as follows: 1) analyzing the learning performance with a DP constraint and then optimizing system parameters and 2) enhancing the DP mechanism by obtaining tighter estimates of the overall privacy loss.
b) *Model Compression:* Model compression techniques for distributed SGD and FL systems, e.g., sketches, can achieve provable privacy benefits [114], [173]. Therefore, a novel sketch-based framework (DiffSketch) for distributed learning has been proposed, improving absolute test accuracy while offering a certain privacy guarantee and communication

Table 5 Taxonomy of Defenses in Level-1 Distributed ML With Sharing Models

Method	Ref.	Description	Key Challenges	Effectiveness
DP	[126], [167]–[172]	Introducing a level of uncertainty into the released model sufficient to mask the contribution of any individual user	Finding a balance between the training performance and privacy level	Low complexity in preserving privacy
Model compression	[114], [173]	Encoding local models before transferring them to the server	Measuring the effect on the privacy and reduce the negative effect on the training performance	Low complexity and high communication efficiency
HE	[116], [174]	Mathematical operations applied on an encrypted message result in the same mathematical operation being applied to the original message	Increasing computation complexity and transmission bits	Strongly effective in security
Secure MPC	[78]	Allowing two or more participants to jointly compute functions over their collective data without disclosing any sensitive information	Lack of a common protocol for various tasks	A lower complexity than HE and a higher security than DP
Statistical analysis	[175], [176]	Detecting and filtering the outliers based on the statistical information, e.g., Euclidean distance and principle component	Destroying the training performance, especially in the non-i.i.d. setting	Low complexity to detect outliers
Pretest on Auxiliary Datasets	[177], [178]	Calculating the accuracy score for all local models and reducing the effect of low-quality ones	Performance governed by the quality of auxiliary datasets	Directly detecting malicious users with sensitive datasets
Authentication	[179]	Using trust composition for determining the trust and reputation values for unknown agents	Relying on the trust transfer and vulnerable to the collusion	Low complexity in security
	[180]–[182]	Combining blockchain technology and reaching an agreement by a group of agents	Vulnerable to the 51% attack	Guaranteeing fairness in integrity
Authorization	[183]–[185]	Constructing capability-based access and different agent privilege levels	Formulating corresponding authorization standards for differential privilege levels	Guaranteeing the quality of participants

compression. Moreover, the work in [173] has presented a family of vector quantization schemes, termed vector-quantized SGD (VQSGD), and provides an asymptotic reduction in the communication cost and automatic privacy guarantees.

c) *Encryption:* Encryption, e.g., HE [116] and multiparty computation (MPC) [78], is also adopted to protect user data privacy through parameter exchange under the well-designed mechanism during ML. A novel DL system [116], bridging asynchronous SGD and cryptography, has been proposed to protect gradients over a honest-but-curious cloud server, using additively HE, where all gradients are encrypted and stored on the cloud server. To verify whether the cloud server is operating correctly, VerifyNet [174] has been proposed to guarantee the confidentiality of users' local gradients via a double-masking protocol in FL, where the cloud server is required to provide proof of the correctness of its aggregated results to each user.

d) *MPC:* The work in [78] has outlined an approach to advancing privacy-preserving ML by leveraging MPC to compute sums of model parameter updates from individual users' devices in a secure manner. The problem of computing a multiparty sum where no party reveals its updates to the aggregator is referred to as secure aggregation. Via encoding local models into multiple secret shares in the first round and then splitting each share into a public share and a private share, the work in [205] can provide stronger protections for the security and privacy of the training data. MPC integrates encryption technology and interactive protocols, aiming to make the receiver avoid sensitive information and obtain the necessary messages [206], [207], [208], [209].

e) *Statistical Analysis:* The work in [175] has proposed a robust aggregation rule, called adaptive federated averaging, that detects and discards bad or malicious local model updates based on a hidden Markov model. To tackle adversarial attacks in the FL aggregation process, the work in [176] presented a novel aggregation algorithm with the residual-based reweighting method, in which the weights for the average of all local models are estimated robustly. Via controlling the global model smoothness based on clipping and smoothing on model parameters, a samplewise robustness certification FL framework has been proposed, which can train certifiably robust FL models against backdoors [210]. Most of the defenses for FL aim to explore the latent model exception, such as similarities between malicious and benign clients, and then lessen the influence of this exceptional model [211], [212], [213], [214].

f) *Pretest on Auxiliary Datasets:* For detecting poisoned updates in collaborative learning [177], the results of client-side cross validation were applied for adjusting the weights of the updates when performing aggregation, where each update is evaluated over other clients' local data. The work in [177] considered the existence of unreliable participants and used the auxiliary validation data to compute a utility score for each participant to reduce the impact of these participants. The work in [178] has proposed a novel poisoning defense method in FL, in which a participant whose accuracy is lower than a predefined threshold will be identified as an attacker, and the corresponding model parameters will be removed from the training procedure in this iteration.

g) *Authentication and Access Control:* The key question in considering security in a MARL consists of increasing the confidence that all parties involved in the system (agents, platforms, and users) will behave correctly, and this can be achieved through the authentication of these parties. The identification of the parties can make up a system and possibly establish an agent-trust relationship. Thus, how to design efficient identity certification mechanisms to uniquely authenticate known and trusted users and agents in the system has drawn considerable attention. A domain-independent and reusable MARL infrastructure has been developed in [215], in which the system uses a certification authority (CA) and ensures full cooperation of secured agents and already existing (unsecured) agents. The work in [179] has introduced a method called trust composition, which combines several trust values from different agents. We can note that the trust composition can play a critical role in determining the trust and reputation values of unknown agents, since it is impractical for an agent to get complete knowledge about other agents. A work called personalized trust framework (PTF) has been proposed to establish a trust/reputation model for each application with personalized requirements [216]. Naturally, the idea of using blockchain technology to solve security problems in multirobot systems was discussed in [180]. The work in [180] stated that combining peer-to-peer networks with cryptographic algorithms allows reaching an agreement by a group of agents (with the following recording this agreement in a verifiable manner) without the need for a controlling authority. Thus, blockchain-based innovations can provide a breakthrough in MARL applications. The work in [181] has developed an approach to using decentralized programs based on smart contracts to create secure swarm coordination mechanisms, as well as for identifying and eliminating Byzantine swarm members through collective decision-making. The work in [182] has proposed an approach combining blockchain technology and explainability supporting the decision-making process of MARL, in which blockchain technology offers a decentralized authentication mechanism capable of ensuring trust and reputation management.

h) *Authorization and Trust Model:* Combined with authentication, authorization is used to restrict the actions that an agent can perform in a system and control the access to resources by these agents. Sensitive information about principals is transferred online even across the Internet and is stored in local and remote machines. Without appropriate protection mechanisms, a potential attacker can easily obtain information about principals without their consent. In the context of authorization mechanisms, the algorithm proposed in [183] is designed to solve the problem of systems that are constantly changing. The main goal is to build a flexible and adaptive security policy management capable of configuring itself and reflect the actual needs of the system. According to the authors, a system is not safe if a security model is developed but never managed afterward. Security of the proposed system in [184] has been further explored in the form of authorization and encryption of the data by introducing an authorization layer between the user and the system that will be responsible for providing access to the legitimate users of the system only. The work in [185] has ensured agent authorization and platform security with capability-based access and different agent privilege levels, in which the agent behavior is modeled with an activity transition graph (ATG) and implemented entirely in JavaScript with a restricted and encapsulated access to the platform API (AgentJS).

4) Real Examples for Level-1 Distributed ML:

a) *Electronic Medical Records (EMRs) [217]:* The use of information and network technologies in the healthcare field inevitably produces EMR, which is a necessary trend for the modernization of medical records in hospitals. The initial adoption of EMR in clinical practice has vastly improved the efficiency and quality of health care provided by hospitals. Empowered by algorithm technologies and data reconstruction, BaseBit [217] has constructed a robust and comprehensive knowledge base system and has a series of intelligent models with excellent abilities of expression. In various applications centered around EMRs, the proposed models effectively improve the abilities, such as automatic medical record writing, overall quality control, cost monitoring systems for single diseases, early warning for infectious diseases, prompt for critical illnesses, clinical decision-making assistance for rare diseases, and enabling hierarchical diagnosis and treatments.

5) Brief Summary: As shown in Fig. 9, although, due to the local training process, the raw data of each participant will not be exposed to the curious server or

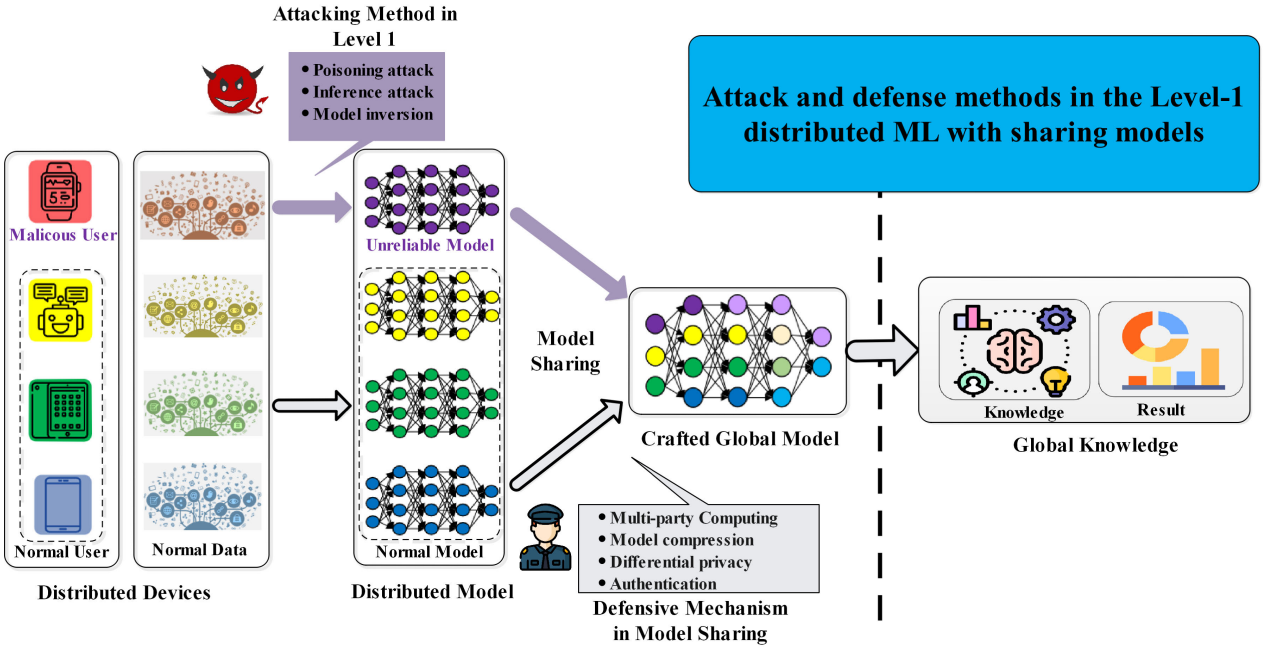


Fig. 9. ***Breakout figure from Fig. 3: an illustration of privacy and security issues in Level-1 distributed learning with model sharing.***

external attackers, defensive mechanisms are also necessary because of the existing possibility of feature inference and data reconstruction from model sharing, in addition to the model poisoning paradigm. Traditional HE and DP have been proven beneficial to privacy preservation but lead to low efficiency or reduced utility. Therefore, the quantitative analysis of the relationship between the sensitive feature and the published model is imperative.

C. Level 2: Sharing Knowledge

Recent configurations that rely on knowledge sharing techniques can be summarized as split learning [47], vertical FL [9], and distillation-based FL [221]. Split learning allows multiple clients to hold different modalities of vertically partitioned data and learn partial models up to a certain layer (the so-called cut layer). Then, the outputs at the cut layer from all clients are then concatenated and sent to the server that trains the rest of the model. In vertical FL, participants hold the same set of samples but with disjoint features, and only one participant owns the labels, which need to combine split NNs and privacy-preserving techniques [222]. Distillation-based FL [46], [221], [223] exchanges model outputs instead of model parameters, where the communication overhead cannot scale up according to the model size and has been proven to satisfy the DP guarantee.

1) Threat Models: In knowledge sharing paradigms, adversarial participants or eavesdroppers still possibly exist. The adversarial participants can be categorized into two kinds: 1) honest-but-curious (semi-honest) participants, who do not deviate from the defined learning protocol, but attempt to infer private training data from the legitimately received information and 2) malicious participants, who may deviate from the defined learning protocol and destroy this training task or inject Trojans to the training model.

2) Taxonomy of Attacks: The existing attacks on knowledge sharing paradigms can be mainly categorized as label leakage, feature inference, and data reconstruction, as shown in Table 6. We discuss the existing attacks as follows.

a) *Label Leakage:* The labels in distributed learning frameworks might be highly sensitive, e.g., whether a person has a certain kind of disease. However, the bottom model structure and the gradient update mechanism of vertical federated learning (VFL) or split learning can be exploited by a malicious participant to gain the power to infer the privately owned labels [224]. Worse still, by abusing the bottom model, he/she can even infer labels beyond the training dataset [225]. The work in [218] first made an attempt at a norm attack that uses the norm of the communicated gradients between the parties, and it can largely reveal the ground-truth labels from participants. The adversary (either clients or servers) can accurately retrieve the private labels by collecting the exchanged gradients and smashed data [226]. Thus, it is necessary to make gradients from samples with different labels similar.

b) *Feature Inference:* Through analysis, the work in [227] and [228] demonstrated that, unless the feature dimension is exceedingly large, it remains feasible, both theoretically and practically, to launch a reconstruction attack with an efficient

Table 6 Taxonomy of Attacks in Level-2 Distributed ML With Knowledge Sharing

Method	Ref.	Attacker's knowledge	Learning Model	Effectiveness
Label leakage	[218]	Black box	Split learning	Revealing the ground-truth labels from the participants
Feature inference	[190]	Black box	Vertical FL	Inferring the feature values of new samples belong to the passive parties successfully
Data reconstruction	[219]	Black box	Split learning	Activated output after two and three convolutional layers can be used to reconstruct the raw data
	[220]	Black box	Vertical FL	Stealing partial raw training data successfully

search-based algorithm that prevails over current feature protection techniques. In this article, the authors have performed the first systematic study of relation inference attacks to reveal VFL's risk of leaking samples' relations. Specifically, the adversary is assumed to be a semi-honest participant. Then, according to the adversary's knowledge level, the work [228] formulated three kinds of attacks based on different intermediate representations and revealed VFL's risk of leaking samples' relations. Luo et al. [190] considered the most stringent setting that the active party (i.e., the adversary) only controls the trained vertical FL model and the model predictions and then observed that those model predictions will leak a lot of information about features by learning the correlations between the adversary's and the attacking target's features.

c) *Data Reconstruction:* The work in [219] has provided the leakage analysis framework via three empirical and numerical metrics (distance correlation and dynamic time warping), indicating that the activated outputs after two or more convolutional layers can be used to reconstruct the raw data, i.e., sharing the intermediate activation from these layers may result in severe privacy leakage. In vertical FL, two simple yet effective attacks, the reverse multiplication attack and reverse sum attack, have been proposed to steal the raw training data of the target participant [220]. Though not completely equivalent to the raw data, these stolen partial orders can be further used to train an alternative model, which is as effective as the one trained on the raw data [229].

