Front. Comput. Sci., 2022, 16(5): 165817
https://doi.org/10.1007/s11704-021-0598-z

REVIEW ARTICLE

Challenges and future directions of secure federated learning: a survey

Kaiyue ZHANG[1,2], Xuan SONG (✉)[3,4], Chenhan ZHANG[2], Shui YU (✉)[2]

1 Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China
2 Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney 2007, Australia
3 SUSTech-UTokyo Joint Research Center on Super Smart City, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China
4 Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China

Abstract Federated learning came into being with the increasing concern of privacy security, as people's sensitive information is being exposed under the era of big data. It is an algorithm that does not collect users' raw data, but aggregates model parameters from each client and therefore protects user's privacy. Nonetheless, due to the inherent distributed nature of federated learning, it is more vulnerable under attacks since users may upload malicious data to break down the federated learning server. In addition, some recent studies have shown that attackers can recover information merely from parameters. Hence, there is still lots of room to improve the current federated learning frameworks. In this survey, we give a brief review of the state-of-the-art federated learning techniques and detailedly discuss the improvement of federated learning. Several open issues and existing solutions in federated learning are discussed. We also point out the future research directions of federated learning.

Keywords federated learning, privacy protection, security

1 Introduction

In recent years, artificial intelligence has gradually entered into every aspect of people's life. The combination of modern deep learning algorithms and massive data makes the deep learning technology a promising tool to solve complicated real-world problems. However, with the emergence of more and more deep learning services leveraging the data, how to protect data privacy becomes a serious challenge [1]. Since artificial intelligence is supported by massive data, in the field of both the industry and academia, it is an emerging trend that large datasets are increasingly collected. Traditional deep learning fashions inevitably collect a large amount of data containing sensitive information for model training, and the training process is usually conducted in a centralized server. These factors render that privacy and security issues are likely to arise during the learning process.

To solve the privacy and security concerns, many researchers have proposed methods such as differential privacy [2], homomorphic encryption [3], and federated learning also came into being. The prototype of federated learning is first proposed by Google in 2016 [4]. According to Google, federated learning was first used on the google keyboard, mainly to protect users' private data [5] and to improve the language model quality [6]. Nevertheless, the concept of FL has been around for a long time, and at its core idea is the distributed deep learning, such as the privacy-protected deep learning system proposed by [7]. For federated learning, the key design to fulfill distributed learning is that it only requires parameters instead of collecting raw data from the user. With holding the data in each user's own equipment, the sensitive information is well protected. With this distinguished capability compared to other deep learning algorithms, federated learning attracts great attention from plenty of applications. Like wake word detection [8], emoji prediction [9], personalized model training [10], Internet of Things [11–13] and so on. Lim et al. [14] introduce a range of federated learning applications in different scenarios.

While federated learning has been used in most scenarios, researchers have found that there are still many challenges to be addressed. For example, a large amount of research work has realized that federated learning, originally intended to protect privacy, is more vulnerable to attacks by malicious nodes in many practical scenarios than traditional deep learning frameworks [15]. As only the parameters, which do not reveal the client's identity, are collected by a federated learning server, the anonymous clients may contain attackers that upload malicious data to the server. In this case, federated learning may be much less effective than traditional learning algorithms. Since these challenges limit the performance of

Received December 17, 2020; accepted March 31, 2021

E-mail: songx@sustech.edu.cn; shui.yu@uts.edu.au

federated learning, there are plenty of research works devoted to resolving these problems [16–19]. However, there is still room for discussion on these issues. Our research will focus on these challenges, aimed to propose novel applications with privacy-preserving federated learning in different scenarios.

In the remainder of this paper, we will review the developments of federated learning in Section 2. In Section 3, the challenges mentioned above of federated learning and the corresponding solutions will be introduced. Section 4 will discuss the future research direction of federated learning. The summary will be given in Section 5.

2 Secure learning algorithm: federated learning

In this section, we will first discuss about the learning algorithm that protects data privacy and security: federated learning, and review some applications. Then we will give a review about new challenges of federated learning and their corresponding solutions.

2.1 Emergence of federated learning

The reason for the birth of federated learning is the growing importance of data privacy. As security awareness increases, people will be more reluctant to contribute their own private data, which seriously impede the development of deep learning. Meanwhile, in real life, except for a few giant companies, most of the enterprises only have insufficient data with limited qualities, which are not enough to support the deployment of data-hungry AI services. From an enterprise's perspective, the data from commercial companies are often of great potential value. Different companies or even different departments within the same company usually do not share data. Accordingly, within the same company, the data is often in the form of isolated islands [20]. For individual users, most of their data also contain their personal information, such as travel trajectory, health status, etc. In this situation, it is unsafe to upload the unprotected raw data to the deep learning server.

Federated learning is essentially a distributed framework of the deep learning as shown in Fig 1. It can improve the effectiveness of the model through the model aggregation of multiple clients on the basis of ensuring the security of data privacy [21]. Therefore, federated learning can play a significant role in protecting the user's privacy since the training data cannot be collected centrally due to their wide distribution scope and involving sensitive privacy. Based on different data distributions among multiple participants, according to [20], federal learning can be divided into three categories: horizontal federated learning, vertical federated learning, and federated transfer learning.

