SAFE FEDERATED CLOUD LEARNING: PREVENTING PORTABLE DIAGNOSTIC POISONING

#1SABBIDI ASHWITHA,

#2Dr. P.VENKATESHWARLU, *Assistant Professor,*

#3Dr.V.BAPUJI, *Associate Professor& HOD,*

Department of Master of Computer Applications,
VAAGESWARI COLLEGE OF ENGINEERING, KARIMNAGAR, TELANGANA

Abstract Federated learning arose in response to an increase in worry about privacy security in the age of big data, when people's sensitive information is exposed. It is an algorithm that collects model parameters from each client rather than raw data from users, so maintaining the privacy of the users. However, because federated learning is decentralized, it is more vulnerable to assaults, as users may contribute malicious data in order to bring down the federated learning server. Furthermore, a new study has showed that attackers can simply change parameters to recover data. As a result, there is a lot of room for improvement in the present federated learning frameworks. In this survey, we provide a quick overview of current federated learning approaches before delving into strategies to improve federated learning. Several open issues and existing federated learning systems are highlighted. In addition, we identify promising research topics in federated learning.

Keywords federated learning, privacy protection, security

1. INTRODUCTION

Artificial intelligence has been progressively seeping into every facet of human life in recent years. Deep learning technology is a promising tool for tackling difficult real-world problems because it combines state-of-the-art deep learning algorithms with huge volumes of data. Privacy protection is getting harder [1] as more and more deep learning services start using data. A growing number of huge datasets are being collected by both industry and academia as support for artificial intelligence. Training models using traditional deep learning methods always requires collecting a large quantity of data that may contain private information, and this process is typically executed on a centralized server. These features make it likely that learning-related privacy and security issues may arise.

Differential privacy [2], homomorphic encryption [3], and federated learning are only some of the answers presented by researchers to privacy and security issues. To demonstrate the potential of federated learning, Google presented a prototype in 2016 [4]. To protect user privacy [5] and enhance language model quality [6], Google claims that the Google Keyboard was the first implementation of federated learning. However, the fundamental idea of FL has always revolved around distributed deep learning methods like the privacy-protected deep learning system described in [7]. The fundamental benefit of federated learning for distributed learning is that it requires simply parameters and not the user's raw input. Sensitive information is effectively shielded when kept locally on each user's device. Because of its many advantages over competing deep learning methods, federated learning has found widespread use in a variety of contexts. The identification of wake words [8], the prediction of emojis [9], the development of individually tailored models [10], the Internet of Things [11–13], etc. The federated learning applications introduced by Lim et al. [14] can be used in a wide range of situations.

Despite widespread adoption of federated learning, researchers have identified a few problems that have yet to be fixed. Even though federated learning was developed to safeguard user privacy, multiple studies have demonstrated that it is more susceptible to assaults from malicious nodes than typical deep learning frameworks [15]. Since a federated learning server collects only the parameters, which do not expose the client's identity, an attacker could submit malicious data to the server using an anonymous client. It's possible that federated learning would fare much worse here than more traditional approaches to AI-assisted learning. Due of these challenges, the success of these concerns in federated learning are the focus of a lot of studies [16–19]. Concerning these matters, further discussion is warranted. We plan to address these concerns by designing privacy-preserving, context-aware, federated-learning applications.

2. SECURE LEARNING ALGORITHM: FEDERATED LEARNING

In the following section, we'll take a closer look at federated learning, a style of instruction that protects students' personal information, before moving on to the investigation of more specific applications. We will next move on to a discussion of the problems currently plaguing federated learning and an investigation of possible solutions.The development of federated education

Protecting personal information is becoming increasingly important, which has contributed to the rise of federated learning. Limits to the growth of deep learning could be imposed by people's decreasing willingness to give personal data as security awareness rises. In contrast, most companies operating in real-world contexts have insufficient and subpar data to back up the introduction of AI services that rely largely on data. When analyzed from an enterprise's perspective, the data it collects often reveals substantial value. Most information is not shared across companies or even between different divisions of the same company. As a result, information within a single company can take the form of many entities [20]. Users' data typically consists of sensitive information about their lives, such as where they go and how they feel about their health. In this situation, there is a possibility that unprocessed, unencrypted data will be transferred to the deep learning server. Figure 1 depicts a federated learning setup, a networked framework for implementing deep learning. By combining data from multiple users while keeping their anonymity intact, the model's performance can be improved [21]. Due to the dispersed and distributed nature of training data, centralized data gathering is impractical, making federated learning a critical tool for protecting user privacy. According to [20], there are three types of federal learning that can be distinguished by how information is shared among many people. Different types of federated learning include those that are horizontal, vertical, and lateral.