3) Taxonomy of Defenses: Defensive mechanisms found in multiple works of the literature are grouped by their underlying defensive strategy, as shown in Table 7. Hence, we will discuss various defenses in model sharing frameworks as follows.

a) *DP:* The work in [230] has proposed a privacy-preserving protocol for composing a differentially private aggregate classifier using local classifiers from different parties. In order to overcome the effects of the proposed information inference attacks [219], DP has been proven helpful in

Table 7 Taxonomy of Defenses in Level-2 Distributed ML With Knowledge Sharing

Method	Ref.	Use case	Key idea	Effectiveness
DP	[232]	Deriving aggregate information without revealing information about individual data instances	Differentially private aggregate in a multi-party setting	DP analysis on the perturbed aggregate classifier
	[221]	Against DCM and DTWM attacks in split learning	Laplace mechanism on the split layer activation	Strong DP level ($\epsilon = 1$) works but degrading the classification accuracy
MPC	[233]	Vertical decision tree training, random forest (RF), and gradient boosting decision tree (GBDT)	A hybrid framework of threshold partially HE (TPHE) and MPC	Be independent of any trusted third party against a semi-honest adversary that may compromise $m-1$ out of m clients
Encryption	[234]	Asymmetrically split learning	Partial HE (PHE), additive noise	Achieving a lossless performance and more than 100 times speedup
	[234]	Vertical tree-boosting system	HE	Revealing no information of each participant and achieving a lossless performance
Secure aggregation	[235]	Vertical GBDT	Lightweight secure aggregation because the whole training relies on the order of the data instead of the values	Achieving the same level of the area under the ROC curve (AUC) with centralized training
Others	[236]	Privacy attributes inferring from extracted features	Adversarial training and neural network based mutual information estimator	First task-independent privacy-respecting data crowdsourcing framework
	[221]	Against DCM and DTWM attacks in split learning	Adding more hidden layers	Preventing privacy leakage with a slight reduction in performance
	[220]	Against norm-based attack	Adding Gaussian noise by making the expected norm of the positive and negative gradients in a mini-batch equal	Preventing label leakage against some extreme scenarios.

reducing privacy leakage but leading to a significant drop in model accuracy.

b) *MPC:* The work in [231] has proposed a novel solution for privacy-preserving vertical decision tree training and prediction, termed Pivot, ensuring that no intermediate information is disclosed other than necessary releases (i.e., the final tree model and the prediction output).

c) *Encryption:* A novel privacy-preserving architecture has been proposed in [232], which can collaboratively train a DL model efficiently while preserving the privacy of each party's data via the HE technique. The work in [232] has explored a lossless privacy-preserving tree-boosting system known as SecureBoost by using the additive HE scheme.

d) *Secure Aggregation:* The work in [233] has proposed the vertical FederBoost, which runs the gradient boosting decision tree (GBDT) training algorithm in exactly the same way as centralized learning. Via further utilizing packetization and DP, this algorithm can protect the order of samples: participants partition the sorted samples of a feature into buckets, which only reveals the order of the buckets and add differentially private noise to each bucket.

e) *Others:* The work in [234] has presented task-independent privacy-respecting data crowdsourcing (TIPRDC) to learn a feature extractor that can hide the private information from the intermediate representations using an adversarial training process while maximally retaining the original information embedded in the raw data to accomplish unknown learning tasks. In [219], adding more hidden layers to the client side was proven helpful in reducing privacy leakage, but increasing the number of layers seems ineffective with the most highly correlated channels. In order to relieve the negative impact of random perturbation preserving techniques on the learned model's predictive performance, the work in [218] has introduced an improved way to add Gaussian noise by making the expected norm of the positive and negative gradients in a mini-batch equal (undistinguishable).

4) Real Examples for Level-2 Distributed ML:

a) *Federated AI Technology Enabler (FATE):* An open-source project, named FATE, provides a secure computing framework to support the federated AI ecosystem [235], led by WeBank's AI Department. It can enable big data collaboration without privacy leakage by implementing multiple secure computation protocols, such as DP, HE, and so on. FATE accesses out-of-box usability and excellent operational performance with a modular modeling pipeline, explicit visual interface, and flexible scheduling system [236]. eHi Car Services, a national chain car rental brand, and WeBank jointly announced a deep strategic partnership, announcing that the two sides will carry out multiscene and multidimensional innovation cooperation in car travel, member services, finance and insurance, blockchain technology, and other fields. eHi Car Services uses federal transfer learning, AI face authentication technology, payment technology, and other fintech to deeply integrate into the car rental service process for the purpose of optimizing and improving user experience and combines the car rental scene with the bank's big data risk control system, so as to provide new travel services.

5) Brief Summary: As shown in Fig. 10, split learning, vertical FL, and distillation-based FL are classical knowledge sharing systems, in which the knowledge can be viewed as the partial processing result to meet the requirement of the system learning. It is also challenging for knowledge sharing systems to hide sensitive information when there is shared knowledge.

D. Level 3: Sharing Results

We define the sharing results category as follows: there is no interaction or communication during the process of training. The distributed clients only share the training results after the process ends. The history of sharing results can be traced back to ensemble ML over partitioned datasets [237], [238], where a number of base classifiers collectively determine the output for an instance based on a predefined aggregation strategy. Ensemble techniques were originally introduced to increase the overall performance of the final classification, but it is also straightforward to utilize them for distributed ML systems [13]. The shared results [239] in distributed learning can be either the final training models, e.g., private aggregation of teacher ensembles (PATE) and multiagent multiarm bandits (MAMABs), or the prediction (output) of the models, e.g., crowdsourcing.

1) Threat Models: For the result sharing models, malicious participants may exist and provide false advice or results to hinder the learning performance of other participants or the global model. In addition, curious participants can infer some confidential information from the shared results.

2) Taxonomy of Attacks: As stated by Silva et al. [240], the existence of malicious participants is a key concern in agent advice. The work in [241] has proposed the attack model that some of these agents might become self-interested and try to maximize car owners' utility by sending out false information. Based on [241], Hayes et al. [242] have investigated attacks in the setting where the adversary is only permitted to access the shared results (such as the generated samples set in GAN), by retraining a local copy of the victim model. In addition, Hilprecht et al. [243] have proposed to count the number of generated samples that are inside an ϵ-ball of the query, based on an elaborate design of distance metric.

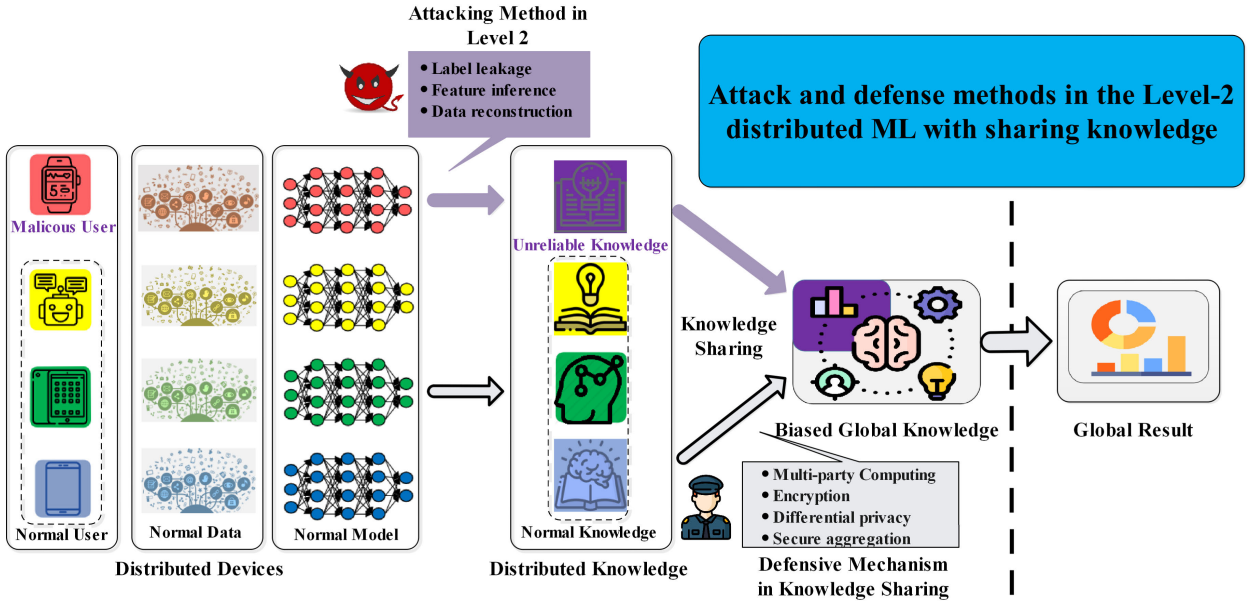


Fig. 10. *Breakout figure from Fig. 3: an illustration of privacy and security issues in Level-2 distributed learning with knowledge sharing.*

The work in [244] has presented the first taxonomy of MIAs and focused on MIA against deep generative models that reveals information about the training data used for victim models. In the spirit of Hilprecht et al. [243], this work scored each query by the reconstruction error directly, which does not introduce additional hyperparameters while achieving superior performance. We further summarize these attacks in Table 8.

3) Taxonomy of Defenses: In results sharing paradigms, Table 9 summarizes the use cases, key ideas, and effectiveness for the existing attacks. Moreover, we will discuss various defenses in model sharing frameworks as follows.

a) *DP:* The work in [172] has proposed a novel differentially private agent advising approach, which employs the Laplace mechanism to add noise to the rewards used by student agents to select teacher agents. By using the advising approach and the DP technique, this approach can reduce the impact of malicious agents without identifying them and naturally control communication overhead. The work in [245] adopted DP and studied regret upper and lower bounds for MAB algorithms with a given LDP guarantee. The differentially private PATE framework has been proposed to achieve individual privacy guarantees with provable privacy bounds [247], [248].

b) *MPC:* Zhao [246] has proposed to use the teacher–student framework in a more general distributed learning setting. The goal of this work is to address distributed DL under DP using the teacher–student paradigm. In the setting, there are a number of distributed entities and one aggregator. Each distributed entity leverages DL to train a teacher network on sensitive and labeled training data. The knowledge of the teacher networks is transferred to the student network at the aggregator in a privacy-preserving manner that protects the sensitive data. This transfer results from training nonsensitive and unlabeled data, which also applies secure MPC to securely combine the outputs of local ML for updating.

c) *Others:* If an ensemble contains enough models, and each model is trained with disjoint subsets of the training data in a distributed manner, then "any predictions made by most of the models should not be based on any particular part of the training data" [249]. The PATE is based on this idea [10]. In more detail, the ensemble is seen as a set of "teachers" for a new "student" model. The student is linked to the teachers only by their prediction capabilities and is trained by "querying the teachers about unlabeled examples." The prediction result is disjointed from the training data through this process. Therefore, data privacy can be protected. The privacy budget for PATE is much lower than traditional DP-based ML approaches. But, it may not work in many practical scenarios, as it relies on an unlabeled public dataset.

4) Real Examples for Level-3 Distributed ML:

a) Large-scale online taxicab platforms, such as Uber and DiDi, have revolutionized the way people travel and socialize in cities worldwide and are increasingly becoming essential components of the modern transit infrastructure [250], [251]. An RL-based dynamic bipartite graph matching approach has been adopted to assign each worker with one or more tasks to

Table 8 Taxonomy of Attacks in Level-3 Distributed ML With Results Sharing

Method	Ref.	Attacker's knowledge	Learning Model	Effectiveness
Poisoning attack	[241]	Black box	Street random waypoint (STRAW) mobility	Average speed of vehicles in the network decreases as the percentage of liars increase
Inference attack	[242]	White-box, black-box	GAN	Achieving 100% and 80% successful at membership inferring in white-box and black-box settings, respectively
	[243]	Black-box	GAN, variational autoencoders (VAEs)	Success rates superior to previous work with mild assumptions
	[244]	White-box, partial black-box, black-box	GAN	Consistently outperforms the state-of-the-art models with an increasing number of generated samples

maximize the overall revenue of the platform, where the workers are dynamic, while the tasks arrive sequentially. Specifically, for each worker–task pair, the platform can obtain a reward based on value-based RL. Then, via some solutions to bipartite graph matching, such as greedy search, the platform can make near-optimal decisions. However, if the platform can obtain all workers' information and its purpose is only aiming to maximize the overall revenue, workers may be out of control. Thus, using DP to achieve fairness may be a solution [252].

5) Brief Summary: As shown in Fig. 11, although the results from ML systems are different from the raw data, there are also the existing risks of privacy leakage, such as the generated samples from the generator in GAN. Hence, several defensive mechanisms are utilized for preventing privacy leakage and against malicious participants.

E. Relationship Among the Privacy and Security Issues in the Four Levels of Distributed ML

From Level 0 to Level 3, there is no specific rule for the privacy and security level, but we may conclude that the forms of data expose different degrees of information in the considered four levels. For example, compared with the prediction results in Level 3, much more information can be extracted from the raw or original data in Level 0. Regarding the protection methods, designing a general mechanism for the four levels is a nontrivial task. For example, the DP-based mechanisms can be well adopted in Level 0 (i.e., LDP [130], [151]), Level 1 (i.e., DP in DL [126]), and Level 3 (i.e., PATE-GAN [10]), but they may lose the effectiveness in Level 2 (sharing knowledge).

VI. LESSONS LEARNED

In this section, we summarize the key lessons learned from this survey, which provides an overall view of the current research on security and privacy issues in distributed learning.

A. Lessons Learned From Definitions of Security and Privacy

The public often conflates the terminologies of "privacy" and "security," which are, in fact, distinctively different. From the expression of privacy and security in distributed learning, we can learn lessons as follows.