The scenario of horizontal federated learning is one in which the data distribution provided by the participants is similar, but data providers do not overlap. In the training process, the model of each machine training is identical and complete, and can be independently predicted when making predictions. Consequently, this process can be regarded as a distributed training based on samples. Since each user's raw data are trained locally, and only local gradients parameters are shared and uploaded, the user's privacy is also protected but with some model loss. The adoption of this framework can be seen in [4,8–10].

The vertical federated learning scenario is opposite to the horizontal federated learning. The user sets are all the same, but different datasets have different kinds of data from these users. For example, airlines and hotels have different data from the same user, i.e., flight records and accommodation records. As a result, vertical federated learning requires sample alignment and model encryption [22]. During the training process, vertical federated learning ensures that other participants do not know the data and characteristics of the other party. In this way, the global model can obtain data information of all participants, and there is no loss of the model.

The application scenario of federated transfer learning is strict, which has a limited number of identical users, and very small datasets with the same features. A lot of recent work has narrowed down scenarios to very specific topics for discussion, such as wearable healthcare [23].

2.2 Bottleneck in federated learning

Plenty of applications take advantage of federated learning, and its concept has been deployed from many places. For instance, some applications allow users to train models on their mobile phones without having to upload raw data [24]. Nevertheless, though existing applications are diverse and

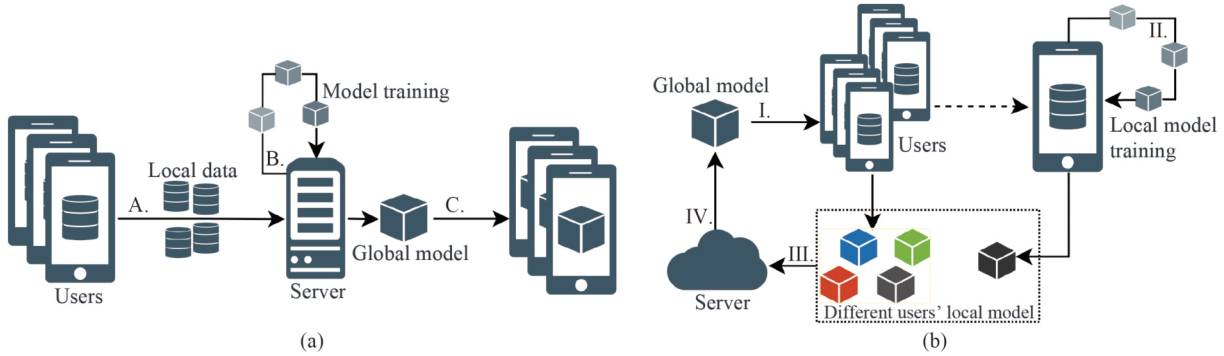


Fig. 1 Traditional deep learning versus federated learning. (a) Traditional deep learning. Step A: server collects data from users. Step B: server uses the whole dataset to train the model. Step C: server sends back the complete model to all the users; (b) Federated learning. Step I: server sends the global model to all the users. Step II: each user uses own data to train local model. Step III: each user sends their model to the server. Step IV: server aggregates models as a global model

successful, federated learning techniques still needs to be improved. The idea of distributed learning and keeping data locally makes it easier for malicious nodes to attack the federated learning framework. In addition, there are many other problems, such as many heterogeneous users and data themselves also bring more processing difficulty to the algorithm. Some of them are the same challenges as normal deep learning, and some are new challenges. Moreover, the challenges that exist in other classical algorithms may even have a more serious impact on federated learning frameworks. For example, if the dataset has only positive labels, it can directly break down the federated learning system [25].

In the followings, four challenges that future federated learning systems worth to be focused on will be discussed: high communication cost, systems heterogeneity, statistical heterogeneity, privacy concerns [19,20,26]. Additionally, some other vulnerabilities will be also involved in our discussion.

3 Challenges and corresponding solutions

Although federated learning is very promising and can be applied in most scenarios, as people continue to study, the challenging issues of federated learning are also exposed.

According to [27], model poisoning attacks are highly possible. Specifically, the federated learning process involves multiple clients uploading their own parameters, the central server receiving the local parameters, global aggregation, and finally returning the updated parameters to each client. Therefore, once malicious nodes are involved, they can misclassify the input with a high degree of confidence, which lead to model poisoning. Besides, Zhu et al. [28] pointed out the privacy issue, and they showed that even if each client uploads a local gradient instead of the original data, the malicious node still has a way to recover the contents of the raw data from the gradient. Meanwhile, in some scenarios, the data of different clients are not independent and identically distributed (i.e., non-IID), and clients' devices are also very different. The performance of federated learning in these cases is also worth to be explored [19].

