5

Federated Learning Models in Decentralized Critical Infrastructure

Ilias Siniosoglou[1], Stamatia Bibi[1], Konstantinos-Filippos Kollias[1], George Fragulis[1], Panagiotis Radoglou-Grammatikis[1], Thomas Lagkas[2], Vasileios Argyriou[3], Vasileios Vitsas[2], and Panagiotis Sarigiannidis[1]

[1]University of Western Macedonia, Greece
[2]International Hellenic University, Greece
[3]Kingston University, England
E-mail: isiniosoglou@uowm.gr; sbibi@uowm.gr; dece00063@uowm.gr; gfragulis@uowm.gr; pradoglou@uowm.gr; tlagkas@cs.ihu.gr; vasileios.argyriou@kingston.ac.uk; vitsas@it.teithe.gr; psarigiannidis@uowm.gr

Abstract

Federated learning (FL) is a novel methodology aiming at training machine learning (ML) and deep learning (DL) models in a decentralized manner in order to solve three main problems seen in the artificial intelligence (AI) sector, namely, (a) model optimization, (b) data security and privacy, and (c) resource optimization. FL has been established as the "status quo" in today's AI applications especially in the industrial and critical infrastructure (CI) domain, as the three aforementioned pillars are invaluable in assuring their integrity. CIs include important facilities such as industrial infrastructures (smart grids, manufacturing, powerlines, etc.), medical facilities, agriculture, supply chains, and more. Deploying AI applications in these infrastructures is an arduous task that can compromise the CI's security and production procedures, requiring meticulous integration and testing. Even a slight mistake leading to the disruption of operations in these infrastructures can have dire consequences, economical, functional, and even loss of life. FL offers the needed functionalities to galvanize the integration and optimization of

artificial intelligence in critical infrastructures. In this chapter, we will outline the application of federated learning in decentralized critical infrastructures, its advantages and disadvantages, as well as the different state-of-the-art techniques used in the CI domain. We will showcase how the centralized ML approach transitions into the federated domain while we will show practical examples and practices of deploying the federated learning example in representative CIs, like, power production facilities, agricultural sensor networks, smart homes, and more.

Keywords: Federated learning, artificial intelligence, data security, critical infrastructures, model optimization, resource optimization.

5.1 Introduction

5.1.1 Definition and motivation

Federated learning (FL) is a distributed machine learning technique that allows multiple devices or entities to collaboratively train a model while keeping their data on-device. In federated learning, the data is distributed across a large corpus of devices or entities. This approach trains an AI model on the remote device using the local data and then sends only the model to a specified aggregation unit. There, a new and optimized global model is created by aggregating the model updates from all the devices. This approach allows for the training of models on large amounts of data without the need to transmit or centralize it, thus addressing the challenges of data privacy, security, and resource allocation.

The methodology was first introduced by the Google Research team in a 2016 paper titled "Communication-Efficient Learning of Deep Networks from Decentralized Data" [1]. It represents an advancement from traditional distributed machine learning and is designed to address the challenges of training AI models without the need to transfer data, for reasons related to computation, allocation, and privacy.

The motivation behind FL is to enable machine learning in scenarios where data is distributed across devices or is sensitive and cannot be centralized. For example, in the case of personalized healthcare, data may be collected from multiple devices such as wearables, smartphones, and hospitals. In these scenarios, it is not practical or secure to centralize the data and allows for the training of models without compromising the privacy and security of the data. Additionally, this approach can be applied in mobile computing, where data is distributed across millions of mobile devices [2],