1) Difference Between Security and Privacy: The concerns of security and privacy issues are different [253], [254], [255]. On the one hand, security issues refer to unauthorized/malicious access, change, or denial of data or learning models. Such attacks are usually launched by adversaries with expert/full knowledge of the target system. Hence, the fundamental three goals of security are confidentiality, integrity, and availability [163]. On the other hand, privacy issues generally refer to the unintentional disclosure of personal information. For example, from a side-by-side comparison of a vote registration

Table 9 Taxonomy of Defenses in Level-3 Distributed ML With Results Sharing

Method	Ref.	Use case	Key idea	Effectiveness
DP	[172]	Malicious agent advising	Laplace mechanism	Reducing the impact of malicious agents without identifying them
	[245]	Against inference attacks from any party or eavesdropper	Laplace mechanism, Bernoulli mechanism	Providing regret upper and lower bounds for MAB with local DP
MPC	[246]	PATE	Training non-sensitive and unlabeled data, Securely combining the outputs by MPC	Guaranteeing data security
Others	[10]	PATE	The student is linked to the teachers only by their prediction capabilities and trained by "querying the teachers about unlabelled examples"	Achieving much lower privacy budget than traditional DP approaches

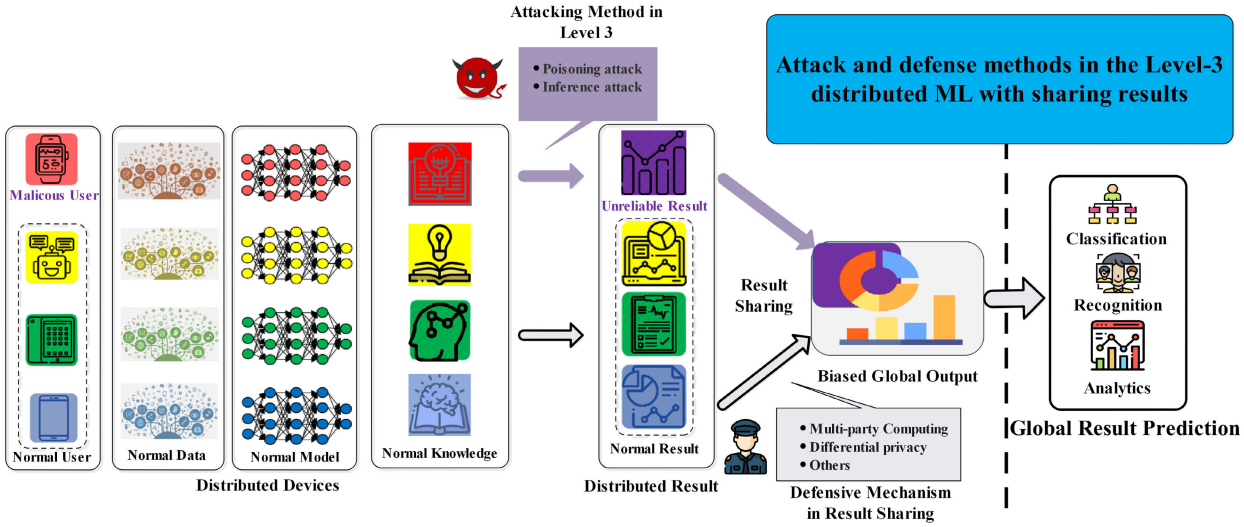


Fig. 11. ***Breakout figure from Fig. 3: an illustration of privacy and security issues in Level-3 distributed learning with results sharing.***

dataset and an anonymous set of health-care sensor records (e.g., no individual name and ID), an adversary may have the ability to identify particular individuals and the health conditions of these individuals leaks [68], [190], [256]. This is because attributes, such as gender, birth date, and zip code, are the same in both datasets.

2) Connection Between Security and Privacy: Security and privacy go hand in hand. Privacy issues can further induce security issues in some scenarios. If an adversary steals the private information of individuals, substantial profit from the information can be easily obtained. For example, when the adversary extracts the health conditions of an important person, he/she can blackmail the victim person by threatening to reveal the information. We know that one can envision an environment that is secure but does not guarantee privacy. Similarly, one can imagine an environment that is private, but it does not guarantee security from outsiders. Security can be achieved without privacy, but privacy cannot be achieved without security. This is because whether the security is weak or vulnerable, it will automatically affect privacy.

B. Lessons Learned From Evaluations of Security and Privacy

The evaluations of security and privacy guide the research directions in this area. In the following, we will provide some lessons by reviewing the state of the art.

1) Bayes-Based Methods: Privacy leakage can be formalized as a Bayes optimization problem from the aspect of an adversary with different assumptions on the probability distributions of the input data and interactive messages (such as gradients and extracted features). For example, the work in [257] constructed a theoretical framework that can measure the expected risk that an adversary has in the process of reconstructing an input, given the joint probability distribution of inputs and their gradients. This framework can reveal the gradient leakage level by analyzing the Bayes optimal adversary, which minimizes this risk with a specific optimization problem involving the joint distribution. DP constitutes a strong standard for privacy guarantees for algorithms on aggregate databases [123], [126], [170]. It is defined in terms of the application-specific concept of adjacent databases and aims to hide whether one sample exists in the database. Thus, DP is defined as the detecting probability of outputs of any two adjacent databases.

2) Experiment-Based Methods: Attack algorithms can evaluate the security and privacy levels directly. In order to evaluate the adversarial robustness of image classification tasks, large-scale experiments have been conducted, and the performance of different defense methods can be evaluated [258]. In addition, we can apply adversaries to DP-SGD, which allows for evaluating the gap between the private information that an attacker leaks (a lower bound) and what the privacy analysis establishes as being the maximum leak (an upper bound) [129]. We can notice that attack methods constantly emerge to face advanced defense methods. Thus, the experiment-based methods need to consume a lot of computation resources, such as 3000 GPU hours with parallelized over 24 GPUs, as shown in [129].

C. Lessons Learned From Attacks and Defenses

The research on attacks and defenses in distributed learning is faced with an "arms race," i.e., a defense method proposed to prevent the existing attacks will be soon evaded by new attacks and vice versa.

1) Attacks in Distributed Learning: Attack algorithms in the white-box scenario have attracted considerable of attention in the last few decades, but they seem to be

impractical and can only be used as an upper bound. For example, model poisoning attacks in FL can be divided into three scenarios based on various levels of background knowledge, i.e., full knowledge, partial knowledge, and no knowledge. The attack performance decreases drastically, as the background knowledge decreases [163], [259], [260]. In this context, practical attack algorithms with no knowledge should be studied to explore potential privacy and security risks. In addition, the organizer usually obtains more background knowledge than the rest of the participants. In order to mitigate the risk of the organizer being an adversary/eavesdropper, the decentralized framework can be adopted as a solution.

For the same attack purpose, different levels of distributed learning require different background knowledge, since the level of distributed learning determines interactive messages, which usually contain the private information of participants, such as extracted features and NN gradients of private data. Thus, various attack methods have emerged to infer private information or poison training processes instead of unified attack schemes. For example, MIA in Level 1 (sharing model) needs shadow datasets to train shadow models and then estimates the confidence of the training models [54]. We know that the shadow datasets and their distribution affect the attack performance obviously. However, how to obtain the shadow datasets becomes controversial, such as generative networks, stealing, and so on.

2) Defenses in Distributed Learning: Although distributed learning can achieve privacy-enhanced and scalable data sharing, it also presents some security and privacy risks. Four-level distributed learning frameworks show various risk levels of privacy leakage, due to the different interactive messages [261], [262]. The interactive messages usually contain the private information of participant users, such as extracted features and NN gradients of private data. This data processing can protect private data to some degree. Thus, it is of interest to study the potential privacy protection levels owing to these data processing functions and then design effective protection schemes to achieve a better trade-off between training performance and privacy.

Privacy/confidential computing for distributed learning is a high requirement compared with conventional privacy protection. However, the existing privacy computing techniques usually cannot provide systematic privacy preservation, which will degrade the learning performance or training efficiency [154], [263]. In addition, the protection effectiveness of different privacy computing techniques varies. For example, DP is seen as an effective method to prevent MIAs by perturbing the impact on whether one instance exists in the training process. Thus, the sensitivity of interactive messages in distributed learning for DP should be carefully investigated when estimating the privacy budget. MPC is another widely used privacy computing technique. However, the transfer ability of MPC is limited, and the MPC protocols for different paradigms of distributed learning need to be well designed. Overall, it is crucial to combine these privacy computing techniques and design a general privacy-preserving framework for different paradigms of distributed learning [264], [265].

D. Lessons Learned From FL

Reviewing the state of the art in the field, we find that FL plays an increasingly important role in facilitating training ML models for distributed data, as highlighted as follows.

1) Advantages of FL: Three classic paradigms in FL, i.e., horizontal FL, vertical FL, and federated transfer learning, can be categorized as Levels 1–3 of distributed learning and have the capability to address most of the challenges of training ML models in distributed scenarios. FL is an efficient approach for federated data sharing among multiple clients, in which raw data are kept on the client side, which, in turn, protects data privacy from tensor mining. The primary purpose of FL is to train a satisfied ML model without exposing participants' data privacy. Thus, when we select or design a training framework, both participants' data characteristics and privacy requirements should be considered. In addition, an increasing number of advanced paradigms have emerged to handle various challenges in FL training, such as multimodal FL [266], [267], [268], federated knowledge distillation [269], [270], [271], quantized FL [117], and so on, which help to construct a secure and efficient federated AI ecosystem.

2) Disadvantages of FL: FL can benefit data privacy, security, and privacy risks induced by the interactive messages. In particular, FL can be combined with other privacy techniques, such as DP, MPC, HE, and so on, to improve the privacy of local updates, by integrating them into gradient descent training to enable privacy-enhancing FL. Moreover, the security of FL-based data sharing can be improved by combining it with blockchain technology [272], [273], [274], [275]. In this context, the information of trained parameters can be appended into immutable blocks on a blockchain during client–server communications. Furthermore, the substantial communication cost in vertical FL should be noticed [276], [277], [278]. Specifically, in vertical FL, the total computation and communication cost is proportional to the training dataset size. In other words, the widely adopted batch computation method in horizontal FL cannot be applied to vertical FL. When facing a massive amount of data, e.g., billions of advertising data, communication and local computation may be in many orders of magnitude, and the system may lose vitality due to limited resources, such as hardware capacity, bandwidth, and power.

VII. RESEARCH CHALLENGES AND FUTURE DIRECTIONS

As discussed in Sections V and VI, distributed learning systems can alleviate security and privacy concerns by advancing defense mechanisms. In Section VII, we provide and reveal several critical research challenges for

Table 10 Summary of Challenges Along With Their Descriptions and Possible Solutions

Challenges	Description	Solution
Balance between ML performance and Security/Privacy Level	The tradeoff between the learning performance, such as convergence, and the privacy and security level should be well designed.	Dynamic parameter optimization Specific/personalized protection mechanism
Decentralized Paradigm	In the distributed fashion, the regulations as well as the incentives among multiple participants should be investigated.	Authentication and access control Consensus design Blockchain assisted distributed learning
Complexity Reduction	Distributed learning with high complexity security and privacy protection is sometimes impractical. How to alleviate this complexity burden under a required protection level still needs investigation.	Lightweight encryption High-efficiency secure protocols Model compression

further improvement in system implementation. In addition, related possible solutions are also discussed, and a summary is provided in Table 10.

A. Balance Between ML Performance and Security/Privacy Level

1) Convergence Analysis: As mentioned above, DP has been widely adopted in training of distributed ML models, by adding random noise to gradients during the training process. However, a strict privacy guarantee usually requires a large noise variance, so the DP-based training will lead to significant performance degradation. Although the existing works in [279] and [170] have explored the training performance of the differentially private distributed learning systems and provided some theoretical results, these results can only bring out some intuitions and cannot enhance the learning performance directly. Therefore, an accurate estimation of convergence performance of differentially private ML training would be beneficial to find a proper balance between utility and privacy.

2) Dynamic Parameter Optimization: In addition to the accurate estimation of convergence performance, dynamic parameter optimization is also a promising direction to balance the trade-off between utility and privacy. Because of privacy protection, the training performance caused by the original parameters has been changed. Correspondingly, the conventional parameter optimization method for distributed ML also becomes inapplicable. For example, the work in [170] has developed an upper bound on differntially private FL and revealed that there exists an optimal number of communication rounds with a given privacy level. This discovery brings a new perspective on the communication round in FL and suggests a rethinking of the choice of communication parameters. Dynamic parameter optimization for differentially private ML has also been considered, which implements a dynamic privacy budget allocator over the course of training to improve model accuracy [280]. Although the existing dynamic optimization methods have already been proposed and proven to improve a number of distributed learning systems, there is still considerable room for improvement.

3) Specific/Personalized Protection Mechanisms: The various requirements for different scenarios or different participants in distributed ML systems are also challenging, especially when the data distribution is non-IID [281], [282]. Therefore, designing a specific/personal protection mechanism for a distributed ML system can bring out a better balance between utility and privacy. The work in [283] has considered a social network and achieved a proven DP requirement by perturbing each participant's option with a designated probability in each round. Combining sketch and DP techniques, the work in [114] has proposed a novel sketch-based framework, which compresses the transmitted messages via sketches to simultaneously achieve communication efficiency and provable privacy benefits. These designs can obtain a satisfactory trade-off between utility and privacy because of the deep combination of original scenarios and DP techniques. Therefore, how to balance utility and privacy in the amount of distributed learning scenarios has not been fully explored.

4) Private Set Intersection: Private set intersection (PSI) is an important step in distributed learning because of the individual differences among multiple users. For example, in horizontal FL/SGD systems, we need to ensure that each record has the same features. Existing PSI protocols are third party-based PSI [284], [285], public-key-based PSI [286], [287], circuit-based PSI [288], and oblivious transfer (OT)-based PSI [289]. However, there is still a research gap in terms of using PSI in distributed learning to investigate the trade-off between the privacy level and the learning performance.

B. Decentralized Paradigm

1) Authentication and Access Control: The key question in adding security to a decentralized paradigm is to increase the confidence that all parties involved in the system (agents, platforms, and users) will behave correctly, and this can be achieved by authentication. The identification of the parties establish a trusting environment between clients. Cryptology has been proven useful in a large number of authentication and access control scenarios, but it cannot address the problem of fully new participants. In addition, a trust/reputation model has been proposed to determine the participating values for unknown clients, since it is hard for an agent to obtain complete knowledge about other participants [179], [215], [216]. Consequently, how to design efficient identity certification

mechanisms to uniquely authenticate known, and trusted users and agents in the system has drawn much attention.

2) Consensus Design: Coordination and cooperative control of multiple clients in distributed ML has attracted considerable attention from various research communities, where a fundamental approach to achieving cooperative control is the consensus-based algorithm [290]. Traditional consensus designs are mostly based on a single and finite-time domain [291], [292], where, in reality, the dynamics of the system are usually complicated and nonlinear. Therefore, a useful and effective consensus design with dynamic or unknown parameters is an important topic for future research. For example, the time-varying resources and requirements for participating clients are key and untrivial factors in design. In addition, the security of consensus has also raised several issues recently [293]. How to protect the integrity of the consensus from inside or outside attackers and how to prevent private information leakage from the published consensus are other interesting research directions.

3) Blockchain-Assisted Distributed Learning: The reasons for implementing blockchain in a distributed learning system are to increase the interaction efficiency between participants by providing more trusted information exchange, reaching a consensus in trust conditions, assessing participant productivity or detecting performance problems, identifying intruders, allocating plans and tasks, and deploying distributed solutions and joint missions [294], [295]. However, the challenges consist of assessing feasibility and finding an architectural approach for combining blockchain-based consensus algorithms with real-time distributed learning systems, while assuring incentive information exchange and compatibility with the already existent local processing protocols [255]. In addition, the incentive mechanism is also key to consensus design [296], [297].