3.1 Communication cost

Because sending raw data may cause privacy issues, the data generated on each device must be saved locally. This makes communication a bottleneck in federated learning. In a real world scenario, there can be millions of devices involved in the network, and each device may spend far less time training model locally than the network communication [29]. When there is an overlarge number of participants, although the model's quality will be improved since it is trained by more data, the communication overhead also increase. Especially, when mobile phones are the carriers of clients' data, the communication efficiency becomes much more decreased. This is because the local models are required to be uploaded to the server periodically and for large models, this step can be a bottleneck due to the limited bandwidth of wireless network. Also, in respect that connection speeds are asymmetric: uplink is usually slower than downlink, it is important to reduce the uplink communication cost [30]. To reduce communication cost, researchers should focus on two areas: first is to reduce the total number of communication rounds, and second is to reduce information size in each round of communication.

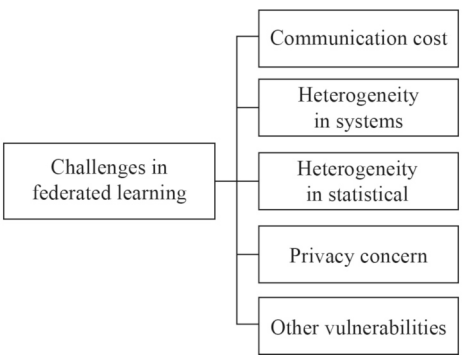


Fig. 2 Five challenges that we mainly discussed in this paper

3.2 Heterogeneity in systems

Due to the variety of people's devices, network status, storage, and processing capability of devices, the training process of computing and communication capabilities will be different. The presence of this heterogeneity exacerbates delayed mitigation and fault tolerance [31]. Bonawitz et al. [32] proposed a solution that is to filter a subset of valid devices from a cluster of devices. It is usually necessary to confirm whether the device is idle, the power status of the device, and whether it is a billing network. There may even be a situation where the device goes offline [33]. The heterogeneous nature of devices and networks, as well as the sudden loss of active members, will make people consider the issues of latency and fault tolerance. In a word, motivating user participation, handling heterogeneous devices, and designing fault-tolerant mechanism for the unstable network can contribute to solving the problem of the system heterogeneity in federated learning.

3.3 Heterogeneity in statistical

In addition to the above heterogeneity of the system, there is heterogeneity in the data itself. Due to the different generation and collection methods, the data from different users can easily be heterogeneous, in other words non-IID. Non-IID data will be more difficult to the process, which increases the complexity of modeling and evaluation. In particular, the federated learning usually adopts the stochastic gradient descent, which is widely used to train deep networks. The IID training data can better ensure that the stochastic gradient is unbiased [34]. Fortunately, there are some existed methods to deal with heterogeneous data, such as meta-learning which enables personalized modeling [18]. Sattler et al. [35] claimed that top-k sparsification performs very well in non-IID federated learning environments. In their work, they adopted top-k sparsification, designing a caching mechanism used on the server side and extended the compression to the downstream. Their results show that, in the worst case, their algorithm can still achieve an accuracy of at least 50% while the federated averaging algorithm does not even converge. Li et al. [36] demonstrated the convergence rate of federated averaging without assuming constraints. They also claimed that the learning rate will definitely decay if federated

averaging algorithm is adopted for handling non-IID data. Unlike previous analysis, they did not assume that the data in each client is IID, and it is hard to satisfy this assumption in practice.

3.4 Privacy concerns

Privacy is a top concern of federated learning. According to [37], most research of model attacks will assume that the attacker can barely access the model input, because the data for the training model remains internally confidential. However, when we consider the real world situations, we find that most service providers require users to upload private data to train. Once the uploading is over, users will lose control of their data: they can not know how the data is being used nor can they actively delete them. Federated learning has gone a long way toward protecting privacy by providing each user with local gradient information instead of raw data. Nevertheless, simply maintaining the localization of data during training does not provide sufficient privacy guarantees. Just passing on gradient information can still give away privacy to third parties or central servers [38]. From the service providers' perspective, even if they do a good job of protecting the user's raw dataset, the models themselves can give away private information. The model inversion attack is one of the ways that user's information can be implies from the model [39], which connects with the target by manipulating the relationship between unknown input and corresponding output. Zhu et al. [28] proposed an algorithm to recover the original image input from the user by obtaining the aggregation gradient returned by the central server to each client. In their algorithm, the malicious attacker will participate in learning process, and initialize a meaningless random graph. The attacker will train its random input to make its own local gradient as close to the global gradient returned by server as possible. In this way, they can recover the original image input of other users. Furthermore, Geiping et al. [40] proved that even the trained federated learning network, rather than the network during training, can faithfully be leveraged to reconstruct a high resolution image from the gradient parameters.

Shokri et al. [7] propose a privacy-protected deep learning system that allows multiple users to participate. Using local data, the participants first calculate the gradient of the neural network. Then, a partial gradient (e.g., 5%) must be sent to the parameter cloud server. They assume that the server is honest but curious. The system points out that the privacy can be perfectly protected without sharing the local gradient, but the model is completely inaccurate. On the other hand, sharing all of the local gradients would break some privacy, but it would give good accuracy. In order to balance the trade off, they share part of the local gradient, which is the main solution to maintain as little precision degradation as possible. The distributed selective SGD is the core of their idea. There are two methods to select parameters uploaded to the server. The first is sorting all the gradients from the largest to the smallest according to the absolute value, and to select the gradient of the first k to upload. The second is to randomly select a part of the parameter gradients whose absolute values are greater than the threshold value to upload. Other users can then download the gradient uploaded by user i to update their local parameters. The order in which a user uploads and downloads can be Round Robin, Random Order, and Asynchronous.

