Federated Learning

Liam Glennie

April 16, 2023

Abstract

With the exponential growth of data production, finding ways to effectively utilise increasingly large datasets has become a major challenge. Federated Learning (FL) has emerged as a promising solution that enables decentralised data to be used for model training without exchanging data. In this review paper, we provide a comprehensive explanation of FL, including its different types, potential use cases, advantages, and limitations. We also highlight the security concerns associated with FL, such as vulnerabilities that could compromise the global model or expose the training data. Finally, we conclude that there is still a great deal of scope for future work to be done and applied in practice.

1 Introduction

With the exponential growth of data production, the size of datasets has become increasingly large, leading to a growing interest in finding ways to exploit this data effectively. Federated Learning (FL) is a machine learning technique that has gained popularity in recent years as a means of utilising decentralised data to train models without exchanging data. This approach has proven particularly relevant in areas such as healthcare, IoT, and finance, where data privacy and security are paramount concerns. In this review paper, we provide a comprehensive explanation of FL, its different types, possible use cases and security concerns. Our aim is to provide readers with a thorough understanding of FL, including its advantages and limitations.

2 Federated Learning

Federated Learning is a novel machine learning technique that has gained significant attention in recent years due to its potential for exploiting decentralised data from different owners to train models without exchanging the data. This approach was first introduced by Google in 2016 [7], where they attempted to use mobile devices to train a shared model using local data.

The main idea of FL is to perform a series of rounds, consisting of the following steps:

1. **Client Selection**: A subset of clients (mobile devices) is selected from the client pool to participate in the current round.
2. **Model Download**: The selected clients download the current model from the server.
3. **Local Training**: Each client trains the model using their local data.
4. **Update Transmission**: After training, clients send model updates to the server. There are two proposed update methods:
   - (a) **Structured Update**: This method involves directly learning an update on a restricted space that can be reduced to use a smaller number of variables.
   - (b) **Sketched Update**: In this method, a full model update is learned and compressed before it is sent to the server.
5. **Model Aggregation**: The server aggregates the updates to build an improved shared model. There are several aggregation methods that can be used, including Federated Averaging, and Secure Multiparty Computation Average, among others. [8] [1]

By using this method, a model can be trained without ever seeing the raw data. However, there are several challenges in terms of privacy concerns and security issues that will later be discussed.

It is important to differentiate between Federated Learning and distributed learning. Both aim to distribute the training of a global model across multiple devices or servers, but FL specifically prioritises data privacy by assuring that raw data is never shared between devices. Distributed learning works under the assumption that all the data is accessible to all of the devices. Ultimately, the choice between FL and DL depends on the specific needs and priorities of the project.

In the following sections, we will review the different forms FL can take according to network topology, data partition, data availability and aggregation algorithms.

2.1 Network Topology

In this section, we will discuss the underlying architecture of FL. Depending on the network topology, FL can be categorised as centralised, clustered or decentralised.

2.1.1 Centralised FL

Although Federated Learning is a decentralised approach to data, it still relies on a central server to host the global model, distribute it to clients, collect model updates, aggregate them, and send the updated model back to clients. Usually, this central server belongs to a third party trusted by both clients. This sets it apart from traditional centralised server-based methods, where data is hosted and models are trained on shared data within the same server. Currently, this is the most commonly used approach in practical applications of Federated Learning.

2.1.2 Clustered FL

Clustered FL is an improvement on centralised FL that offers greater efficiency. It capitalises on the variability in client data and groups clients with similar data distributions into clusters. By doing so, local training can take place within each cluster, leading to faster convergence of the global model and reduced communication with the centralised server. Additionally, this method provides enhanced privacy guarantees as local updates are shared only within clusters. Clustered FL could be of interest in the case of developing a language translation model. It could be interesting to cluster together clients with data in the same language.