Participants in a horizontal federated learning environment provide data that is uniformly distributed but does not come from the same data sources. By the end of the process, it is clear that the training models for each machine are sufficiently similar and thorough to provide autonomous prediction at the prediction stage. Thus, you may hear this approach referred to as distributed training by samples. While anonymity for the user is still protected, the performance of the model noticeably drops. To do this, users' raw data is trained independently, and only the users' local gradient parameters are shared or uploaded. The user has entered a numerical range, which includes the digits 4, 8, and 9. Give some examples of how this theoretical framework has been put into practice.

The horizontal federated learning scenario is distinguished from the vertical one by... The user sets are consistent, however the types of information collected from these individuals change from one dataset to the next. Hotels and airlines, for instance, keep track of separate personal information files, one for the guest's lodging history and the other for his or her flights.

Sample alignment and model encryption [22] are two methods that can be used to ease the burden of vertical federated learning. By keeping each party in the training process in the dark about the skills and traits held by the others, vertical federated learning ensures that participant privacy is maintained. Using this method, the global model may efficiently collect data from all participants with little to no reduction in model accuracy.

A small number of users with similar characteristics and tiny datasets with similar qualities characterize the environment of the federated transfer learning application. Recent academic studies have zeroed in on certain niches, including the application of wearable healthcare technologies [23], to discuss and analyze.In the context of federated learning, a bottleneck is a limiting factor that prevents the process from operating at its full potential.

Numerous businesses have successfully used federated learning because of its wide range of useful applications. Numerous apps let people train models on the go without sharing sensitive information [24]. Even though there are a lot of apps out there,

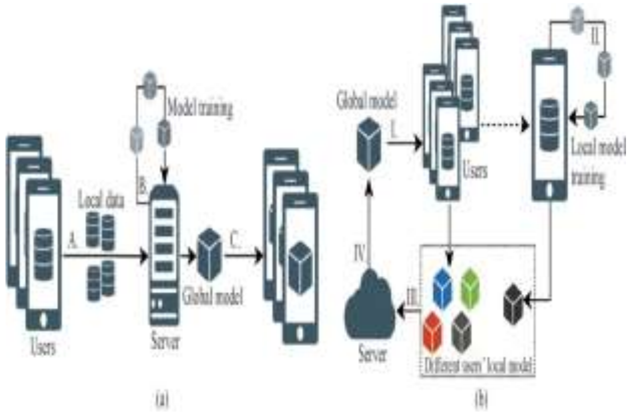

Fig. 1 The data from the users is transferred to the computer..

Step A: The model is trained in its entirety on the server.

Step B: When the server distributes the entire model to all clients, we have federated learning.

Step C: The server-side model of the world is downloaded by all users. Step I: server sends the global model to all the users.

Step II: A user's personal data is used to train a local model.

Step III: Everyone submits their model to the server.

Step IV: server aggregates models as a global model

Model hosted on a server There is room for development in federated learning systems, despite the model's widespread popularity. Since distributed learning and locally stored data are concepts, the federated learning system is more vulnerable to attacks from malicious nodes. The algorithm's processing time is long because of the many users and the complexity of the data. Some are brand-new to deep learning, while others are familiar to the field as a whole. Furthermore, federated learning systems may be harmed considerably more by the issues that plague other common algorithms. The federated learning system can be easily dismantled [25] if, for instance, the dataset only contains "yes" marks.

High communication costs, diverse systems, different statistics, and privacy concerns are just four of the issues that future federated learning systems will address [19,20,26]. We'll be discussing further issues during this gathering.

3. CHALLENGES AND CORRESPONDING SOLUTIONS

Although federated learning shows great promise and has wide applicability, ongoing research has revealed the existence of challenging issues associated with this approach.