Aono et al. [22] further proved that in the system of [7], even small gradients stored on cloud servers can be exploited to infer user's information because local data can be extracted from these gradients, the results were unsatisfactory. Hence, they went a step further and proposed a new learning system that uses homomorphic encryption additionally to protect gradients on the honest-but-curious cloud server. All the gradients uploaded will be encrypted before stored on the cloud server. Users participating in the learning process jointly set public key p_k and key s_k to realize the addition homomorphic encryption scheme. The key s_k is secret to the cloud server, but is known to all learning participants. A separate TLS/SSL secure channel will be established between participants to communicate and protect the integrity of homomorphic cipher text. The cloud server is where the encrypted weight parameters are updated recursively, thanks to the addition operation of homomorphic encryption. This system can play a role of privacy protection without compromising the accuracy in deep learning models.

However, it is undeniable that homomorphic encryption will increase the communication cost to some extent. Naturally, there are other cryptographic methods that can be applied instead of homomorphic encryption. Bonawitz et al. [17] adopted secret sharing and double-masking protocol to resolve the challenges of federal learning. In their paper, they also consider the possibility of network fluctuations and proposed a mechanism to support users to quit during the training process. Users can also verify that the cloud server is operating correctly. In t-out-of-$\mathcal{N}$ secret sharing protocol, a secret will be divided into $\mathcal{N}$ disjoint pieces, and if someone gets the t pieces, he/she can recover the original secret. Therefore, in this system, a trusted authority will randomly create key pairs (N_n^{PK}, N_n^{SK}) and a random noise β_n for each user n. When a user wants to upload his/her local gradients, he/she will add some pieces of his/her secret key encrypted by other users' public key and some pieces of the β_n. All these additional information will be recovered by secret sharing protocol, or be canceled out as they are added up. Then, the server side receives the messages. Server will calculate the β_n and other information based on the secret sharing protocol, and finally calculate the aggregation gradients.

As a classical cryptography method, differential privacy can also be applied to federated learning. Geyer et al. [41] took advantage of the idea of differential privacy in their study. They suggest that customer involvement can also be hidden in federated learning while maintaining high model performance. Their proposed algorithm sets a threshold. If the probability that whether a piece of data is part of the training set exceeds the given threshold, the training will stop. The number of participating customers has a noticeable impact on the performance of the implemented model. Information is easier to hide when the number of participants is large. Similarly, Wei et al. [42] also applied differential privacy. Their proposed algorithm reduces the hamming distance between the gradient parameters calculated from two different datasets

with the same length by adding noise. Thus, the gradient parameters from different users cannot be traced back to their owners since neither the server nor the malicious user can distinguish them.

3.5 Other vulnerabilities

In addition to the above difficulties, the security of federated learning should also be emphasized. Although security and privacy are often considered equivalent intuitively, the difference actually exists. As mentioned above, privacy protection generally refers to the non-public exposure of sensitive personal information. Such information may be the user's health information, travel trajectories, salary level, etc. Although it may not directly expose the user's identity, it may enable others to indirectly identify specific individuals, thus exposing the user's sensitive information. The protection of security requires consideration of confidentiality, integrity, and availability. The challenges to security are generally the lack of access control of data and attacks from malicious nodes, which are usually launched by hackers for the system or model itself.

For example, model poisoning is one of the most common attacks on federated learning. Since there may be hundreds of clients participating in a federated learning network, there is no guarantee that there will be no malicious participants among these participants. In addition, generally speaking, defending against poisoning attacks require uploading data to the server for exception detection, which can compromise user privacy. The attacks, for examples, fake label attacks [43] and backdoor attacks [44], may lead to serious damage if the number of poisoned data is large.

The norm clipping techniques and differential privacy can defend model poisoning, while the overall performance will not be affected [45]. Since attackers generally want to have more impact on the model, they tend to produce updates with larger norms. Consequently, when the server selectively ignores those large updates, maybe those exceeding the threshold M, the attack of malicious nodes can be effectively defended. Meantime, they only add a small amount of Gaussian noise to defending against backdoor attacks. This kind of "weak" differential privacy can limit attacks while ensuring performance.

To enhance the privacy, Bittau et al. [46] were the first to propose PROCHLO implementation on the basis of differential privacy. They introduced a trusted third party to run a shuffler. The shuffler removes parts of the data that contain user privacy, such as timestamps, IP addresses, and so on, thus achieving the process of anonymity. However, simply deleting some metadata does not prevent an attacker from gaining access to the data owners. Malicious nodes may find out the link between data and users by monitoring network traffic and other methods. Therefore, the shuffler will do thresholding along with the shuffling. If some item classes have too few data, the shuffler will diacard them. Currently, the shuffle model with differential privacy has attracted more attention and been applied to prevent collusion attacks and poisoning attacks [47,48].