4) Fairness: Fairness has attracted increasing attention in recent years, especially in the scenario where multiple participants are evolved in one learning task [298]. A max–min fairness distributed learning system has been developed in [299], where multiple clients are matched with bandits having minimum regret. Furthermore, collaborative fairness in FL has been investigated in [300]. Although several works throw out the idea of fairness, there is a lack of a common definition of fairness in distributed learning. Whether attending the same rounds of training or allocating training trials according to the users' capability represents fairness is still an unclear question. In addition, the relationship between fairness with security and privacy also requires further consideration.

C. Complexity Reduction

1) Lightweight Encryption: One of the oldest and most popular techniques used in information security is cryptography, and its use to protect valuable information is usually relies on symmetric encryption and decryption algorithms, such as elliptic curve cryptography (ECC), homomorphic hash functions, or secret sharing technology. A secure lightweight ECC-based protocol, i.e., broadcast-based secure mobile agent protocol (BROSMAP) [301], has been improved to fulfill the needs of multiagent-based IoT systems in general and obtained better performance than its predecessor with the same security requirements. An HE-assisted MPC framework [174], enabling a participant to compute functions on values while keeping the values hidden, can allow certain mathematical operations (such as aggregation) to be performed directly on ciphertexts, without prior decryption. However, cryptographic algorithms usually require complicated computation protocols and may not be achieved efficiently.

2) High-Efficiency Secure Protocols: Secure protocols are designed to enable computation over data distributed between different parties, so that only the result of the computation is revealed to the participants, but no other private information. Secure protocols usually combine several efficient security and privacy techniques, e.g., MPC, DP, and HE, and need several interactions to exchange intermediate results. However, too many interactions may increase the information leakage risk, communication, and computing overhead. Moreover, it is also challenging to explore generic secure protocols over remote parties, especially for complicated scenarios and various applications. To realize an efficient communication protocol in a trusted and secure environment, an alternative way is to increase the transmission rate using an intelligent reflecting surface (IRS) by smartly reconfiguring the wireless propagation environment, with the help of massive numbers of low-cost passive reflecting elements integrated on a planar surface [302].

3) Model Compression: High accuracy of large NNs is often achieved by paying the cost of considerable memory consumption and complex computational capability, which greatly impedes the deployment and development in distributed systems [303]. To efficiently accelerate the learning process, privacy preservation-based methods, such as compact models [304], [305], tensor decomposition [306], data quantization [307], and network sparsification [308], are recent key advances.

D. Distributed ML and Futuristic Technologies

1) Robotics: Distributed ML can enhance the ability to identify and control robotics with remote and distributed control or wireless connections to clouds. This scenario requires high precision control, which raises increasing security issues and vulnerability to transmission errors [309], [310]. How to preserve the integrity of such control systems and how to prevent information leakage during data transmission need further investigation. In addition, ethical issues related to bionic robots are hotly debated concerns [311], [312].

2) Virtual Reality and Augmented Reality: ML and its distributed versions can improve the quality of generated images and videos, such as GAN and diffusion models. With the rapid development in virtual reality (VR)- and augmented reality (AR)-based applications, private information from generated videos may lead to personal information leakage [313], [314]. Adversaries can take advantage of fake videos to analyze the unique behaviors, personal interests, and background environments of participants [315].

3) Distributed Quantum Computing: Quantum ML operates based on quantum mechanics, taking advantage of superposition to store and process information [316], [317]. However, if information sources are from distributed clients, information leakage and inside or outside attacks may occur during data transmission. Thus, conducting the protection on distributed ML raises several challenging problems, such as identifying attackers, ensuring the integrity and availability of transmission data, and preserving privacy.

4) Metaverse: Metaverse seamlessly integrates the real world with the virtual one. It allows avatars to carry out rich activities, including creation, display, entertainment, social networking, and trading. Thus, it is a promising technology for building a exciting digital world and better physical scenario by exploring the metaverse [318], [319]. Intuitively, the breakthroughs of AI in the real world motivate people to realize the metaverse. For example, distributed ML via integrating distributed data from metaverse users can provide technical support for metaverse systems to reach or exceed the level of human learning. This can significantly affect the operational efficiency and the intelligence of the metaverse. Intelligent voice services provide technical support, such as voice recognition and communication. However, several new security and privacy challenges that can compromise the systems or divulge users' privacy raise attention in the interaction process, such as the communication between metaverse users and service providers.

5) Digital Twin: The digital twin can fill the gap between physical systems and digital spaces. Leveraging FL to construct digital twin models of IoT devices based on their running data has been proposed in [320] and [321]. The physical security of IoT devices is critical, as they can be damaged, destroyed, or even stolen by attackers. Digital twin systems also have other priorities than the traditional network/system security requirements because of their interactions with the physical components. For instance, defects in a critical product may lead to death, injuries, or environmental damage. For this reason, safety could be ranked as the top security requirement. Safety can broadly be defined as the avoidance of harm or hazard to the physical environment and infrastructure that could occur from system faults [322]. Meanwhile, the possible privacy leakage from the interactions with the physical components must also be considered.

6) Web 3.0: Web 3.0 has attracted considerable attention due to its unique decentralized characteristics [323]. In Web 3.0, data present a distributed storage structure, so there will be no central node for data management, significantly reducing the service cost of managing data. Web 3.0 emphasizes the protection of users' personal data, and therefore, as a key technology to solve the data privacy problem, privacy computing is becoming the immediate need of Web 3.0. Privacy computing technology can analyze and calculate data under the premise of protecting data privacy and security, which provides a strong guarantee for the efficient and safe circulation of data across industries and organizations.

7) Generative Design AI: Generative design uses AI to come up with multiple design variations for products or parts. This leads to a faster generation of design options than would be developed through manual design, which leads to faster product development times and more creative choices to select from. For example, the meteoric rise of diffusion models has been one of the most significant developments in ML in the past several years [324]. Although generative design AI can improve the quality of several tasks, it also relies on massive data and may induce several security and privacy issues, especially for fake digital assets, such as photographs or videos, that are indistinguishable from real things.

E. Development of IEEE Standards, Policies, and Regulations

Privacy and security are paramount considerations in the field of distributed learning, where data are shared and processed across various decentralized nodes. To ensure a robust and trustworthy environment for distributed learning systems, several IEEE standards, policies, and regulations come into play. These guidelines help establish a solid foundation for protecting user data and maintaining the integrity of the learning process.

1) IEEE Standards:

a) *IEEE 1363 (Standard Specifications for Public-Key Cryptography):* Encryption is vital for securing data in distributed learning. IEEE 1363 provides specifications for public-key cryptography algorithms, ensuring confidentiality and integrity of communication in distributed systems.

b) *IEEE P2089 (Standard for Privacy Impact Assessment for IoT):* This standard provides a framework for assessing the privacy impact of IoT systems, which often play a crucial role in distributed learning scenarios. It guides the identification of potential privacy risks and suggests mitigation strategies.

c) *IEEE 3652.1-2020 (Guide for Architectural Framework and Application of Federated ML)*[3]*:* It provides

[3]https://standards.ieee.org/standard/3652_1-2020.html

a blueprint for data usage and model building across organizations and devices while meeting applicable privacy, security, and regulatory requirements in FL. In particular, the description and definition; the categories and the application scenarios to which each category applies; the performance evaluation; and the associated regulatory requirements of FL are defined.

d) *IEEE P7000 series (Model Process for Addressing Ethical Concerns During System Design):* Distributed learning involves ethical considerations, and this series offers a comprehensive model process to address ethical concerns throughout system design and development. It emphasizes transparency, accountability, and user consent.

2) Policies and Regulations:

a) *General Data Protection Regulation (GDPR)*[4]*:* Although not an IEEE standard, GDPR is a significant regulation that affects distributed learning. It emphasizes the protection of personal data and requires explicit user consent for data processing. Organizations handling data in distributed learning must adhere to GDPR's principles to ensure user privacy.

b) *Health Insurance Portability and Accountability Act (HIPAA)*[5]*:* In healthcare-related distributed learning applications, HIPAA plays a crucial role. It sets regulations for protecting the privacy and security of patients' health information, including data used in distributed learning scenarios.

c) *National Institute of Standards and Technology (NIST) Guidelines*[6]*:* While not IEEE-specific, NIST provides guidelines on security and privacy, including those applicable to distributed systems. NIST's cybersecurity framework and privacy framework offer valuable insights for building secure and privacy-preserving distributed learning systems.

d) *IEEE Code of Ethics*[7]*:* While not a policy or regulation in the legal sense, the IEEE Code of Ethics guides professionals working in technical fields, including distributed learning. It encourages ethical behavior, respect for privacy, and responsible decision-making.

VIII. CONCLUSION

As an important and emerging technology, distributed ML has the capability to leverage the incremental amount of data in UEs to the maximum extent. However, this emergence raises increased concerns about privacy and security. In this survey, we have proposed a new framework, which divides distributed ML into four levels for the purpose of understanding privacy and security issues. Moreover, we have discussed and summarized the state-of-the-art related to these issues and revealed the particular characteristics of adversaries at each level. In addition, several research challenges and future directions have also been discussed. ■

REFERENCES

[1] J. Chen and X. Ran, "Deep learning with edge computing: A review," *Proc. IEEE*, vol. 107, no. 8, pp. 1655–1674, Aug. 2019.

[2] E. Elsebakhi et al., "Large-scale machine learning based on functional networks for biomedical big data with high performance computing platforms," *J. Comput. Sci.*, vol. 11, pp. 69–81, Nov. 2015.

[3] P. A. Bernstein and E. Newcomer, *Principles of Transaction Processing*, 2nd ed. Burlington, MA, USA: Morgan Kaufmann, 2009.

[4] I. Raicu, I. Foster, A. Szalay, and G. Turcu, "AstroPortal: A science gateway for large-scale astronomy data analysis," in *Proc. Teragrid Conf.*, 2016, pp. 12–15.

[5] R. Gu et al., "From server-based to client-based machine learning: A comprehensive survey," 2019, *arXiv:1909.08329*.

[6] J. Konecný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2016, *arXiv:1610.05492*.

[7] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[8] C. Ma et al., "On safeguarding privacy and security in the framework of federated learning," *IEEE Netw.*, vol. 34, no. 4, pp. 242–248, Jul. 2020.

[9] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol. (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[10] N. Papernot, M. Abadi, Ú. Erlingsson, I. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," 2016, *arXiv:1610.05755*.

[11] Z.-K. Zhang, M. C. Y. Cho, C.-W. Wang, C.-W. Hsu, C.-K. Chen, and S. Shieh, "IoT security: Ongoing challenges and research opportunities," in *Proc. IEEE 7th Int. Conf. Service-Oriented Comput. Appl.*, Nov. 2014, pp. 230–234.

[12] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A survey on security and privacy issues in Internet-of-Things," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1250–1258, Oct. 2017.

[13] D. Peteiro-Barral and B. Guijarro-Berdiñas, "A survey of methods for distributed machine learning," *Prog. Artif. Intell.*, vol. 2, no. 1, pp. 1–11, Mar. 2013.

[14] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, and J. S. Rellermeyer, "A survey on distributed machine learning," *ACM Comput. Surv.*, vol. 53, no. 2, pp. 1–33, Mar. 2020.

[15] E. De Cristofaro, "An overview of privacy in machine learning," 2020, *arXiv:2005.08679*.

[16] B. Liu, M. Ding, S. Shaham, W. Rahayu, F. Farokhi, and Z. Lin, "When machine learning meets privacy: A survey and outlook," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–36, Mar. 2021.

[17] P. Vepakomma, T. Swedish, R. Raskar, O. Gupta, and A. Dubey, "No peek: A survey of private distributed deep learning," 2018, *arXiv:1812.03288*.

[18] M. Gong, Y. Xie, K. Pan, K. Feng, and A. K. Qin, "A survey on differentially private machine learning," *IEEE Comput. Intell. Mag.*, vol. 15, no. 2, pp. 49–64, May 2020.

[19] M. Amiri-Zarandi, R. A. Dara, and E. Fraser, "A survey of machine learning-based solutions to protect privacy in the Internet of Things," *Comput. Secur.*, vol. 96, Sep. 2020, Art. no. 101921.

[20] C. Briggs, Z. Fan, and P. Andras, "A review of privacy-preserving federated learning for the Internet-of-Things," 2020, *arXiv:2004.11794*.

[21] D. Enthoven and Z. Al-Ars, *An Overview of Federated Deep Learning Privacy Attacks and Defensive Strategies*. Cham, Switzerland: Springer, 2021.

[22] L. Lyu, H. Yu, and Q. Yang, "Threats to federated learning: A survey," 2020, *arXiv:2003.02133*.

[23] G. Xu, H. Li, H. Ren, K. Yang, and R. H. Deng, "Data security issues in deep learning: Attacks, countermeasures, and opportunities," *IEEE Commun. Mag.*, vol. 57, no. 11, pp. 116–122, Nov. 2019.

[24] R. Xu, N. Baracaldo, and J. Joshi, "Privacy-preserving machine learning: Methods, challenges and directions," 2021, *arXiv:2108.04417*.

[25] R. J. Schalkoff, *Pattern Recognition*. Atlanta, GA, USA: American Cancer Society, 2007.

[26] *Training, Validation, and Test Sets—Wikipedia, the Free Encyclopedia*, Wikimedia Found., Inc., San Francisco, CA, USA, 2021.

[27] E. P. Xing, Q. Ho, P. Xie, and D. Wei, "Strategies and principles of distributed machine learning on big data," *Engineering*, vol. 2, no. 2, pp. 179–195, Jun. 2016.

[4]https://ec.europa.eu/info/law/law-topic/data-protection_en

[5]https://www.hhs.gov/hipaa/index.html

[6]https://csrc.nist.gov/

[7]https://www.ieee.org/about/ieee-code-of-ethics.html

[28] M. Zinkevich, M. Weimer, L. Li, and A. Smola, "Parallelized stochastic gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 23, 2010, pp. 1–9.

[29] Q. Ho et al., "More effective distributed ML via a stale synchronous parallel parameter server," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2013, pp. 1223–1231.

[30] X. Lian, Y. Huang, Y. Li, and J. Liu, "Asynchronous parallel stochastic gradient for nonconvex optimization," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. 2015, pp. 2737–2745.

[31] B. Recht, C. Re, S. Wright, and F. Niu, "Hogwild: A lock-free approach to parallelizing stochastic gradient descent," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2011, pp. 693–701.

[32] M. Li, D. G. Andersen, A. J. Smola, and K. Yu, "Communication efficient distributed machine learning with the parameter server," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2014, pp. 19–27.

[33] M. Abadi et al., "TensorFlow: A system for large-scale machine learning," in *Proc. USENIX Symp. Operating Syst. Design Implement. (OSDI)*, Savannah, GA, Nov. 2016, pp. 265–283.

[34] T. Chilimbi, Y. Suzue, J. Apacible, and K. Kalyanaraman, "Project Adam: Building an efficient and scalable deep learning training system," in *Proc. USENIX Symp. Operating Syst. Design Implement. (OSDI)*, Broomfield, CO, USA, Oct. 2014, pp. 571–582.