2.1.3 Decentralised FL

Decentralised Federated Learning [6] is a promising approach to address the challenges of centralised FL. In centralised FL, a single server hosts the global model, which poses a risk of a single point of failure and scalability issues when the number of clients increases. On the other hand, decentralised FL utilises algorithms that replace the centralised server with a peer-to-peer approach. In this manner, clients train their local models based on their own data and communicate model updates with their neighbouring clients. This way, the burden of model aggregation and communication is distributed among the clients, reducing the reliance on a central server and enabling the training of the global model in a distributed manner.

However, decentralised FL comes with its own issues regarding communication and synchronisation problems, as also privacy and security between clients.

2.2 Data Partition

Federated Learning is trained using datasets, and these datasets are formed by a number of rows containing an ID and features. Depending on the partition of the data, we can determine which type of FL would be of interest.

2.2.1 Vertical FL

Vertical FL, also known as feature-based FL, can be utilised when different datasets hold varying information about the same set of entities, such as IDs. A practical example can be observed in the healthcare industry, where sensitive patient data cannot be shared due to privacy concerns. Suppose a healthcare company wants to develop a predictive model for identifying patients at risk of developing a particular disease, but they only have access to medical and demographic information of the patients which they cannot share with other organisations. However, there exists a research centre that has genetic information of the same patients. With vertical FL, the healthcare company and the research centre can collaborate without sharing any raw sensitive data to develop a predictive model to accurately predict the risk of patients developing the disease. Each entity trains a model with its own data. Then, a joint model is created by combining the models developed by each entity. To ensure privacy, this can be done using homomorphic encryption [11]. Finally, the joint model is shared amongst the entities and can be used to make predictions using their data. By having access to more varied data during training, it is likely that the federated model be more robust and accurate than a centralised model only using medical data or genetic data.

2.2.2 Horizontal FL

Horizontal FL is a technique that proves useful when two entities possess data that belong to the same feature space but with different instances or IDs. Google has implemented this idea of Federated Learning with a model that improves next-word prediction on their Gboard keyboard [5]. The process involves a global model that is trained using a federation of mobile devices. The server only receives model updates, and no personal raw data is shared between the clients and the server, thus maintaining the privacy of the users' data. This approach enables Google to take advantage of the collective knowledge of its users without compromising their privacy.

2.2.3 Federated Transfer Learning

Transfer Learning is a machine learning technique that departs from the traditional approach of training a model on data from the same domain in which it will be used. Instead, Transfer Learning takes advantage of the knowledge and features learned by a pre-trained model and adapts it to a different and related task. [10]

One of the main advantages of Transfer Learning is its ability to solve problems with small datasets. In this context, a pre-trained model can serve as a starting point for the new task, reducing considerably the amount of data needed to train a new model.

For instance, consider a scenario where a dog owner wants to develop a model to control an automatic doggie door, allowing only their own dog to enter and no other dogs or animals. To achieve this, the owner can use a pre-trained model, such as VGG-16 from the VGG family [12], and fine-tune it using a small dataset of images of their own dog. By adjusting the model's weights, the owner can train a new model that will accurately detect when their dog is at the door, and keep other animals out.

Federated Transfer Learning can be applied when two entities share a very small amount of data. For example, a bank in China and a commercial company in the USA could share a small subset of individuals or features. Both companies could be interested in improving their fraud detection systems. Therefore, they could implement a federated transfer learning model. First, both companies would independently develop their models using their local data, the models would then be combined to form a global model. This global model would be fine-tuned using the subset of data that coincides in both entities.

Federated Transfer Learning is a technique that can be utilised in scenarios where two entities have only a limited intersection of data. An example of such a scenario is where a bank located in China and a commercial company in the USA share only a small subset of individuals or features in their respective datasets. Both entities may be interested in enhancing their fraud detection systems, and Federated Transfer Learning can be a suitable approach to achieving this.

To implement this approach, the entities would independently develop their models using their local data. These models would then be combined to form a global model through Federated Learning. Afterwards, the global model would be fine-tuned using the subset of data that coincides with both entities.

2.3 Data Availability

We can categorise Federated Learning into two types based on the availability and number of client nodes: Cross-Silo FL and Cross-Device FL.