Ma et al. [49] noted that the security of federated learning also requires special attention. They conduct a large series of experiments discussing possible effective solutions on the well-known classification dataset: MNIST. According to the experimental results, we can draw many conclusions. If after each aggregation, the server can adjust the aggregation weight for each client based on the quality of the learning parameters uploaded by the client, the client will show better convergence rate and learning performance. In each learning epoch, the server collects the required client parameters with a fixed number before performing the next round of learning. That is to mitigate network fluctuations by dropping users, which can have a significant impact on the entire system. Once the waiting time exceeds a threshold period and the data is not fully collected, the current learning round will be abandoned. In addition, there is room for improvement in the parameter aggregation step. For example, we can add parameters tests on the server side and personalize aggregation weights for each parameter uploaded by the client based on different test performance results. The better performance the certain user brings, the higher weight factor that the user's parameters will get. At the same time, experimental results also have shown that increasing the local generation of each customer can also help improve model accuracy.

4 Future direction

4.1 Privacy and security protection

As mentioned above, although federated learning has been widely used, it still faces many challenges. Among issues, we believe that learning how to better protect users' privacy will be one of the most important points in the future development of federated learning. With the establishment of the EU general Data protection Regulations (GDPR) in 2018 [50], enterprises or website operators must pay more attention to the protection of data privacy. Hence, federated learning will have more room for development and there will be more application scenarios in the future. There's a lot of work going on in the healthcare and medical systems [23,51–54]. Recently, in the COVID-19 outbreak, some researchers have suggested that federated learning could better assist in the patient diagnosis while protecting the privacy of medical data [55,56].

4.2 Incentive mechanism for federated learning

In federated learning, due to network latency and communication overhead, data owners are likely to lose interest in participating in the learning system and no longer provide their data [57,58]. Meanwhile, when we require all users to frequently upload their local model parameters, the improvement to the global model may not be proportional to the communication cost. Accordingly, the incentive mechanisms for users to upload parameters also needs to be balanced [59]. Besides, it is also necessary to motivate high-quality users to participate in, and neglect or reject untrustworthy users at the same time, according to the uneven quality of data provided by users [60,61].

4.3 Personalized federated learning

Personalized service is much needed by users and has a broad prospect. On the one hand, many users prefer federated learning because they want to get a more personalized local

model to better serve themselves [62]. The Google keyboard mentioned many times before is an example of personalized federated learning. Users can train a prediction model that is more in line with their language habits while ensuring that the data is kept locally. The usual approach of model aggregation is naturally no longer applicable in this kind of problem. Many researchers have designed various model aggregation algorithms for personalized federated learning [63–65]. On the other hand, in the context of the Internet of Things, personalized federated learning can better mitigate the impact due to the heterogeneity of users' data. Mansour et al. [66] proposed that similar users can be clustered and then the model can be customized for each cluster. The idea of federated transfer learning also helps with personalization, different users relearn the parameters returned by the global model from their own local data [67].

5 Summary

This review focuses on the motivation of the emergence of federated learning models, its original concept, and its challenging issues discovered in recent research work and possible solutions. One reason for the advent of federated learning was that people found the existing assumptions of traditional deep learning too ideal. In the real world, the problem of data fragmentation and isolation is quite severe, and data providers are increasingly unwilling to expose their raw data and give out all control over it. Similarly, individual users do not want to fully expose their raw data because it contains a lot of sensitive information. Federated learning is proposed to solve these problems at first. In theory, federated learning greatly unites data from different isolation and breaks down barriers. At the same time, through federated learning, users can participate in deep learning without having to expose raw data, helping to train quality models while protecting privacy. However, there is still room for further improvement of this algorithm. Because of its distributed framework and very high degree of freedom of participation, it may attract more malicious attacks and other kinds of privacy leakage. Therefore, researchers hope to maximize the advantages of federated learning, protect the privacy of users, and train a more accurate model as the same time. The future research will also focus on the protection of privacy and security, incentive mechanisms, and personalized federal learning.

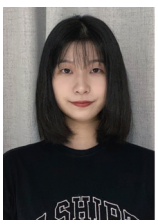
Acknowledgements This work was supported by Guangdong Provincial Key Laboratory (2020B121201001).