[35] K. Hsieh et al., "Gaia: Geo-distributed machine learning approaching LAN speeds," in *Proc. USENIX Symp. Networked Syst. Design Implement. (NSDI)*, Boston, MA, USA, Mar. 2017, pp. 629–647.

[36] M. Li, Z. Liu, A. J. Smola, and Y.-X. Wang, "DiFacto: Distributed factorization machines," in *Proc. 9th ACM Int. Conf. Web Search Data Mining*, San Francisco, CA, USA, Feb. 2016, pp. 377–386.

[37] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2017, pp. 1273–1282.

[38] J. Konecný, B. McMahan, and D. Ramage, "Federated optimization: Distributed optimization beyond the datacenter," 2015, *arXiv:1511.03575*.

[39] K. Bonawitz et al., "Towards federated learning at scale: System design," 2019, *arXiv:1902.01046*.

[40] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Commun. ACM*, vol. 24, no. 2, pp. 84–90, Feb. 1981.

[41] W. Y. B. Lim et al., "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, 3rd Quart., 2020.

[42] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2019, pp. 691–706.

[43] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, Denver, CO, USA, Oct. 2015, pp. 1322–1333.

[44] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the GAN: Information leakage from collaborative deep learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 603–618.

[45] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, vol. 108, Aug. 2020, pp. 2938–2948.

[46] C. He, M. Annavaram, and S. Avestimehr, "Group knowledge transfer: Federated learning of large CNNs at the edge," in *Proc. Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Vancouver, BC, Canada, 2020, pp. 14068–14080.

[47] P. Vepakomma, O. Gupta, T. Swedish, and R. Raskar, "Split learning for health: Distributed deep learning without sharing raw patient data," 2018, *arXiv:1812.00564*.

[48] P. Vepakomma, O. Gupta, A. Dubey, and R. Raskar, "Reducing leakage in distributed deep learning for sensitive health data," presented at the ICLR AI Social Good Workshop, 2019.

[49] O. Gupta and R. Raskar, "Distributed learning of deep neural network over multiple agents," *J. Netw. Comput. Appl.*, vol. 116, pp. 1–8, Aug. 2018.

[50] P. Goyal et al., "Accurate, large minibatch SGD: Training ImageNet in 1 hour," 2017, *arXiv:1706.02677*.

[51] J. Chen, X. Pan, R. Monga, S. Bengio, and R. Jozefowicz, "Revisiting distributed synchronous SGD," 2016, *arXiv:1604.00981*.

[52] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proc. 53rd Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2015, pp. 909–910.

[53] W. Li, B. Jin, X. Wang, J. Yan, and H. Zha, "F2A2: Flexible fully-decentralized approximate actor-critic for cooperative multi-agent reinforcement learning," 2020, *arXiv:2004.11145*.

[54] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 3–18.

[55] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, Apr. 2017, pp. 506–519.

[56] C. Ma, J. Li, M. Ding, K. Wei, W. Chen, and H. V. Poor, "Federated learning with unreliable clients: Performance analysis and mechanism design," *IEEE Internet Things J.*, vol. 8, no. 24, pp. 17308–17319, Dec. 2021.

[57] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.

[58] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2018, pp. 19–35.

[59] O. Suciu, R. Marginean, Y. Kaya, H. Daume III, and T. Dumitras, "When does machine learning FAIL? Generalized transferability for evasion and poisoning attacks," in *Proc. USENIX Secur. Symp.*, Baltimore, MD, USA, Aug. 2018, pp. 1299–1316.

[60] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2805–2824, Sep. 2019.

[61] H. Kwon, Y. Kim, H. Yoon, and D. Choi, "Selective audio adversarial example in evasion attack on speech recognition system," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 526–538, 2020.

[62] K. Eykholt et al., "Robust physical-world attacks on deep learning visual classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1625–1634.

[63] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," in *Proc. USENIX Secur. Symp.*, 2014, pp. 17–32.

[64] J. A. Calandrino, A. Kilzer, A. Narayanan, E. W. Felten, and V. Shmatikov, "'You might also like': Privacy risks of collaborative filtering," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2011, pp. 231–246.

[65] C. Song, T. Ristenpart, and V. Shmatikov, "Machine learning models that remember too much," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 587–601.

[66] G. Ateniese, L. V. Mancini, A. Spognardi, A. Villani, D. Vitali, and G. Felici, "Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers," *Int. J. Secur. Netw.*, vol. 10, no. 3, pp. 137–150, 2015.

[67] Y. Phong et al., "Privacy-preserving deep learning: Revisited and enhanced," in *Proc. Int. Conf. Appl. Techn. Inf. Secur. (ATIS)*, 2017, pp. 100–110.

[68] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," 2018, *arXiv:1805.04049*.

[69] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei, "Towards demystifying membership inference attacks," 2018, *arXiv:1807.09173*.

[70] M. Backes, P. Berrang, M. Humbert, and P. Manoharan, "Membership privacy in MicroRNA-based studies," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 319–330.

[71] N. Homer et al., "Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays," *PLoS Genet.*, vol. 4, no. 8, Aug. 2008, Art. no. e1000167.

[72] A. Pyrgelis, C. Troncoso, and E. De Cristofaro, "Knock knock, who's there? Membership inference on aggregate location data," 2017, *arXiv:1708.06145*.

[73] C. Dwork, A. Smith, T. Steinke, J. Ullman, and S. Vadhan, "Robust traceability from trace amounts," in *Proc. IEEE 56th Annu. Symp. Found. Comput. Sci.*, Oct. 2015, pp. 650–669.

[74] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction APIS," in *Proc. USENIX Secur. Symp.*, 2016, pp. 601–618.

[75] B. Wang and N. Z. Gong, "Stealing hyperparameters in machine learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2018, pp. 36–52.

[76] S. J. Oh, B. Schiele, and M. Fritz, *Towards Reverse-Engineering Black-Box Neural Networks*. Cham, Switzerland: Springer, 2019.

[77] T. Orekondy, B. Schiele, and M. Fritz, "Knockoff nets: Stealing functionality of black-box models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4949–4958.

[78] K. Bonawitz et al., "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Dallas, TX, USA, Oct. 2017, pp. 1175–1191.

[79] J. Liu, M. Juuti, Y. Lu, and N. Asokan, "Oblivious neural network predictions via MiniONN transformations," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Dallas, TX, USA, Oct. 2017, pp. 619–631.

[80] P. Mohassel and Y. Zhang, "SecureML: A system for scalable privacy-preserving machine learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 19–38.

[81] V. Nikolaenko, U. Weinsberg, S. Ioannidis, M. Joye, D. Boneh, and N. Taft, "Privacy-preserving ridge regression on hundreds of millions of records," in *Proc. IEEE Symp. Secur. Privacy*, May 2013, pp. 334–348.

[82] W. Du, Y. S. Han, and S. Chen, "Privacy-preserving multivariate statistical analysis: Linear regression and classification," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2004, pp. 222–233.

[83] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, "Machine learning classification over encrypted data," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, 2014, pp. 1–34.

[84] J. W. Bos, K. Lauter, and M. Naehrig, "Private predictive analysis on encrypted medical data," *J. Biomed. Informat.*, vol. 50, pp. 234–243, Aug. 2014.

[85] T. Graepel, K. Lauter, and M. Naehrig, "ML confidential: Machine learning on encrypted data," in *Proc. Int. Conf. Inf. Secur. Cryptol. (ICISC)*, 2012, pp. 1–21.

[86] N. Kilbertus, A. Gascón, M. J. Kusner, M. Veale, K. P. Gummadi, and A. Weller, "Blind justice: Fairness with encrypted sensitive attributes," 2018, *arXiv:1806.03281*.

[87] Y. Li, Y. Duan, Y. Yu, S. Zhao, and W. Xu, "PrivPy: Enabling scalable and general privacy-preserving machine learning," 2018, *arXiv:1801.10117*.

[88] P. Mohassel and P. Rindal, "ABY^3: A mixed protocol framework for machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2018, pp. 35–52.

[89] T. Araki, J. Furukawa, Y. Lindell, A. Nof, and

K. Ohara, "High-throughput semi-honest secure three-party computation with an honest majority," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 805–817.
[90] P. Mohassel, M. Rosulek, and Y. Zhang, "Fast and secure three-party computation: The garbled circuit approach," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2015, pp. 591–602.
[91] J. Furukawa, Y. Lindell, A. Nof, and O. Weinstein, "High-throughput secure three-party computation for malicious adversaries and an honest majority," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn. (EUROCRYPT)*, 2017, pp. 225–255.
[92] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–35, 2018.
[93] Q. Zhang, L. T. Yang, and Z. Chen, "Privacy preserving deep computation model on cloud for big data feature learning," *IEEE Trans. Comput.*, vol. 65, no. 5, pp. 1351–1362, May 2016.
[94] J. Yuan and S. Yu, "Privacy preserving back-propagation neural network learning made practical with cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 1, pp. 212–221, Jan. 2014.
[95] W. Lu, S. Kawasaki, and J. Sakuma, "Using fully homomorphic encryption for statistical analysis of categorical, ordinal and numerical data," in *Proc. Annu. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, San Diego, CA, USA, 2017, pp. 1–21.
[96] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, Jul. 1982, doi: 10.1145/357172.357176.
[97] P. Blanchard, E. M. E. Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA, 2017, pp. 119–129.
[98] E. M. El Mhamdi, R. Guerraoui, and S. Rouault, "The hidden vulnerability of distributed learning in byzantium," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 80, Jul. 2018, pp. 3521–3530.
[99] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, and J. Han, "Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, Jun. 2014, pp. 1187–1198.
[100] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *IEEE Trans. Signal Process.*, vol. 70, pp. 1142–1154, 2022.
[101] H. Chang, V. Shejwalkar, R. Shokri, and A. Houmansadr, "Cronus: Robust and heterogeneous collaborative learning with black-box knowledge transfer," 2019, *arXiv:1912.11279*.
[102] I. Diakonikolas, G. Kamath, D. M. Kane, J. Li, A. Moitra, and A. Stewart, "Being robust (in high dimensions) can be practical," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 70, Aug. 2017, pp. 999–1008.
[103] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 30, 2017, pp. 1–12.
[104] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "signSGD: Compressed optimisation for non-convex problems," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 80, Jul. 2018, pp. 560–569.
[105] W. Wen et al., "TernGrad: Ternary gradients to reduce communication in distributed deep learning," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA, 2017, pp. 1509–1519.
[106] D. Alistarh, T. Hoefler, M. Johansson, N. Konstantinov, S. Khirirat, and C. Renggli, "The convergence of sparsified gradient methods," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2018, pp. 5977–5987.
[107] S. U. Stich, J. Cordonnier, and M. Jaggi, "Sparsified SGD with memory," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Montreal, QC, Canada, 2018, pp. 4452–4463.
[108] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, *arXiv:1312.4400*.
[109] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Communication-efficient on-device machine learning: Federated distillation and augmentation under non-IID private data," 2018, *arXiv:1811.11479*.
[110] Z. Liu, M. Sun, T. Zhou, G. Huang, and T. Darrell, "Rethinking the value of network pruning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, New Orleans, LA, USA, 2019, pp. 1–21.
[111] Y. Jiang et al., "Model pruning enables efficient federated learning on edge devices," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Apr. 25, 2022, doi: 10.1109/TNNLS.2022.3166101.
[112] F. Haddadpour, B. Karimi, P. Li, and X. Li, "FedSKETCH: Communication-efficient and private federated learning via sketching," 2020, *arXiv:2008.04975*.
[113] J. Jiang, F. Fu, T. Yang, and B. Cui, "SketchML: Accelerating distributed machine learning with data sketches," in *Proc. Int. Conf. Manag. Data*, Houston, TX, USA, May 2018, pp. 1269–1284.
[114] T. Li, Z. Liu, V. Sekar, and V. Smith, "Privacy for free: Communication-efficient learning with differential privacy using sketches," 2019, *arXiv:1911.00972*.
[115] J. Yoon, W. Jeong, G. Lee, E. Yang, and S. J. Hwang, "Federated continual learning with weighted inter-client transfer," 2020, *arXiv:2003.03196*.
[116] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 5, pp. 1333–1345, May 2018.
[117] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, vol. 108, Aug. 2020, pp. 2021–2031.
[118] D. Rothchild et al., "FetchSGD: Communication-efficient federated learning with sketching," in *Proc. 37th Int. Conf. Mach. Learn. (ICML)*, vol. 119, Jul. 2020, pp. 8253–8265.
[119] R. Jin, Y. Huang, X. He, H. Dai, and T. Wu, "Stochastic-sign SGD for federated learning with theoretical guarantees," 2020, *arXiv:2002.10940*.
[120] H. Li and T. Han, "An end-to-end encrypted neural network for gradient updates transmission in federated learning," 2019, *arXiv:1908.08340*.
[121] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012, *arXiv:1207.0580*.
[122] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.
[123] C. Dwork, "Differential privacy: A survey of results," in *Proc. Int. Conf. Theory Appl. Models Comput. (TAMC)*, 2008, pp. 1–19.
[124] B. I. P. Rubinstein, P. L. Bartlett, L. Huang, and N. Taft, "Learning in a large function space: Privacy-preserving mechanisms for SVM learning," 2009, *arXiv:0911.5708*.
[125] J. Zhang, Z. Zhang, X. Xiao, Y. Yang, and M. Winslett, "Functional mechanism: Regression analysis under differential privacy," 2012, *arXiv:1208.0219*.
[126] M. Abadi et al., "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2016, pp. 308–318.
[127] I. Mironov, "Rényi differential privacy," in *Proc. IEEE Comput. Secur. Found. Symp. (CSF)*, Aug. 2017, pp. 263–275.
[128] C. Dwork and G. N. Rothblum, "Concentrated differential privacy," 2016, *arXiv:1603.01887*.
[129] M. Nasr, S. Songi, A. Thakurta, N. Papernot, and N. Carlin, "Adversary instantiation: Lower bounds for differentially private machine learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, San Francisco, CA, USA, May 2021, pp. 866–882.
[130] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *Proc. 51st Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Oct. 2013, p. 1592.
[131] T. Wang, J. Blocki, N. Li, and S. Jha, "Locally differentially private protocols for frequency estimation," in *Proc. USENIX Secur. Symp.*, Vancouver, BC, USA, Aug. 2017, pp. 729–745.
[132] R. C.-W. Wong and A. W.-C. Fu, *Privacy-Preserving Data Publishing: An Overview*. Morgan & Claypool, San Rafael, CA, USA, 2010.
[133] Y. Dong et al., "Boosting adversarial attacks with momentum," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 9185–9193.
[134] S. Ji, P. Mittal, and R. Beyah, "Graph data anonymization, de-anonymization attacks, and de-anonymizability quantification: A survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1305–1326, 2nd Quart., 2017.
[135] S. H. Huang, N. Papernot, I. J. Goodfellow, Y. Duan, and P. Abbeel, "Adversarial attacks on neural network policies," in *Proc. Int. Conf. Learn. Represent.*, Toulon, France, Apr. 2017, pp. 1–10.
[136] X. Pan et al., "Characterizing attacks on deep reinforcement learning," in *Proc. Int. Conf. Auto. Agents Multiagent Syst. (AAMAS)*, Virtual Event, New Zealand, 2022, pp. 1010–1018.
[137] C. Zhong, F. Wang, M. C. Gursoy, and S. Velipasalar, "Adversarial jamming attacks on deep reinforcement learning based dynamic multichannel access," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, May 2020, pp. 1–6.
[138] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2008, pp. 111–125.
[139] Y. Dong, Z. Deng, T. Pang, J. Zhu, and H. Su, "Adversarial distributional training for robust deep learning," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, Dec. 2020, pp. 8270–8283.
[140] L. Sweeney, "K-anonymity: A model for protecting privacy," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, Oct. 2002.
[141] S. Shaham, M. Ding, B. Liu, S. Dang, Z. Lin, and J. Li, "Privacy preserving location data publishing: A machine learning approach," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 9, pp. 3270–3283, Sep. 2021.
[142] M. Malekzadeh, R. G. Clegg, A. Cavallaro, and H. Haddadi, "Mobile sensor data anonymization," in *Proc. Int. Conf. Internet Things Design Implement.*, Montreal, QC, Canada, Apr. 2019, pp. 49–58.
[143] M. Maximov, I. Elezi, and L. Leal-Taixe, "CIAGAN: Conditional identity anonymization generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 5446–5455.
[144] P. Samarati and L. Sweeney, "Protecting privacy when disclosing information: K-anonymity and its enforcement through generalization and suppression," Carnegie Mellon Univ., Pittsburgh, PA, USA, 2018, doi: 10.1184/R1/6625469.v1.
[145] P. Samarati, "Protecting respondents identities in microdata release," *IEEE Trans. Knowl. Data Eng.*, vol. 13, no. 6, pp. 1010–1027, Nov. 2001.
[146] K. El Emam and F. K. Dankar, "Protecting privacy using k-anonymity," *J. Amer. Med. Inform. Assoc.*, vol. 15, no. 5, pp. 627–637, Sep. 2008.
[147] N. Li, T. Li, and S. Venkatasubramanian, "T-closeness: Privacy beyond k-Anonymity and l-Diversity," in *Proc. IEEE 23rd Int. Conf. Data Eng.*, Apr. 2007, pp. 106–115.
[148] S. Shaham, M. Ding, B. Liu, S. Dang, Z. Lin, and J. Li, "Privacy preservation in location-based services: A novel metric and attack model," *IEEE Trans. Mobile Comput.*, vol. 20, no. 10, pp. 3006–3019, Oct. 2021.
[149] A. M. Shaker, M. Tantawi, H. A. Shedeed, and M. F. Tolba, "Generalization of convolutional neural networks for ECG classification using