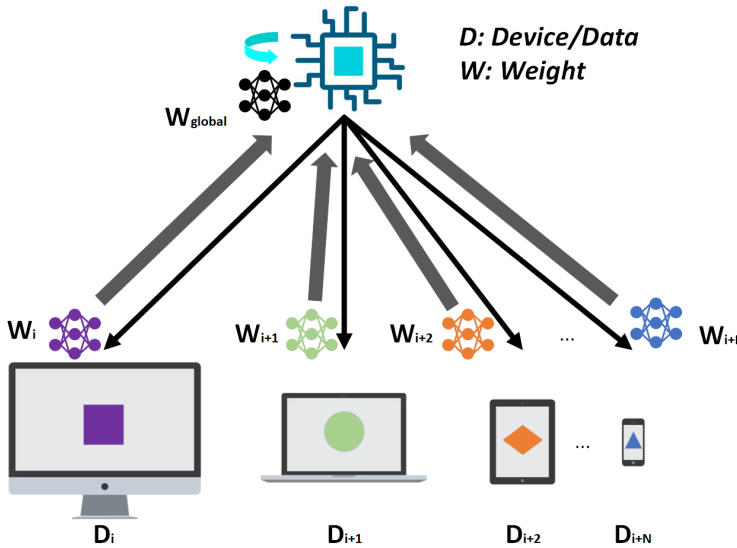

Figure 5.1 Federated learning concept.

and it allows training models on this data without the need to transmit large amounts of data over the network.

Federated learning also has the potential to democratize machine learning by enabling the participation of a large number of devices and entities in the training process. This can lead to more diverse and representative datasets, and also allows for training models in remote or underserved areas where data may not be easily accessible.

Federated learning can also be used to improve the performance of models in edge computing applications. By allowing devices to train models locally, federated learning can reduce the need for transmitting large amounts of data over the network, which can be beneficial in low-bandwidth or high-latency environments. Additionally, federated learning can enable the training of models that can be deployed on resource-constrained devices, such as IoT sensors or mobile phones.

5.1.2 Federated learning domains

Federated learning is an approach that aims to leverage the benefits of distributed AI model training. This approach is centered around three main pillars:

- **Model optimization:** Improves the model optimization process [3], [4] for the local node by providing an aggregated (global) model that contains knowledge accumulated by the aggregated models from all the devices.
- **Data privacy:** Preserves the integrity, security, and privacy of the data by keeping it at the edge nodes, rather than transferring it to a central infrastructure.
- **Resource optimization:** Designed to optimize [3], [5] the use of resources by communicating only the model parameters and some metadata between the federated server and the federated clients, instead of transferring the entire dataset. This conserves network resources and avoids possible bottlenecks, leads to lower latency, and allows for the distribution of the computing power needed for the AI model training among various nodes. Additionally, it enables to use the remote machines for the training process only when they are not used for other purposes, are connected to a steady power supply, and/or when there is a stable internet connection, which reduces the energy consumption of the federated process.

5.1.3 Use cases and applications

Federated learning has a wide range of use cases and applications, including but not limited to the following:

- **Personalized healthcare** can be used to train models that can predict a patient's health status or risk of developing a certain condition. This can be done by aggregating data from multiple devices such as wearables, smartphones, and hospitals. FL allows for the training of models without compromising the privacy and security of the patient's data, which is particularly important in the healthcare industry.
- **Mobile computing** can be used to train models on the large amounts of data generated by mobile devices such as smartphones and tablets. This can be used to improve the performance of mobile applications, such as natural language processing, image recognition, and more. For example, federated learning can be used to train models that can predict the battery life of a mobile device based on usage patterns.
- **Internet of Things** can be used to train models on data collected from IoT devices such as sensors and cameras. This can be used to improve the performance of edge computing applications, such as image and video processing, anomaly detection, and more.

- **Banking and finance** can be used to train models that can predict fraudulent transactions, by leveraging data from multiple banking institutions to train AI model, without actually transferring any data.
- **Natural language processing** can also be used to train language models by aggregating data from multiple sources without compromising the privacy of the data.

These are some examples of the utilization of the federated learning methodology in a variety of different popular domains. However, FL is continuously being adapted and tested to new applications as it is slowly becoming the baseline for machine learning in modern distributed infrastructures.