References

1. Shen S, Zhu T, Wu D, Wang W, Zhou W. From distributed machine learning to federated learning: in the view of data privacy and security. Concurrency and Computation: Practice and Experience, 2020, DOI: 10.1002/cpe.6002
2. Abadi M, Chu A, Goodfellow I, McMahan H B, Mironov I, Talwar K, Zhang L. Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. 2016, 308–318
3. Li P, Li J, Huang Z, Li T, Gao C Z, Yiu S M, Chen K. Multi-key privacy-preserving deep learning in cloud computing. Future Generation Computer Systems, 2017, 74: 76–85
4. McMahan B, Moore E, Ramage D, Hampson S, Arcas y B A. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of Artificial Intelligence and Statistics. 2017, 1273−1282
5. Yang T, Andrew G, Eichner H, Sun H, Li W, Kong N, Ramage D, Beaufays F. Applied federated learning: Improving google keyboard query suggestions. 2018, arXiv preprint arXiv: 1812.02903
6. Hard A, Rao K, Mathews R, Ramaswamy S, Beaufays F, Augenstein S, Eichner H, Kiddon C, Ramage D. Federated learning for mobile keyboard prediction. 2018, arXiv preprint arXiv: 1811.03604
7. Shokri R, Shmatikov V. Privacy-preserving deep learning. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. 2015, 1310−1321
8. Leroy D, Coucke A, Lavril T, Gisselbrecht T, Dureau J. Federated learning for keyword spotting. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing. 2019, 6341−6345
9. Ramaswamy S, Mathews R, Rao K, Beaufays F. Federated learning for emoji prediction in a mobile keyboard. 2019, arXiv preprint arXiv: 1906.04329
10. Fallah A, Mokhtari A, Ozdaglar A. Personalized federated learning with theoretical guarantees: a modelagnostic meta-learning approach. Advances in Neural Information Processing Systems, 2020: 33
11. Ye D, Yu R, Pan M, Han Z. Federated learning in vehicular edge computing: a selective model aggregation approach. IEEE Access, 2020, 8: 23920–23935
12. Lu Y, Huang X, Dai Y, Maharjan S, Zhang Y. Federated learning for data privacy preservation in vehicular cyber-physical systems. IEEE Network, 2020, 34(3): 50–56
13. Zhou C, Fu A, Yu S, Yang W, Wang H, Zhang Y. Privacy-preserving federated learning in fog computing. IEEE Internet of Things Journal, 2020, 7(11): 10782–10793
14. Lim W Y B, Luong N C, Hoang D T, Jiao Y, Liang Y C, Yang Q, Niyato D, Miao C. Federated learning in mobile edge networks: a comprehensive survey. IEEE Communications Surveys & Tutorials, 2020, 22(3): 2031–2063
15. Mothukuri V, Parizi R M, Pouriyeh S, Huang Y, Dehghantanha A, Srivastava G. A survey on security and privacy of federated learning. Future Generation Computer Systems, 2021, 115: 619–640
16. Fung C, Yoon C J, Beschastnikh I. Mitigating sybils in federated learning poisoning. 2018, arXiv preprint arXiv: 1808.04866
17. Bonawitz K, Ivanov V, Kreuter B, Marcedone A, McMahan H B, Patel S, Ramage D, Segal A, Seth K. Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017, 1175−1191
18. Zhao Y, Li M, Lai L, Suda N, Civin D, Chandra V. Federated learning with non-iid data. 2018, arXiv preprint arXiv: 1806.00582
19. Li T, Sahu A K, Talwalkar A, Smith V. Federated learning: challenges, methods, and future directions. IEEE Signal Processing Magazine, 2020, 37(3): 50–60
20. Yang Q, Liu Y, Chen T, Tong Y. Federated machine learning: concept and applications. ACM Transactions on Intelligent Systems and Technology, 2019, 10(2): 1–19
21. Nilsson A, Smith S, Ulm G, Gustavsson E, Jirstrand M. A performance evaluation of federated learning algorithms. In: Proceedings of the 2nd Workshop on Distributed Infrastructures for Deep Learning. 2018, 1–8