generative adversarial networks," *IEEE Access*, vol. 8, pp. 35592–35605, 2020.
[150] Z. Ji, Z. C. Lipton, and C. Elkan, "Differential privacy and machine learning: A survey and review," 2014, *arXiv:1412.7584*.
[151] Ú. Erlingsson, V. Pihur, and A. Korolova, "RAPPOR: Randomized aggregatable privacy-preserving ordinal response," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2014, pp. 1054–1067.
[152] G. Fanti, V. Pihur, and Ú. Erlingsson, "Building a RAPPOR with the unknown: Privacy-preserving learning of associations and data dictionaries," *Proc. Privacy Enhancing Technol.*, vol. 2016, no. 3, pp. 41–61, Jul. 2016.
[153] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. Theory Cryptography Conf. (TCC)*, 2006, pp. 265–284.
[154] M. Balcan, A. Blum, S. Fine, and Y. Mansour, "Distributed learning, communication complexity and privacy," in *Proc. Annu. Conf. Learn. Theory (COLT)*, vol. 23, Edinburgh, Scotland, Jun. 2012, pp. 1–22.
[155] O. Fagbohungbe, S. R. Reza, X. Dong, and L. Qian, "Efficient privacy preserving edge computing framework for image classification," 2020, *arXiv:2005.04563*.
[156] D. Froelicher et al., "Scalable privacy-preserving distributed learning," *Proc. Privacy Enhancing Technol.*, vol. 2021, no. 2, pp. 323–347, Apr. 2021.
[157] K. Xu, H. Yue, L. Guo, Y. Guo, and Y. Fang, "Privacy-preserving machine learning algorithms for big data systems," in *Proc. IEEE 35th Int. Conf. Distrib. Comput. Syst.*, Columbus, OH, USA, Jun. 2015, pp. 318–327.
[158] R. Zhang and Q. Zhu, "Secure and resilient distributed machine learning under adversarial environments," in *Proc. 18th Int. Conf. Inf. Fusion (Fusion)*, Washington, DC, USA, Jul. 2015, pp. 644–651.
[159] J. Hur, "Improving security and efficiency in attribute-based data sharing," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 10, pp. 2271–2282, Oct. 2013.
[160] S. L. Warner, "Randomized response: A survey technique for eliminating evasive answer bias," *J. Amer. Stat. Assoc.*, vol. 60, no. 309, pp. 63–69, Mar. 1965.
[161] (2022). *Learning With Privacy at Scale-Apple Machine Learning Research*. [Online]. Available: https://machinelearning.pple.com/research/learning-with-privacy-at-scale
[162] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 97, Long Beach, CA, USA, Jun. 2019, pp. 634–643.
[163] M. Fang, X. Cao, J. Jia, and N. Z. Gong, "Local model poisoning attacks to Byzantine-robust federated learning," in *Proc. USENIX Secur. Symp.*, Aug. 2020, pp. 1605–1622.
[164] X. Pan, W. Wang, X. Zhang, B. Li, J. Yi, and D. Song, "How you act tells a lot: Privacy-leaking attack on deep reinforcement learning," in *Proc. Int. Conf. Auto. Agents MultiAgent Syst. (AAMAS)*, Montreal QC, Canada, 2019, pp. 368–376.
[165] Y. Zhao, I. Shumailov, H. Cui, X. Gao, R. Mullins, and R. Anderson, "Blackbox attacks on reinforcement learning agents using approximated temporal information," in *Proc. 50th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. Workshops (DSN-W)*, Jun. 2020, pp. 16–24.
[166] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2019, pp. 2512–2520.
[167] J. Ge, Z. Wang, M. Wang, and H. Liu, "Minimax-optimal privacy-preserving sparse PCA in distributed systems," in *Proc. 21st Int. Conf. Artif. Intell. Statist. (AISTATS)*, vol. 84, Apr. 2018, pp. 1589–1598.
[168] Z. Huang, R. Hu, Y. Guo, E. Chan-Tin, and Y. Gong, "DP-ADMM: ADMM-based distributed learning with differential privacy," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1002–1012, 2020.
[169] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," 2017, *arXiv:1712.07557*.
[170] K. Wei et al., "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3454–3469, 2020.
[171] C. X. Wang, Y. Song, and W. P. Tay, "Arbitrarily strong utility-privacy tradeoff in multi-agent systems," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 671–684, 2021.
[172] D. Ye, T. Zhu, W. Zhou, and P. S. Yu, "Differentially private malicious agent avoidance in multiagent advising learning," *IEEE Trans. Cybern.*, vol. 50, no. 10, pp. 4214–4227, Oct. 2020.
[173] V. Gandikota, D. Kane, R. K. Maity, and A. Mazumdar, "vqSGD: Vector quantized stochastic gradient descent," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, vol. 130, Apr. 2021, pp. 2197–2205.
[174] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, "VerifyNet: Secure and verifiable federated learning," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 911–926, 2020.
[175] L. Muñoz-González, K. T. Co, and E. C. Lupu, "Byzantine-robust federated machine learning through adaptive model averaging," 2019, *arXiv:1909.05125*.
[176] S. Fu, C. Xie, B. Li, and Q. Chen, "Attack-resistant federated learning with residual-based reweighting," 2019, *arXiv:1912.11464*.
[177] L. Zhao et al., "Shielding collaborative learning: Mitigating poisoning attacks through client-side detection," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 5, pp. 2029–2041, Sep. 2021.
[178] Y. Zhao, J. Chen, J. Zhang, D. Wu, J. Teng, and S. Yu, "PDGAN: A novel poisoning defense method in federated learning using generative adversarial network," in *Proc. Int. Conf. Algorithms Architectures Parallel Process. (ICAPP)*, 2020, pp. 595–609.
[179] Y. Wang and M. P. Singh, "Formal trust model for multiagent systems," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, M. M. Veloso, Ed. Hyderabad, India, Jan. 2007, pp. 1551–1556.
[180] K. Danilov, R. Rezin, I. Afanasyev, and A. Kolotov, "Towards blockchain-based robonomics: Autonomous agents behavior validation," in *Proc. Int. Conf. Intell. Syst. (IS)*, Sep. 2018, pp. 222–227.
[181] V. Strobel, E. C. Ferrer, and M. Dorigo, "Managing Byzantine robots via blockchain technology in a swarm robotics collective decision making scenario," in *Proc. Int. Conf. Auto. Agents MultiAgent Syst. (AAMAS)*, Stockholm, Sweden, Jul. 2018, pp. 541–549.
[182] D. Calvaresi, Y. Mualla, A. Najjar, S. Galland, and M. Schumacher, "Explainable multi-agent systems through blockchain technology," in *Proc. Int. Workshop Explainable, Transparent Auto. Agents Multi-Agent Syst. (EXTRAAMAS)*, vol. 11763, Montreal, QC, Canada, May 2019, pp. 41–58.
[183] L. Xiao et al., "An adaptive security model for multi-agent systems and application to a clinical trials environment," in *Proc. 31st Annu. Int. Comput. Softw. Appl. Conf.*, Beijing, China, Jul. 2007, pp. 261–268.
[184] S. Ahmad and M. U. Bokhari, "A new approach to multi agent based architecture for secure and effective e-learning," *Int. J. Comput. Appl.*, vol. 46, no. 22, pp. 26–29, May 2012.
[185] S. Bosse, "Mobile multi-agent systems for the Internet-of-Things and clouds using the Javascript agent machine platform and machine learning as a service," in *Proc. IEEE 4th Int. Conf. Future Internet Things Cloud (FiCloud)*, Aug. 2016, pp. 244–253.
[186] P. Kairouz et al., "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, nos. 1–2, pp. 1–210, 2019.
[187] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar, "Adversarial machine learning," in *Proc. ACM Workshop Security Artif. Intell. (AISec)*, Chicago, IL, USA, 2011, pp. 43–58.
[188] G. Baruch, M. Baruch, and Y. Goldberg, "A little is enough: Circumventing defenses for distributed learning," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2019, pp. 8632–8642.
[189] X. Gong, Y. Chen, H. Huang, Y. Liao, S. Wang, and Q. Wang, "Coordinated backdoor attacks against federated learning with model-dependent triggers," *IEEE Netw.*, vol. 36, no. 1, pp. 84–90, Jan. 2022.
[190] X. Luo, Y. Wu, X. Xiao, and B. C. Ooi, "Feature inference attack on model predictions in vertical federated learning," in *Proc. IEEE 37th Int. Conf. Data Eng. (ICDE)*, Apr. 2021, pp. 181–192.
[191] Y.-C. Lin, Z.-W. Hong, Y.-H. Liao, M.-L. Shih, M.-Y. Liu, and M. Sun, "Tactics of adversarial attack on deep reinforcement learning agents," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 3756–3762.
[192] V. Behzadan and A. Munir, "Vulnerability of deep reinforcement learning to policy induction attacks," in *Proc. Int. Conf. Mach. Learn. Data Mining Pattern Recognit. (MLDM)*, 2017, pp. 262–275.
[193] A. Russo and A. Proutiere, "Optimal attacks on reinforcement learning policies," 2019, *arXiv:1907.13548*.
[194] Y. Ma, X. Zhang, W. Sun, and X. Zhu, "Policy poisoning in batch reinforcement learning and control," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Vancouver, BC, Canada, Dec. 2019, pp. 14543–14553.
[195] P. Kiourti, K. Wardega, S. Jha, and W. Li, "TrojDRL: Evaluation of backdoor attacks on deep reinforcement learning," in *Proc. 57th ACM/IEEE Design Autom. Conf. (DAC)*, Jul. 2020, pp. 1–6.
[196] X. Zhang, Y. Ma, A. Singla, and X. Zhu, "Adaptive reward-poisoning attacks against reinforcement learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jul. 2020, pp. 11225–11234.
[197] M. Huai, J. Sun, R. Cai, L. Yao, and A. Zhang, "Malicious attacks against deep reinforcement learning interpretations," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 472–482.
[198] J. Sun et al., "Stealthy and efficient adversarial attacks against deep reinforcement learning," in *Proc. 34th AAAI Conf. Artif. Intell. (AAAI)*, Feb. 2020, pp. 5883–5891.
[199] A. Rakhsha, G. Radanovic, R. Devidze, X. Zhu, and A. Singla, "Policy teaching via environment poisoning: Training-time adversarial attacks against reinforcement learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 119, Jul. 2020, pp. 7974–7984.
[200] A. Lonzetta, P. Cope, J. Campbell, B. Mohd, and T. Hayajneh, "Security vulnerabilities in Bluetooth technology as used in IoT," *J. Sensor Actuator Netw.*, vol. 7, no. 3, p. 28, Jul. 2018.
[201] A. Gleave, M. Dennis, N. Kant, C. Wild, S. Levine, and S. Russell, "Adversarial policies: Attacking deep reinforcement learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Addis Ababa, Ethiopia, Apr. 2020, pp. 1–16.
[202] A. N. Bhagoji, W. He, B. Li, and D. Song, "Practical black-box attacks on deep neural networks using efficient query mechanisms," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 154–169.
[203] M. Inkawhich, Y. Chen, and H. Li, "Snooping attacks on deep reinforcement learning," in *Proc. Int. Conf. Auto. Agents MultiAgent Syst. (AAMAS)*, Auckland, New Zealand, 2020, pp. 557–565.
[204] J. Liu, J. Lou, L. Xiong, J. Liu, and X. Meng, "Projected federated averaging with heterogeneous differential privacy," *Proc. VLDB Endowment*, vol. 15, no. 4, pp. 828–840, Apr. 2022.
[205] C. Zhang, S. Ekanut, L. Zhen, and Z. Li, "Augmented multi-party computation against gradient leakage in federated learning," *IEEE Trans. Big Data*, early access, Sep. 22, 2022, doi: 10.1109/TBDATA.2022.3208736.
[206] R. Kanagavelu et al., "Two-phase multi-party computation enabled privacy-preserving federated learning," in *Proc. 20th IEEE/ACM Int. Symp.*