2.3.1 Cross-Silo FL

Cross-Silo Federated Learning is a type of FL that is commonly utilised in situations where there is a small number of client nodes, typically ranging from 2 to 100 nodes, and where these nodes are consistently available. This approach is often implemented within organisations or small groups of organisations with the aim of training models using confidential data. One of the main challenges with Cross-Silo FL is the communication and computation bottleneck it creates. Additionally, since participating organisations in the federation could be competitors, there is a need to develop an incentive scheme to encourage them to participate more actively.

2.3.2 Cross-Device FL

Cross-Device Federated Learning is commonly utilised in scenarios where there is a massive number of clients, usually in the form of mobile or IoT devices. However, it may encounter some drawbacks as clients may not always be available and can be unreliable, in addition to the possibility of facing a communication bottleneck due to clients being on slow WiFi or mobile network connections. This FL type was applied to enhance the Gboard, as previously discussed in Section 2.2.1.

2.4 Aggregation Algorithms

The aggregation algorithm is a crucial component of any Federated Learning system. It is responsible for receiving model updates from the participating clients and combining them to generate an updated global model. In the next sections, we will explore some examples of aggregation algorithms.

2.4.1 FedAvg

The Federated Averaging algorithm, which was initially introduced by Google [8], is based on the Stochastic Gradient Descent (SGD) optimisation algorithm. The algorithm is initiated by a centralised server, or coordinator node, which creates an initial global model and its corresponding parameters. Then, a subset of clients is selected by the server, to whom it sends the global model and parameters. The selected clients use their local data to train the model and send their respective local updates back to the coordinator. The coordinator then combines the local updates by generating a weighted sum, which is used to modify the global model. This process of sending the updated global model to the selected clients and receiving their local updates from them is repeated until the model converges or the maximum number of configured rounds is reached.

2.4.2 SMC-Avg

In Secure Multiparty Computation based aggregation [1], privacy is maintained by allowing distrustful clients to encrypt their model updates before sending them to the server. The server then aggregates the encrypted private values and sends the encrypted aggregated update back to the clients. The clients decrypt the encrypted update to obtain the updated global model. This process continues until the model converges or until a specified number of rounds is reached.

2.4.3 FedMA

FedMA [15] is specifically designed for Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. It aims to improve the global model by matching and averaging the hidden elements of multiple local models. For CNNs, it matches channels while for LSTMs it matches hidden states and neurons for fully connected layers. By matching and appropriately combining the hidden elements from different local models, the global model is enhanced. FedMA has been shown to outperform the popular FedAvg approach.

3 Security

As with many machine learning approaches, FL can be vulnerable to privacy and security issues. If the data or local models of the clients are compromised, the accuracy of the global model can be undermined. Furthermore, if the client's data can be inferred, it can lead to the leakage of confidential information. In this section, we will delve into the potential security challenges that FL may face.

3.1 Data Poisoning

Data poisoning attacks are one of the most common types of attacks in FL [13]. These attacks involve compromising the data of some clients participating in the federation by altering it in a way that influences the predictive power of the global model, thus reducing its accuracy. This can be achieved by injecting misleading or fraudulent data into the training data set or by modifying the labels of the data. Once the global model is trained on this poisoned data, it can lead to incorrect predictions, which can be detrimental in critical applications such as healthcare or finance. A method to avoid these attacks is to identify the malicious participants based on their local model performance before updating the global one.

3.2 Model Poisoning

Model poisoning attacks [3] are another type of security concern in Federated Learning. In this type of attack, an attacker compromises the local model parameters of some client devices to negatively impact the performance of the global model. Similar to data poisoning attacks, this can be detected by analysing the local performance of the client models. Usually, the error rate and loss function are used to identify compromised clients and prevent their participation in the federation.