22. Aono Y, Hayashi T, Wang L, Moriai S, et al. Privacypreserving deep learning via additively homomorphic encryption. IEEE Transactions on Information Forensics and Security, 2017, 13(5): 1333–1345
23. Chen Y, Qin X, Wang J, Yu C, Gao W. Fedhealth: a federated transfer learning framework for wearable healthcare. IEEE Intelligent Systems, 2020, 35(4): 83–93
24. Wang X, Han Y, Wang C, Zhao Q, Chen X, Chen M. In-edge ai: Intelligentizing mobile edge computing, caching and communication by federated learning. IEEE Network, 2019, 33(5): 156–165
25. Yu F X, Rawat A S, Menon A K, Kumar S. Federated learning with only positive labels. 2020, arXiv preprint arXiv: 2004.10342
26. Kairouz P, McMahan H B, Avent B, Bellet A, Bennis M, Bhagoji A N, Bonawitz K, Charles Z, Cormode G, Cummings R, et al. Advances and open problems in federated learning. 2019, arXiv preprint arXiv: 1912.04977
27. Bhagoji A N, Chakraborty S, Mittal P, Calo S. Analyzing federated learning through an adversarial lens. In: Proceedings of International Conference on Machine Learning. 2019, 634–643
28. Zhu L, Liu Z, Han S. Deep leakage from gradients. Advances in Neural Information Processing Systems, 2019, 32: 14774–14784
29. Konečnỳ J, McMahan H B, Yu F X, Richtárik P, Suresh A T, Bacon D. Federated learning: strategies for improving communication efficiency. 2016, arXiv preprint arXiv: 1610.05492
30. Konečnỳ J, McMahan H B, Yu F X, Richtarik P, Suresh A T, Bacon D. Federated learning: strategies for improving communication efficiency. In: Proceedings of NIPS Workshop on Private Multi-Party Machine Learning. 2016
31. Li T, Sahu A K, Zaheer M, Sanjabi M, Talwalkar A, Smith V. Federated optimization in heterogeneous networks. 2018, arXiv preprint arXiv: 1812.06127
32. Bonawitz K, Eichner H, Grieskamp W, Huba D, Ingerman A, Ivanov V, Kiddon C, Konecny J, Mazzocchi S, McMahan H B, Van Overveldt T, Petrou D, Ramage D, Roselander J. Towards federated learning at scale: system design, 2019, arXiv preprint arXiv: 1902.01046
33. Kang J, Xiong Z, Niyato D, Zou Y, Zhang Y, Guizani M. Reliable federated learning for mobile networks. IEEE Wireless Communications, 2020, 27(2): 72–80
34. Rakhlin A, Shamir O, Sridharan K. Making gradient descent optimal for strongly convex stochastic optimization. In: Proceedings of the 29th International Coference on International Conference on Machine Learning. 2012, 1571−1578
35. Sattler F, Wiedemann S, Müller K R, Samek W. Robust and communication-efficient federated learning from non-iid data. IEEE Transactions on Neural Networks and Learning Systems, 2019, 31(9): 3400–3413
36. Li X, Huang K, Yang W, Wang S, Zhang Z. On the convergence of fedavg on non-iid data. 2019, arXiv preprint arXiv: 1907.02189
37. Ha T, Dang T K, Le H, Truong T A. Security and privacy issues in deep learning: a brief review. SN Computer Science, 2020, 1(5): 253
38. Truex S, Baracaldo N, Anwar A, Steinke T, Ludwig H, Zhang R, Zhou Y. A hybrid approach to privacypreserving federated learning. In: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security. 2019, 1–11
39. Fredrikson M, Jha S, Ristenpart T. Model inversion attacks that exploit confidence information and basic countermeasures. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. 2015, 1322−1333
40. Geiping J, Bauermeister H, Dröge H, Moeller M. Inverting gradients–how easy is it to break privacy in federated learning? 2020, arXiv preprint arXiv: 2003.14053
41. Geyer R C, Klein T, Nabi M. Differentially private federated learning: a client level perspective. 2017, arXiv preprint arXiv: 1712.07557
42. Wei K, Li J, Ding M, Ma C, Yang H H, Farokhi F, Jin S, Quek T Q, Poor H V. Federated learning with differential privacy: algorithms and performance analysis. IEEE Transactions on Information Forensics and Security, 2020, 15: 3454–3469
43. Biggio B, Nelson B, Laskov P. Poisoning attacks against support vector machines. In: Proceedings of the 29th International Coference on International Conference on Machine Learning. 2012, 1467−1474
44. Bagdasaryan E, Veit A, Hua Y, Estrin D, Shmatikov V. How to backdoor federated learning. In: Proceedings of International Conference on Artificial Intelligence. 2020, 2938−2948
45. Sun Z, Kairouz P, Suresh A T, McMahan H B. Can you really backdoor federated learning? 2019, arXiv preprint arXiv: 1911.07963
46. Bittau A, Erlingsson Ú, Maniatis P, Mironov I, Raghunathan A, Lie D, Rudominer M, Kode U, Tinnes J, Seefeld B. Prochlo: strong privacy for analytics in the crowd. In: Proceedings of the 26th Symposium on Operating Systems Principles. 2017, 441–459
47. Liu R, Cao Y, Chen H, Guo R, Yoshikawa M. Flame: differentially private federated learning in the shuffle model. 2020, arXiv preprint arXiv: 2009.08063
48. Wang T, Ding B, Xu M, Huang Z, Hong C, Zhou J, Li N, Jha S. Improving utility and security of the shuffler-based differential privacy. Proceedings of the VLDB Endowment, 2020, 13(13): 3545–3558
49. Ma C, Li J, Ding M, Yang H H, Shu F, Quek T Q, Poor H V. On safeguarding privacy and security in the framework of federated learning. IEEE Network, 2020, 34(4): 242–248
50. Goddard M. The eu general data protection regulation (GDPR): European regulation that has a global impact. International Journal of Market Research, 2017, 59(6): 703–705
51. Lim W Y B, Garg S, Xiong Z, Niyato D, Leung C, Miao C, Guizani M. Dynamic contract design for federated learning in smart healthcare applications. IEEE Internet of Things Journal, 2020, DOI: 10.1109/JIOT.2020.3033806
52. Brisimi T S, Chen R, Mela T, Olshevsky A, Paschalidis I C, Shi W. Federated learning of predictive models from federated electronic health records. International Journal of Medical Informatics, 2018, 112: 59–67
53. Silva S, Gutman B A, Romero E, Thompson P M, Altmann A, Lorenzi M. Federated learning in distributed medical databases: meta-analysis of large-scale subcortical brain data. In: Proceedings of IEEE 16th International Symposium on Biomedical Imaging. 2019, 270–274
54. Xu J, Glicksberg B S, Su C, Walker P, Bian J, Wang F. Federated learning for healthcare informatics. Journal of Healthcare Informatics Research, 2020, 5(1): 1–19
55. Kumar R, Khan A A, Zhang S, Wang W, Abuidris Y, Amin W, Kumar J. Blockchain-federated-learning and deep learning models for covid-19 detection using ct imaging. 2020, arXiv preprint arXiv: 2007.06537
56. Liu B, Yan B, Zhou Y, Yang Y, Zhang Y. Experiments of federated learning for covid-19 chest x-ray images. 2020, arXiv preprint arXiv: 2007.05592
57. Yu H, Liu Z, Liu Y, Chen T, Cong M, Weng X, Niyato D, Yang Q. A fairness-aware incentive scheme for federated learning. In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society. 2020, 393–399
58. Khan L U, Pandey S R, Tran N H, Saad W, Han Z, Nguyen M N, Hong C S. Federated learning for edge networks: resource optimization and incentive mechanism. IEEE Communications Magazine, 2020, 58(10): 88–93