Cluster, Cloud Internet Comput. (CCGRID), May 2020, pp. 410–419.
[207] E. Sotthiwat, L. Zhen, Z. Li, and C. Zhang, "Partially encrypted multi-party computation for federated learning," in *Proc. IEEE/ACM 21st Int. Symp. Cluster, Cloud Internet Comput. (CCGrid)*, May 2021, pp. 828–835.
[208] W. Mou, C. Fu, Y. Lei, and C. Hu, "A verifiable federated learning scheme based on secure multi-party computation," in *Proc. 16th Int. Conf. Wireless Algorithms, Syst., Appl. (WASA)*, Jun. 2021, pp. 198–209.
[209] A. S. Shamsabadi, A. Gascón, H. Haddadi, and A. Cavallaro, "PrivEdge: From local to distributed private training and prediction," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3819–3831, 2020.
[210] C. Xie, M. Chen, P.-Y. Chen, and B. Li, "CRFL: Certifiably robust federated learning against backdoor attacks," in *Proc. Int. Conf. Mach. Learn.*, vol. 139, Jul. 2021, pp. 11372–11382.
[211] X. Cao, J. Jia, Z. Zhang, and N. Z. Gong, "FedRecover: Recovering from poisoning attacks in federated learning using historical information," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2023, pp. 326–343.
[212] X. Cao, Z. Zhang, J. Jia, and N. Z. Gong, "FLCert: Provably secure federated learning against poisoning attacks," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 3691–3705, 2022.
[213] Z. Zhang, X. Cao, J. Jia, and N. Z. Gong, "FLDetector: Defending federated learning against model poisoning attacks via detecting malicious clients," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2022, pp. 2545–2555.
[214] F. Tahmasebian, J. Lou, and L. Xiong, "RobustFed: A truth inference approach for robust federated learning," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manag.*, Oct. 2022, pp. 1868–1877.
[215] P. Novák, M. Rollo, J. Hodík, and T. Vlcek, "Communication security in multi-agent systems," in *Proc. Int. Central Eastern Eur. Conf. Multi-Agent Syst. (CEEMAS)*, vol. 2691, Prague, Czech Republic, Jun. 2003, pp. 454–463.
[216] T. D. Huynh, "A personalized framework for trust assessment," in *Proc. ACM Symp. Appl. Comput.*, Honolulu, HI, USA, Mar. 2009, pp. 1302–1307.
[217] (2022). *AI Solution for Self-Learning EMR*. [Online]. Available: https://www.basebit.me/en/solution1.aspx
[218] O. Li et al., "Label leakage and protection in two-party split learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022, pp. 1–27.
[219] S. Abuadbba et al., "Can we use split learning on 1D CNN models for privacy preserving training?" in *Proc. 15th ACM Asia Conf. Comput. Commun. Secur.*, Oct. 2020, pp. 305–318.
[220] H. Weng et al., "Practical privacy attacks on vertical federated learning," 2020, *arXiv:2011.09290*.
[221] S. Itahara, T. Nishio, Y. Koda, M. Morikura, and K. Yamamoto, "Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-IID private data," *IEEE Trans. Mobile Comput.*, vol. 22, no. 1, pp. 191–205, Jan. 2023.
[222] D. Romanini et al., "PyVertical: A vertical federated learning framework for multi-headed splitNN," in *Proc. ICLR Workshop Distrib. Private Mach. Learn. (DPML)*, 2021, pp. 1–9.
[223] L. Sun and L. Lyu, "Federated model distillation with noise-free differential privacy," in *Proc. 13th Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 1563–1570.
[224] E. Erdogan, A. Kupçu, and A. E. Çiçek, "UnSplit: Data-oblivious model inversion, model stealing, and label inference attacks against split learning," in *Proc. 21st Workshop Privacy Electron. Soc.*, Nov. 2022, pp. 115–124.
[225] C. Fu et al., "Label inference attacks against vertical federated learning," in *Proc. 31st USENIX Secur. Symp.*, Aug. 2022, pp. 1397–1414.
[226] J. Liu and X. Lyu, "Clustering label inference attack against practical split learning," 2022, *arXiv:2203.05222*.
[227] P. Ye, Z. Jiang, W. Wang, B. Li, and B. Li, "Feature reconstruction attacks and countermeasures of DNN training in vertical federated learning," 2022, *arXiv:2210.06771*.
[228] P. Qiu et al., "Your labels are selling you out: Relation leaks in vertical federated learning," *IEEE Trans. Dependable Secure Comput.*, early access, Sep. 22, 2022, doi: 10.1109/TDSC.2022.3208630.
[229] X. Jin, P.-Y. Chen, C.-Y. Hsu, C.-M. Yu, and T. Chen, "CAFE: Catastrophic data leakage in vertical federated learning," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 34, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds. 2021, pp. 994–1006.
[230] M. Pathak, S. Rane, and B. Raj, "Multiparty differential privacy via aggregation of locally trained classifiers," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2010, pp. 1876–1884.
[231] Y. Wu, S. Cai, X. Xiao, G. Chen, and B. C. Ooi, "Privacy preserving vertical federated learning for tree-based models," *Proc. VLDB Endowment*, vol. 13, no. 12, pp. 2090–2103, Aug. 2020.
[232] Y. Zhang and H. Zhu, "Additively homomorphical encryption based deep neural network for asymmetrically collaborative machine learning," 2020, *arXiv:2007.06849*.
[233] Z. Tian, R. Zhang, X. Hou, J. Liu, and K. Ren, "FederBoost: Private federated learning for GBDT," 2020, *arXiv:2011.02796*.
[234] A. Li, Y. Duan, H. Yang, Y. Chen, and J. Yang, "TIPRDC: Task-independent privacy-respecting data crowdsourcing framework for deep learning with anonymized intermediate representations," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 824–832.
[235] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, *Federated Learning*. Cham, Switzerland: Springer, 2019.
[236] (2022). *Overview-Fate*. [Online]. Available: https://fate.fedai.org/overview/
[237] B. Trevizan, J. Chamby-Diaz, A. L. C. Bazzan, and M. Recamonde-Mendoza, "A comparative evaluation of aggregation methods for machine learning over vertically partitioned data," *Exp. Syst. Appl.*, vol. 152, Aug. 2020, Art. no. 113406.
[238] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.
[239] P. K. Chan et al., "Toward parallel and distributed learning by meta-learning," in *Proc. AAAI Workshop Knowl. Discovery Databases*, 1993, pp. 227–240.
[240] F. L. D. Silva, R. Glatt, and A. H. R. Costa, "Simultaneously learning and advising in multiagent reinforcement learning," in *Proc. AAMAS*, Paulo, Brazil, May 2017, pp. 1100–1108.
[241] U. F. Minhas, J. Zhang, T. Tran, and R. Cohen, "A multifaceted approach to modeling agent trust for effective communication in the application of mobile ad hoc vehicular networks," *IEEE Trans. Syst., Man, Cybern., C*, vol. 41, no. 3, pp. 407–420, May 2011.
[242] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro, "LOGAN: Membership inference attacks against generative models," 2017, *arXiv:1705.07663*.
[243] B. Hilprecht, M. Härterich, and D. Bernau, "Monte Carlo and reconstruction membership inference attacks against generative models," *Proc. Privacy Enhancing Technol.*, vol. 2019, no. 4, pp. 232–249, Oct. 2019.
[244] D. Chen, N. Yu, Y. Zhang, and M. Fritz, "GAN-leaks: A taxonomy of membership inference attacks against generative models," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2020, pp. 343–362.
[245] W. Ren, X. Zhou, J. Liu, and N. B. Shroff, "Multi-armed bandits with local differential privacy," 2020, *arXiv:2007.03121*.
[246] J. Zhao, "Distributed deep learning under differential privacy with the teacher–student paradigm," in *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI)*, New Orleans, LA, USA, Feb. 2018, pp. 404–408.
[247] F. Boenisch, C. Mühl, R. Rinberg, J. Ihrig, and A. Dziedzic, "Individualized PATE: Differentially private machine learning with individual privacy guarantees," 2022, *arXiv:2202.10517*.
[248] Y. Long et al., "G-PATE: Scalable differentially private data generator via private aggregation of teacher discriminators," 2019, *arXiv:1906.09338*.
[249] M. Abadi et al., "On the protection of private information in machine learning systems: Two recent approches," in *Proc. IEEE 30th Comput. Secur. Found. Symp. (CSF)*, Aug. 2017, pp. 1–6.
[250] X. Tang et al., "Value function is all you need: A unified learning framework for ride hailing platforms," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2021, pp. 3605–3615.
[251] Y. Wang, Y. Tong, C. Long, P. Xu, K. Xu, and W. Lv, "Adaptive dynamic bipartite graph matching: A reinforcement learning approach," in *Proc. IEEE 35th Int. Conf. Data Eng. (ICDE)*, Apr. 2019, pp. 1478–1489.
[252] T. Zhu, D. Ye, W. Wang, W. Zhou, and P. S. Yu, "More than privacy: Applying differential privacy in key areas of artificial intelligence," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 6, pp. 2824–2843, Jun. 2022.
[253] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," *ACM Comput. Surv.*, vol. 42, no. 4, pp. 1–53, Jun. 2010.
[254] Y. Cheng, Y. Liu, T. Chen, and Q. Yang, "Federated learning for privacy-preserving AI," *Commun. ACM*, vol. 63, no. 12, pp. 33–36, Nov. 2020.
[255] C. Ma et al., "When federated learning meets blockchain: A new distributed learning paradigm," *IEEE Comput. Intell. Mag.*, vol. 17, no. 3, pp. 26–33, Aug. 2022.
[256] H. Hu, Z. Salcic, L. Sun, G. Dobbie, P. S. Yu, and X. Zhang, "Membership inference attacks on machine learning: A survey," *ACM Comput. Surv.*, vol. 54, no. 11, pp. 1–37, Jan. 2022.
[257] M. Balunovic, D. I. Dimitrov, R. Staab, and M. T. Vechev, "Bayesian framework for gradient leakage," in *Proc. Int. Conf. Learn. Represent.*, 2022, pp. 1–16.
[258] Y. Dong et al., "Benchmarking adversarial robustness on image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 318–328.
[259] C. Xie, K. Huang, P.-Y. Chen, and B. Li, "DBA: Distributed backdoor attacks against federated learning," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–19.
[260] K. Wei, J. Li, M. Ding, C. Ma, Y.-S. Jeon, and H. V. Poor, "Covert model poisoning against federated learning: Algorithm design and optimization," 2021, *arXiv:2101.11799*.
[261] D. Zhong, H. Sun, J. Xu, N. Gong, and W. H. Wang, "Understanding disparate effects of membership inference attacks and their countermeasures," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, May 2022, pp. 959–974.
[262] H. Liu, J. Jia, W. Qu, and N. Z. Gong, "EncoderMI: Membership inference against pre-trained encoders in contrastive learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2021, pp. 2081–2095.
[263] G. Abad, S. Picek, V. J. Ramírez-Durán, and A. Urbieta, "On the security & privacy in federated learning," 2021, *arXiv:2112.05423*.
[264] E. Bao et al., "Skellam mixture mechanism: A novel approach to federated learning with differential privacy," *Proc. VLDB Endowment*, vol. 15, no. 11, pp. 2348–2360, Jul. 2022.
[265] W. Ruan, M. Xu, W. Fang, L. Wang, L. Wang, and W. Han, "Private, efficient, and accurate: Protecting models trained by multi-party learning with differential privacy," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2023, pp. 1926–1943.
[266] S. Chen and B. Li, "Towards optimal multi-modal federated learning on non-IID data with hierarchical gradient blending," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, May 2022, pp. 1469–1478.
[267] L. Zong, Q. Xie, J. Zhou, P. Wu, X. Zhang, and B. Xu, "FedCMR: Federated cross-modal retrieval,"