3.3 Membership Inference Attacks

In FL, the training data is never shared outside of the local devices, but it is still possible to infer the data by analysing the model updates. To prevent this privacy issue, several mechanisms have been proposed, including:

1. **Secure Computation:** Secure Multiparty Computation and Homomorphic encryption are examples of mechanisms used to prevent inference attacks. These techniques ensure that local updates cannot be read and therefore, the training data cannot be inferred.

2. **Differential Privacy:** Differential privacy is a technique where noise is added to the model updates to keep the data private. The global model accuracy can be affected by the noise, but it helps to prevent inference attacks. [2]

3. **Trusted Execution Environments:** Trusted Execution Environments (TEEs) [9] have been used for privacy-preserving machine learning. TEEs provide private computing resources, which offer lower computing overhead and higher privacy than software solutions like Homomorphic encryption.

3.4 Backdoor Attacks

Backdoor attacks involve inserting a malicious task into the global model without affecting its current performance. Over time, this malicious task will begin to degrade the accuracy of the model. Compared to poisoning or inference attacks, detecting backdoor attacks can be challenging and resource-intensive, as they are designed to remain undetected for a longer period of time.

4 Dicussion

Federated Learning has gained significant attention across various fields, as demonstrated in this paper with examples from healthcare, natural language processing, and finance. It offers an efficient and privacy-preserving way to train a global model theoretically. However, there are some practical concerns associated with FL.

Although FL is based on the idea of good security and privacy, it is not guaranteed. As previously mentioned, vulnerabilities have been identified in FL that could compromise the global model or expose the training data.

Another concern is the scalability of FL. As the number of clients and the complexity of the models increase, the efficiency of communication and computation may be impacted. Additionally, the FL system must be fault-tolerant as the clients may not always be available due to connectivity problems or when in sleep mode.

Furthermore, in the case of Cross-Silo FL, a lack of an incentive program can result in a bottleneck in the system. Rival organisations in the federation may not find it worthwhile to participate as they may be assisting their competitors more than themselves.

Therefore, there is still a great deal of scope for future work to be done and applied in practice.

Finally, although there have been some theoretical solutions to these problems, for example, One-Shot FL was introduced [4] to reduce the amount of communication by training the global model after only one round of communication. And, in [14], we can find different incentive mechanisms using economic and game models to motivate rival organisations to participate in the federation.

5 Conclusion

This paper has discussed Federated Learning, covering various aspects such as network topology, data partition, data availability, and aggregation methods. Security and privacy concerns that could arise in a FL system have also been addressed, along with potential solutions. Finally, despite the promise of Federated Learning, there are still challenges that need to be tackled, and further research is required to improve its practical application.

References

[1] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for federated learning on user-held data. *arXiv preprint arXiv:1611.04482*, 2016.

[2] Cynthia Dwork. Differential privacy. In *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II 33*, pages 1–12. Springer, 2006.

[3] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Local model poisoning attacks to byzantine-robust federated learning. In *Proceedings of the 29th USENIX Conference on Security Symposium*, pages 1623–1640, 2020.

[4] Neel Guha, Ameet Talwalkar, and Virginia Smith. One-shot federated learning. *arXiv preprint arXiv:1902.11175*, 2019.

[5] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

[6] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

[7] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[8] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[9] Fan Mo and Hamed Haddadi. Efficient and private federated learning using tee. In *Proc. EuroSys Conf., Dresden, Germany*, 2019.

[10] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[11] Ronald L Rivest, Len Adleman, Michael L Dertouzos, et al. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180, 1978.

[12] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[13] Gan Sun, Yang Cong, Jiahua Dong, Qiang Wang, Lingjuan Lyu, and Ji Liu. Data poisoning attacks on federated machine learning. *IEEE Internet of Things Journal*, 9(13):11365–11375, 2021.

[14] Xuezhen Tu, Kun Zhu, Nguyen Cong Luong, Dusit Niyato, Yang Zhang, and Juan Li. Incentive mechanisms for federated learning: From economic and game theoretic perspective. *IEEE Transactions on Cognitive Communications and Networking*, 2022.

[15] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*, 2020.