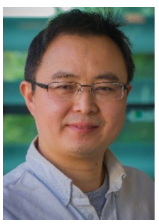
59. Pandey S R, Tran N H, Bennis M, Tun Y K, Manzoor A, Hong C S. A crowdsourcing framework for ondevice federated learning. IEEE Transactions on Wireless Communications, 2020, 19(5): 3241–3256
60. Kang J, Xiong Z, Niyato D, Xie S, Zhang J. Incentive mechanism for reliable federated learning: a joint optimization approach to combining reputation and contract theory. IEEE Internet of Things Journal, 2019, 6(6): 10700–10714
61. Weng J, Weng J, Zhang J, Li M, Zhang Y, Luo W. Deepchain: auditable and privacy-preserving deep learning with blockchain-based incentive. IEEE Transactions on Dependable and Secure Computing, 2019, 18(5): 2438–2455
62. Huang Y, Chu L, Zhou Z, Wang L, Liu J, Pei J, Zhang Y. Personalized federated learning: an attentive collaboration approach. 2020, arXiv preprint arXiv: 2007.03797
63. Dinh C T, Tran N, Nguyen T D. Personalized federated learning with moreau envelopes. Advances in Neural Information Processing Systems, 2020: 33
64. Deng Y, Kamani M M, Mahdavi M. Adaptive personalized federated learning. 2020, arXiv preprint arXiv: 2003.13461
65. Hu R, Guo Y, Li H, Pei Q, Gong Y. Personalized federated learning with differential privacy. IEEE Internet of Things Journal, 2020, 7(10): 9530–9539
66. Mansour Y, Mohri M, Ro J, Suresh A T. Three approaches for personalization with applications to federated learning. 2020, arXiv preprint arXiv: 2002.10619
67. Wang K, Mathews R, Kiddon C, Eichner H, Beaufays F, Ramage D. Federated evaluation of on-device personalization. 2019, arXiv preprint arXiv: 1910.10252



Kaiyue Zhang received the BE degree from the Department of Computer Science and Engineering, Southern University of Science and Technology, China in 2019. She is currently pursuing the PhD degree with the Faculty of Engineering and Information Technology, University of Technology Sydney, Australia, and the Department of Computer Science and Engineering, Southern University of Science and Technology. Her research interests include human mobility modeling, urban computing, privacypreserving mechanisms in deep learning.

Xuan Song received the PhD degree from Peking University, China in 2010. In 2017, he was selected as Excellent Young Researcher of Japan MEXT. He led and participated in many important projects as principal investigator or primary actor in Japan, such as DIAS/GRENE Grant of MEXT; Japan/US Big Data and Disaster Project of JST; Young Scientists Grant and Scientific Research Grant of MEXT; Research Grant of MLIT; Grant of JR EAST Company and Hitachi Company. He served as Associate Editor, Guest Editor, Program Chair, Area Chair, Program Committee Member or reviewer for many famous journals and top-tier conferences, such as IMWUT, WWW Journal, ACM TIST, IEEE TKDE, UbiComp, ICCV, CVPR, ICRA.

Chenhan Zhang received the BEng degrees in Telecommunication Engineering from University of Wollongong, Australia, and Zhengzhou University, China in 2017 and 2018, respectively. He received the MS degree in Engineering Management from City University of Hong Kong, China in 2019. He is currently a PhD student at Faculty of Engineering and Information Technology, University of Technology Sydney, Australia. His research interests include deep learning, intelligent transportation systems, privacy-preserving in AI.



Shui Yu obtained his PhD from Deakin University, Australia in 2004. He currently is a Professor of School of Computer Science, University of Technology Sydney, Australia. He has published three monographs and edited two books, more than 400 technical papers, including top journals and conferences, such as IEEE TPDS, TIFS, TMC, TKDE, ToN, and INFOCOM. Dr. Yu initiated the research field of networking for big data, and his research outputs have been widely adopted by industrial systems, such as Amazon cloud security. He is currently serving a number of prestigious editorial boards, including IEEE Communications Surveys and Tutorials (Area Editor), IEEE Communications Magazine, and so on.