in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2021, pp. 1672–1676.
[268] B. Xiong, X. Yang, F. Qi, and C. Xu, "A unified framework for multi-modal federated learning," *Neurocomputing*, vol. 480, pp. 110–118, Apr. 2022.
[269] C. Wu, F. Wu, L. Lyu, Y. Huang, and X. Xie, "Communication-efficient federated learning via knowledge distillation," *Nature Commun.*, vol. 13, no. 1, p. 2032, Apr. 2022.
[270] X. Gong et al., "Ensemble attention distillation for privacy-preserving federated learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 15056–15066.
[271] L. Zhang, L. Shen, L. Ding, D. Tao, and L.-Y. Duan, "Fine-tuning global model via data-free knowledge distillation for non-IID federated learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10164–10173.
[272] J. Li et al., "Blockchain assisted decentralized federated learning (BLADE-FL): Performance analysis and resource allocation," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 10, pp. 2401–2415, Oct. 2022.
[273] D. C. Nguyen, S. Hosseinalipour, D. J. Love, P. N. Pathirana, and C. G. Brinton, "Latency optimization for blockchain-empowered federated learning in multi-server edge computing," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 12, pp. 3373–3390, Dec. 2022.
[274] X. Deng et al., "Blockchain assisted federated learning over wireless channels: Dynamic resource allocation and client scheduling," *IEEE Trans. Wireless Commun.*, vol. 22, no. 5, pp. 3537–3553, May 2023.
[275] L. Cui, X. Su, and Y. Zhou, "A fast blockchain-based federated learning framework with compressed communications," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 12, pp. 3358–3372, Dec. 2022.
[276] Q. Zhang et al., "AsySQN: Faster vertical federated learning algorithms with better computation resource utilization," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2021, pp. 3917–3927.
[277] F. Fu et al., "VF^{2}Boost: Very fast vertical federated gradient boosting for cross-enterprise learning," in *Proc. Int. Conf. Manag. Data*, Jun. 2021, pp. 563–576.
[278] K. Wei et al., "Vertical federated learning: Challenges, methodologies and experiments," 2022, *arXiv:2202.04309*.
[279] N. Agarwal, A. T. Suresh, F. X. Yu, S. Kumar, and B. McMahan, "CPSGD: Communication-efficient and differentially-private distributed SGD," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, Montreal, QC, Canada, Dec. 2018, pp. 7575–7586.
[280] L. Yu, L. Liu, C. Pu, M. E. Gursoy, and S. Truex, "Differentially private model publishing for deep learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, San Francisco, CA, USA, May 2019, pp. 332–349.
[281] Y. Nie, W. Yang, L. Huang, X. Xie, Z. Zhao, and S. Wang, "A utility-optimized framework for personalized private histogram estimation," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 4, pp. 655–669, Apr. 2019.
[282] X. Gu, M. Li, L. Xiong, and Y. Cao, "Providing input-discriminative protection for local differential privacy," in *Proc. IEEE 36th Int. Conf. Data Eng. (ICDE)*, Apr. 2020, pp. 505–516.
[283] Y. Tao, S. Chen, F. Li, D. Yu, J. Yu, and H. Sheng, "A distributed privacy-preserving learning dynamics in general social networks," 2020, *arXiv:2011.09845*.
[284] R. W. Baldwin and W. C. Gramlich, "Cryptographic protocol for trustable match making," in *Proc. IEEE Symp. Secur. Privacy*, Apr. 1985, p. 92.
[285] C. Hazay and Y. Lindell, "Constructions of truly practical secure protocols using standardsmartcards," in *Proc. 15th ACM Conf. Comput. Commun. Secur.*, Oct. 2008, pp. 491–500.
[286] C. Meadows, "A more efficient cryptographic matchmaking protocol for use in the absence of a continuously available third party," in *Proc. IEEE Symp. Secur. Privacy*, Apr. 1986, p. 134.
[287] M. J. Freedman, K. Nissim, and B. Pinkas, "Efficient private matching and set intersection," in *Proc. Adv. Cryptol. (EUROCRYPT)*, Berlin, Germany, 2004, pp. 1–19.
[288] Y. Huang, D. Evans, and J. Katz, "Private set intersection: Are garbled circuits better than custom protocols?" in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, 2012, pp. 1–15.
[289] C. Dong, L. Chen, and Z. Wen, "When private set intersection meets big data: An efficient and scalable protocol," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 789–800.
[290] J. Li and J. Li, "Brief paper-adaptive iterative learning control for consensus of multi-agent systems," *IET Control Theory Appl.*, vol. 7, no. 1, pp. 136–142, 2013.
[291] D. Meng and Y. Jia, "Finite-time consensus for multi-agent systems via terminal feedback iterative learning," *IET Control Theory Appl.*, vol. 5, no. 18, pp. 2098–2110, Dec. 2011.
[292] D. Meng and Y. Jia, "Iterative learning approaches to design finite-time consensus protocols for multi-agent systems," *Syst. Control Lett.*, vol. 61, no. 1, pp. 187–194, Jan. 2012.
[293] T. D. Nguyen, L. H. Pham, and J. Sun, "SGUARD: Towards fixing vulnerable smart contracts automatically," in *Proc. IEEE Symp. Secur. Privacy (SP)*, San Francisco, CA, USA, May 2021, pp. 1215–1229.
[294] D. C. Nguyen et al., "Federated learning meets blockchain in edge computing: Opportunities and challenges," *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12806–12825, Aug. 2021.
[295] S. Awan, F. Li, B. Luo, and M. Liu, "Poster: A reliable and accountable privacy-preserving federated learning framework using the blockchain," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2019, pp. 2561–2563.
[296] Y. Zhan and J. Zhang, "An incentive mechanism design for efficient edge learning by deep reinforcement learning approach," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Jul. 2020, pp. 2489–2498.
[297] R. H. L. Sim, Y. Zhang, M. C. Chan, and B. K. H. Low, "Collaborative machine learning with incentive-aware model rewards," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 119, Jul. 2020, pp. 8927–8936.
[298] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jul. 2021, pp. 6357–6368.
[299] I. Bistritz, T. Baharav, A. Leshem, and N. Bambos, "My fair bandit: Distributed learning of max-min fairness with multi-player bandits," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 119, Jul. 2020, pp. 930–940.
[300] L. Lyu, X. Xu, Q. Wang, and H. Yu, *Collaborative Fairness in Federated Learning*. Cham, Switzerland: Springer, 2020.
[301] H. Hasan et al., "Secure lightweight ECC-based protocol for multi-agent IoT systems," in *Proc. IEEE 13th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Rome, Italy, Oct. 2017, pp. 1–8.
[302] Q. Wu and R. Zhang, "Towards smart and reconfigurable environment: Intelligent reflecting surface aided wireless network," *IEEE Commun. Mag.*, vol. 58, no. 1, pp. 106–112, Jan. 2020.
[303] L. Deng, G. Li, S. Han, L. Shi, and Y. Xie, "Model compression and hardware acceleration for neural networks: A comprehensive survey," *Proc. IEEE*, vol. 108, no. 4, pp. 485–532, Apr. 2020.
[304] Y. Liu, Z. Ma, X. Liu, S. Ma, and K. Ren, "Privacy-preserving object detection for medical images with faster R-CNN," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 69–84, 2022.
[305] J. Chao et al., "CaRENets: Compact and resource-efficient CNN for homomorphic inference on encrypted medical images," 2019, *arXiv:1901.10074*.
[306] J. Feng, L. T. Yang, Q. Zhu, and K. R. Choo, "Privacy-preserving tensor decomposition over encrypted data in a federated cloud environment," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 4, pp. 857–868, Jul. 2020.
[307] R. Zhang and P. Venkitasubramaniam, "Optimal local differentially private quantization," *IEEE Trans. Signal Process.*, vol. 68, pp. 6509–6520, 2020.
[308] Z. Luo, D. J. Wu, E. Adeli, and L. Fei-Fei, "Scalable differential privacy with sparse network finetuning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5057–5066.
[309] D. Quarta, M. Pogliani, M. Polino, F. Maggi, A. M. Zanchettin, and S. Zanero, "An experimental security analysis of an industrial robot controller," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 268–286.
[310] B. Breiling, B. Dieber, and P. Schartner, "Secure communication for the robot operating system," in *Proc. Annu. IEEE Int. Syst. Conf. (SysCon)*, Apr. 2017, pp. 1–6.
[311] K. Jokinen and G. Wilcock, "Do you remember me? Ethical issues in long-term social robot interactions," in *Proc. 30th IEEE Int. Conf. Robot Human Interact. Commun. (RO-MAN)*, Aug. 2021, pp. 678–683.
[312] A. Sharkey and N. Sharkey, "Granny and the robots: Ethical issues in robot care for the elderly," *Ethics Inf. Technol.*, vol. 14, no. 1, pp. 27–40, Mar. 2012.
[313] D. Adams, A. Bah, C. Barwulor, N. Musaby, K. Pitkin, and E. M. Redmiles, "Ethics emerging: The story of privacy and security perceptions in virtual reality," in *Proc. 14th Symp. Usable Privacy Secur. (SOUPS)*, Baltimore, MD, USA, Aug. 2018, pp. 427–442.
[314] A. Gulhane et al., "Security, privacy and safety risk assessment for virtual reality learning environment applications," in *Proc. 16th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2019, pp. 1–9.
[315] D. Maloney, S. Zamanifard, and G. Freeman, "Anonymity vs. familiarity: Self-disclosure and privacy in social virtual reality," in *Proc. 26th ACM Symp. Virtual Reality Softw. Technol.*, Nov. 2020, pp. 1–9.
[316] M. Roetteler and K. M. Svore, "Quantum computing: Codebreaking and beyond," *IEEE Secur. Privacy*, vol. 16, no. 5, pp. 22–36, Sep. 2018.
[317] K. A. G. Fisher et al., "Quantum computing on encrypted data," *Nature Commun.*, vol. 5, no. 1, pp. 1–7, Jan. 2014.
[318] Q. Yang, Y. Zhao, H. Huang, Z. Xiong, J. Kang, and Z. Zheng, "Fusing blockchain and AI with metaverse: A survey," *IEEE Open J. Comput. Soc.*, vol. 3, pp. 122–136, 2022.
[319] Y. Jiang et al., "Reliable distributed computing for metaverse: A hierarchical game-theoretic approach," *IEEE Trans. Veh. Technol.*, vol. 72, no. 1, pp. 1084–1100, Jan. 2023.
[320] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Communication-efficient federated learning for digital twin edge networks in industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5709–5718, Aug. 2021.
[321] R. Dong, C. She, W. Hardjawana, Y. Li, and B. Vucetic, "Deep learning for hybrid 5G services in mobile edge computing systems: Learn from a digital twin," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4692–4707, Oct. 2019.
[322] E. Karaarslan and M. Babiker, "Digital twin security threats and countermeasures: An introduction," in *Proc. Int. Conf. Inf. Secur. Cryptol. (ISCTURKEY)*, Dec. 2021, pp. 7–11.
[323] C. Chen et al., "When digital economy meets Web3.0: Applications and challenges," *IEEE Open J. Comput. Soc.*, vol. 3, pp. 233–245, 2022.
[324] C. Huang, J. H. Lim, and A. C. Courville, "A variational perspective on diffusion-based generative models and score matching," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2021, pp. 22863–22876.

ABOUT THE AUTHORS

Chuan Ma (Member, IEEE) received the B.S. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2013, and the Ph.D. degree from the University of Sydney, Sydney, NSW, Australia, in 2018.

From 2018 to 2022, he worked as a Lecturer at the Nanjing University of Science and Technology, Nanjing, China. He is currently a Principal Investigator at the Zhejiang Laboratory, Hangzhou, China. He has published more than 40 journal and conference papers, including the best paper in IEEE Wireless Communications and Networking Conference (WCNC) 2018. His research interests include stochastic geometry, wireless caching networks, and distributed machine learning, and now focuses on big data analysis and privacy-preserving.

Dr. Ma received the Best Paper Award from the IEEE Signal Processing Society in 2022.

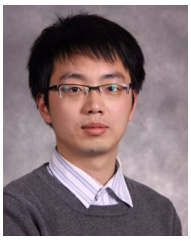
Jun Li (Senior Member, IEEE) received the Ph.D. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2009.

From January 2009 to June 2009, he worked as a Research Scientist with the Department of Research and Innovation, Alcatel-Lucent Shanghai Bell, Shanghai. From June 2009 to April 2012, he was a Postdoctoral Fellow at the School of Electrical Engineering and Telecommunications, University of New South Wales Sydney, Sydney, NSW, Australia. From April 2012 to June 2015, he was a Research Fellow at the School of Electrical Engineering, The University of Sydney, Sydney. He was a Visiting Professor at Princeton University, Princeton, NJ, USA, from 2018 to 2019. Since June 2015, he has been a Professor at the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China. He has coauthored more than 200 papers in IEEE journals and conferences and holds one U.S. patent and more than ten Chinese patents in his research areas. His research interests include network information theory, game theory, distributed intelligence, multiple agent reinforcement learning, and their applications in ultradense wireless networks, mobile edge computing, network privacy and security, and the industrial Internet of things.

Dr. Li is serving as an Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATION and a Wireless Communications and Networking Conference (TPC) member for several flagship IEEE conferences.

Kang Wei (Member, IEEE) received the B.S. degree in information engineering from Xidian University, Xi'an, China, in 2014, and the Ph.D. degree from the Nanjing University of Science and Technology, Nanjing, China, in 2023.

He is currently a Postdoctoral Fellow at The Hong Kong Polytechnic University, Hong Kong. He mainly focuses on privacy protection and optimization techniques for edge intelligence, including federated learning, differential privacy, and network resource allocation.



Bo Liu (Senior Member, IEEE) received the B.Eng. degree from the Department of Computer Science and Technology, Nanjing University of Posts and Telecommunications, Nanjing, China, in 2004, and the M.Eng. and Ph.D. degrees from the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2007 and 2010, respectively.

He is currently an Associate Professor at the University of Technology Sydney, Sydney, NSW, Australia. His research interests include cybersecurity and privacy, location privacy and image privacy, privacy protection, and machine learning.

Ming Ding (Senior Member, IEEE) received the B.S. and M.S. degrees (honors) in electronics engineering and the Ph.D. degree in signal and information processing from Shanghai Jiao Tong University (SJTU), Shanghai, China, in 2004, 2007, and 2011, respectively.

From April 2007 to September 2014, he worked at the Sharp Laboratories of China, Shanghai, as a Researcher/Senior Researcher/Principal Researcher. He is currently a Principal Research Scientist at Data61, CSIRO, Sydney, NSW, Australia. He has authored more than 200 papers in IEEE journals and conferences, all in recognized venues, and around 20 The 3rd Generation Partnership Project (3GPP) standardization contributions, as well as two books, i.e., *Multi-Point Cooperative Communication Systems: Theory and Applications* (Springer, 2013) and *Fundamentals of Ultra-Dense Wireless Networks* (Cambridge University Press, 2022). He holds 21 U.S. patents and has co-invented more than 100 patents on 4G/5G technologies. His research interests include information technology, data privacy and security, and machine learning and Artificial Intelligence (AI).

Dr. Ding has served as a guest editor/co-chair/co-tutor/Wireless Communications and Networking Conference (TPC) member for multiple IEEE top-tier journals/conferences and received several awards for his research work and professional services, including the prestigious IEEE Signal Processing Society Best Paper Award in 2022. He is an Editor of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and IEEE COMMUNICATIONS SURVEYS AND TUTORIALS.

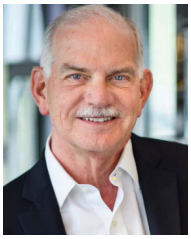
Long Yuan received the B.S. and M.S. degrees from Sichuan University, Chengdu, China, in 2010 and 2013, respectively, and the Ph.D. degree from the Database Group, University of New South Wales Sydney, Sydney, NSW, Australia, in 2017.

He is currently a Professor at the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. He has published papers in conferences and journals, including Very Large Data Base (VLDB), International Conference On Data Engineering (ICDE), International World Wide Web Conference (WWW), *The VLDB Journal*, and IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE). His research interests include graph data management and analysis.

Zhu Han (Fellow, IEEE) received the B.S. degree in electronic engineering from Tsinghua University, Beijing, China, in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, MD, USA, in 1999 and 2003, respectively.

From 2000 to 2002, he was a Research and Development Engineer of JDSU, Germantown, MD, USA. From 2003 to 2006, he was a Research Associate at the University of Maryland. From 2006 to 2008, he was an Assistant Professor at Boise State University, Boise, ID, USA. He is currently a John and Rebecca Moores Professor with the Electrical and Computer Engineering Department and the Computer Science Department, University of Houston, Houston, TX, USA. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid.

Dr. Han has been an American Association for the Advancement of Science (AAAS) Fellow and an Association for Computing Machinery (ACM) Distinguished Member since 2019. He received the National Science Foundation (NSF) Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the European Association For Signal Processing (EURASIP) Best Paper Award for the *Journal on Advances in Signal Processing* in 2015, the IEEE Leonard G. Abraham Prize in the field of communications systems [best paper award in IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (JSAC)] in 2016, and several best paper awards in IEEE conferences. He is also the winner of the 2021 IEEE Kiyo Tomiyasu Award, for outstanding early to mid-career contributions to technologies holding the promise of innovative applications, with the following citation: "for contributions to game theory and distributed management of autonomous communication networks." He was an IEEE Communications Society Distinguished Lecturer from 2015 to 2018. He has been a 1% Highly Cited Researcher since 2017 according to Web of Science.



H. Vincent Poor (Life Fellow, IEEE) received the Ph.D. degree in Electrical Engineering and Computer Science (EECS) from Princeton University, Princeton, NJ, USA, in 1977.

From 1977 to 1990, he was on the faculty of the University of Illinois at Urbana-Champaign, Urbana, IL, USA. Since 1990, he has been on the faculty at Princeton University, where he is currently a Michael Henry Strater University Professor. From 2006 to 2016, he served as the Dean of Princeton's School of Engineering and Applied Science. He has also held visiting appointments at several other universities, including most recently at Berkeley and Cambridge. His research interests include the areas of information theory, machine learning, and network science, and their applications in wireless networks, energy systems, and related fields. Among his publications in these areas is the recent book *Machine Learning and Wireless Communications* (Cambridge Univ. Press, 2022).

Dr. Poor is a member of the National Academy of Engineering and the National Academy of Sciences and a foreign member of the Chinese Academy of Sciences, the Royal Society, and other national and international academies. He received the IEEE Alexander Graham Bell Medal in 2017.