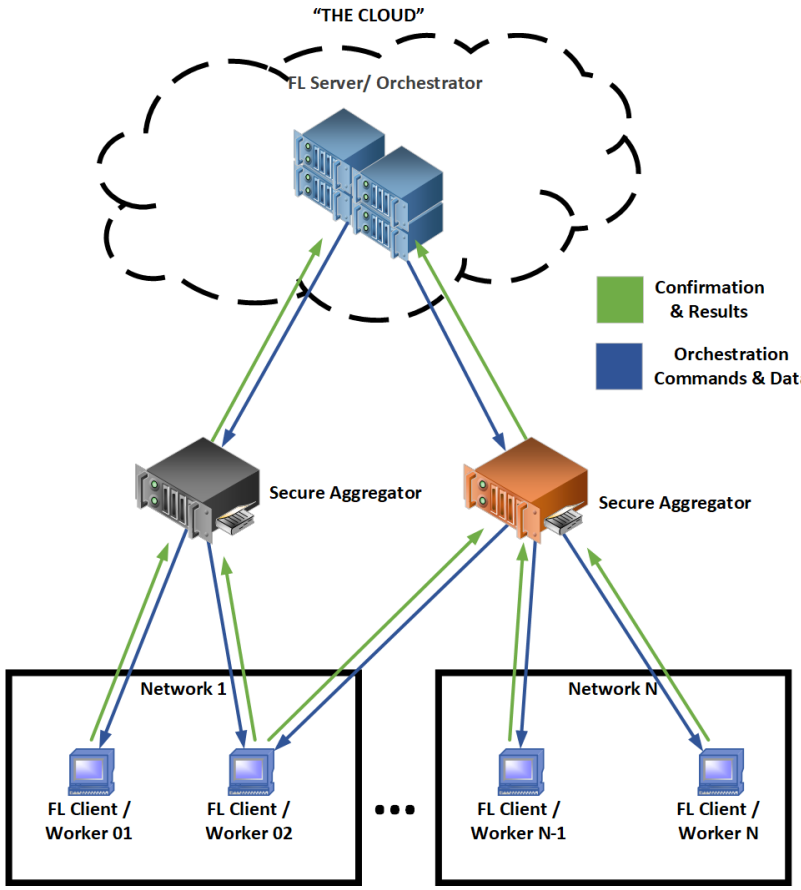

Figure 5.2 Simple federated learning architecture.

5.2 How Federated Learning Works

5.2.1 Overview of the architecture and process

Federated learning is a distributed machine learning methodology that allows for the training of deep learning models on a large corpus of edge devices. In this approach, models are trained locally on the edge devices, and their weights are sent to a central server where they are combined to form a global model using an algorithm such as federated averaging. The global model is then sent back to the remote devices for use. The central server distributes an initial global model to a population of federated devices, each of which holds a set of local data and a local model. These models are trained on the local data and the model weights are then retrieved by the central server to be combined using a predefined fusion algorithm, to create a new global model containing the new knowledge accumulated from the local models. This process is repeated for a number of iterations until the global model converges. Figure 5.3 shows a common process (strategy) followed to realize an FL training between a server and a corpus of devices. Figure 5.3 showcases a simple FL strategy for realizing a training session.

To get an idea about the modeling of the methodology process, we can depict a mathematical formula. Of course, since the process is directly connected to the fusion algorithm used, the FL process can be defined in a number of ways. For simplicity, we shall use the federated averaging algorithm to explain the process. Eqn (5.1) shows the process of fusing the local models from the remote devices in one global model [6].

$$w_G^k = \frac{1}{\sum_{i \in N} D_i} \sum_{i=1}^{N} D_i w_i^k. \tag{5.1}$$

Equation (5.1) Federated aggregation algorithm (FedAvg).

Here, the global model on the *k*th iteration is represented by w_G^k and the remote *i*th model at that iteration is represented by w_i^k. Each node holds a set of local data $D_{i \in \mathbb{N}}$ and local models w_i.

5.2.2 Key components

For the implementation of the described architecture, the system defines three main components [7] in order to realize the operation of the training, namely, a) the orchestrator, b) the aggregator, and c) the worker/client. Figure 5.2 shows how these components fit into the federated architecture.