According to the findings of [27], the occurrence of model poisoning incidents is highly probable. The federated learning process consists of several stages: several clients independently uploading their respective parameters, the central server receiving and aggregating the local values, and finally distributing the updated parameters back to each client. Consequently, when malicious nodes are present, they exhibit a significant level of certainty in incorrectly categorizing the input, thereby leading to model poisoning. Zhu et al. (28) also raised the issue of privacy, illustrating that even when clients upload local gradients instead of actual data, a malicious node can still use the gradients to reconstitute the original data. During the ongoing process, there are certain cases in which the devices used by clients exhibit significant variation, and the data from various clients is not distributed in an independent and identical manner, meaning it is non-IID. Furthermore, it is of significance to explore the efficacy of federated learning in certain contexts [19].

Communication cost

In order to safeguard user privacy, it is imperative to store the data generated on individual devices locally, as the transmission of raw data poses potential risks. Consequently, federated learning encounters a bottleneck in communication. In a practical situation, a network may accommodate a vast number of devices, perhaps reaching millions. In such cases, it is possible that each device will allocate significantly less time to train a model locally compared to the time spent on network communication [29]. The enhancement in the model's quality resulting from increased training data is counterbalanced by the escalation of communication overhead when there is an excessive number of participants. The efficacy of communication is significantly diminished, particularly when clients' data is transmitted over mobile devices. This phenomenon occurs because the local models need to be regularly transferred to the server, and for models of significant size, this process might become a bottleneck due to the limited capacity of the wireless network. Furthermore, it is imperative to reduce the expenses associated with uplink transmission because to the inherent asymmetry in connection speeds, where the uplink is generally slower than the downlink [30]. To achieve cost reduction in communication, researchers should prioritize two key areas: minimizing the total number of communication rounds and minimizing the volume of information transmitted in each round.

Heterogeneity in systems

The training approach for computer and communication skills may vary based on factors such as the diversity of individuals' devices, the status of their network connections, and the storage and processing capability of their devices. The presence of heterogeneity poses challenges in implementing delayed mitigation and fault tolerance [31]. Bonawitz et al. (32) proposed a potential solution, which involves the filtration of a subset of legitimate devices from a larger cluster of devices. It is often important to ascertain the power condition of the device, determine if it is connected to a billing network, and ascertain if it is in an idle state. There is a potential for the gadget to go offline [33]. The heterogeneous nature of devices and networks, as well as the potential loss of active members, give rise to concerns regarding latency and fault tolerance that individuals may consider. To address the issue of system heterogeneity in federated learning, it can be partially resolved through the promotion of user participation, effective management of diverse devices, and the implementation of fault-tolerant techniques to mitigate the impact of an unstable network.

Heterogeneity in statistical

The system exhibits heterogeneity, which is further compounded by the heterogeneity of the data. The heterogeneity, or non-IID nature, of the data can arise due to diverse procedures employed for generation and collection by different users. The handling of non-independent and identically distributed (non-IID) data presents increased difficulties, hence rendering the tasks of modeling and evaluation more intricate. Stochastic gradient descent (SGD) is a commonly utilized optimization algorithm in the context of federated learning for the purpose of training deep neural networks. The impartiality of the stochastic gradient can be enhanced through the utilization of independent and identically distributed (IID) training data [34]. One approach that has been developed to address the challenge of heterogeneous data is meta-learning, which enables the creation of personalized models [18]. Sattler et al. [35] have demonstrated that top-k sparsification exhibits outstanding performance in non-IID scenarios.

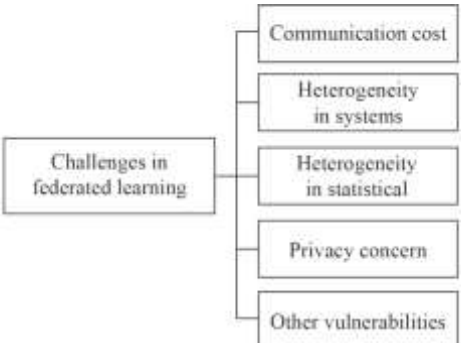

Fig. 2 This study primarily addressed five key concerns pertaining to federated learning environments. The researchers employed the technique of top-k sparsification, developed a caching system on the server-side, and expanded the compression method to the downstream in their investigation. The results of their study indicate that the federated averaging strategy does not achieve convergence, whereas their algorithm is capable of achieving a minimum accuracy of 50% even in the most unfavorable circumstances. Li et al. (36) demonstrated the pace of convergence of federated averaging without making any presumptions regarding its restrictions. Furthermore, it has been said that the utilization of the federated averaging technique for managing non-IID data results in a decrease in the learning rate when user i uploads the gradient to update their local data. In contrast to previous assessments, the researchers did not rely on the challenging assumption that the data from each client is independently and identically distributed (IID).

Privacy concerns

The primary focus of federated learning revolves around safeguarding privacy. According to [37], a significant portion of research on model attacks operates under the assumption that the attacker's access to input data is restricted due to the internal privacy measures employed during model training. Nevertheless, while examining real-world occurrences, it becomes apparent that a significant proportion of service providers require consumers to upload sensitive data for training purposes. Once the uploading process is finalized, users will relinquish control over their data, rendering them unable to actively erase the data or learning has ac in privacy pr users with lo f raw data. H localization du ite to ensure ad ing gradient da entities has the potential to violate the confidentiality of personal information [38]. Despite the service providers' diligent efforts to protect the user's raw dataset, the models utilized in the process may nevertheless possess the capability to disclose sensitive information. A technique commonly employed to infer user knowledge from a model [39] is the model inversion attack, which involves altering the association between an unknown input and its corresponding output. Zhu et al. (28) proposed a methodology for recovering the user's initial image input by gathering the aggregation gradient transmitted to each client from the primary server. The malevolent assailant will participate in the algorithmic learning procedure and commence by generating a random, inconsequential graph. The attacker's arbitrary input will undergo training in order to generate a local gradient that closely resembles the global gradient of the server. The original image input can be retrieved from other users using this approach. Geiping et al. [40] shown that the order of a user's uploads and downloads may be categorized into three modes: Round Robin, Random Order, or Asynchronous.

4. CONCLUSION

The primary focus of federated learning revolves around the preservation of privacy. According to reference [37], a significant portion of studies on model attacks operate under the assumption that attackers possess restricted access to input data. This assumption is based on the premise that the information used for training the model is maintained as internal and private. Nevertheless, by examining real-world occurrences, it becomes apparent that a significant portion of service providers require consumers to upload sensitive data for training purposes. Once the uploading process is finalized, users will no longer retain authority over their data. They will be unable to actively remove their data or ascertain the manner in which it is being utilized. Federated learning has made notable advancements in safeguarding privacy by providing individual users with local gradient information instead of raw data. However, only maintaining data localization during the training process is inadequate in ensuring adequate privacy. The act of transmitting gradient data to external systems or third-party entities has the potential to breach privacy [38]. Despite the service providers' diligent efforts in protecting the user's raw dataset, there is still a possibility that the models themselves can inadvertently disclose sensitive information. A technique that can be employed to infer user knowledge from a model [39] is the model inversion attack, which involves altering the association between an unknown input and its corresponding output. Zhu et al. (28) proposed a methodology for retrieving the original picture input of the user. This involves collecting the aggregation gradient that was transmitted to each client by the main server. The malevolent assailant will participate in the algorithmic learning procedure and commence by generating a random, inconsequential graph. The random input of the attacker will undergo training in order to generate a local gradient that closely resembles the global gradient of the server. The original image input can be retrieved from other users using this approach. Geiping et al. [40] have established that the order of a user's uploads and downloads may be categorized into three modes: Round Robin, Random Order, and Asynchronous.

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

22. evaluation of federated learning algorithms. In: Proceedings of the 2nd Workshop on Distributed Infrastructures for Deep Learning. 2018, 1–8

23. 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

24. 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

25. 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

26. Yu F X, Rawat A S, Menon A K, Kumar S. Federated learning with only positive labels. 2020, arXiv preprint arXiv: 2004.10342

27. 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

28. 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

29. Zhu L, Liu Z, Han S. Deep leakage from gradients. Advances in Neural Information Processing Systems, 2019, 32: 14774–14784

30. 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

31. 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

32. Li T, Sahu A K, Zaheer M, Sanjabi M, Talwalkar A, Smith V. Federated optimization in heterogeneous networks. 2018, arXiv preprint arXiv: 1812.06127

33. 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

34. Kang J, Xiong Z, Niyato D, Zou Y, Zhang Y, Guizani M. Reliable federated learning for mobile networks. IEEE Wireless Communi- cations, 2020, 27(2): 72–80

35. 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

36. 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

37. 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

38. 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

39. Truex S, Baracaldo N, Anwar A, Steinke T, Ludwig H, Zhang R, Zhou

40. Y. A hybrid approach to privacypreserving federated learning. In: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security. 2019, 1–11

41. 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