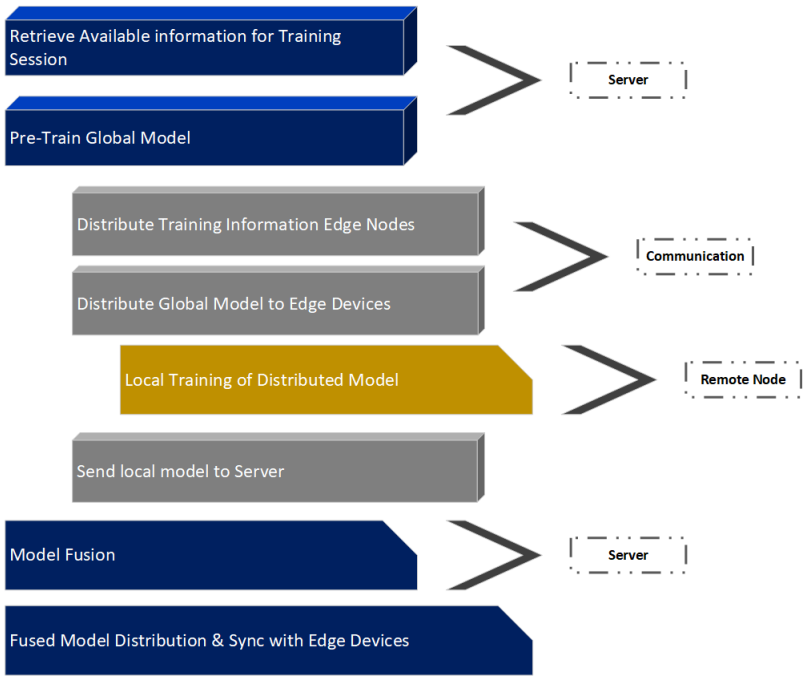

Figure 5.3 Simple federated learning pipeline.

5.2.2.1 Orchestrator

The orchestrator is responsible for managing the federated learning process, including initiating the FL session, selecting the population of devices, organizing the data, algorithm, and pipeline, setting the training context, managing communication and security, evaluating the performance, and, finally, synchronizing the FL procedure.

5.2.2.2 Aggregator

The aggregator is responsible for incorporating the updates from the local models into the global model. In some cases, the orchestrator also acts as the aggregator, particularly for smaller networks or certain security or operational requirements. The aggregator also implements security and privacy measures to protect the FL server and workers from any malicious actors.

5.2.2.3 Worker

The worker, also known as the party, is responsible for the local training that takes place during the FL training session. The worker is the owner of the

data and updates its model based on the newly received version of the global model after the local training and global model generation by the aggregator. The worker has the option of participating in the FL session or not, depending on resource allocation or criticality.

The abovementioned components established the foundation of the methodology. Depending on the type and nature of the deployment, these components can have additional responsibilities and placement or some extra components might be added. The different types of FL are described in the next section.

5.2.3 Types of federated learning

There is a variety of different federated learning application types that depend on a multitude of characteristics. A main characteristic that defines the type of the methodology applied is the way that data and their features are distributed and used by the different nodes. In particular, based on the data, we have the following:

- **Horizontal federated learning:** This type of approach trains models on data that is horizontally partitioned across different devices or entities. For example, training a model on data from different hospitals or different companies (Figure 5.4).
- **Vertical federated learning:** This type of federated learning trains models on data that is vertically partitioned across different devices or

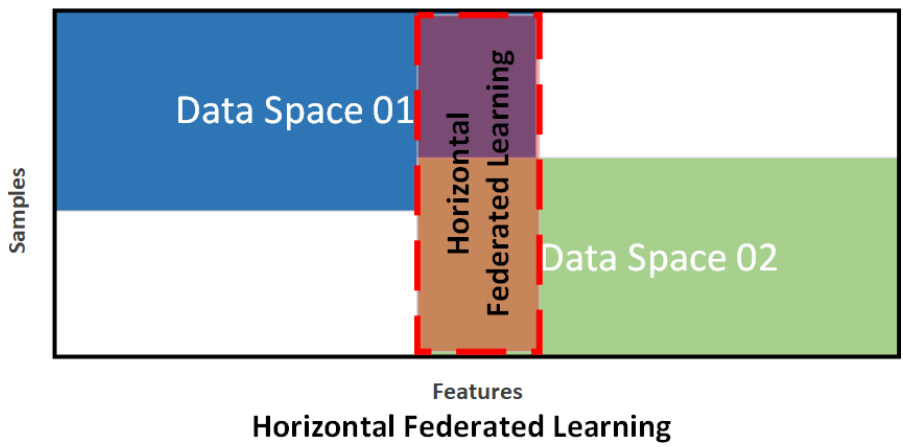

Figure 5.4 Horizontal federated learning.

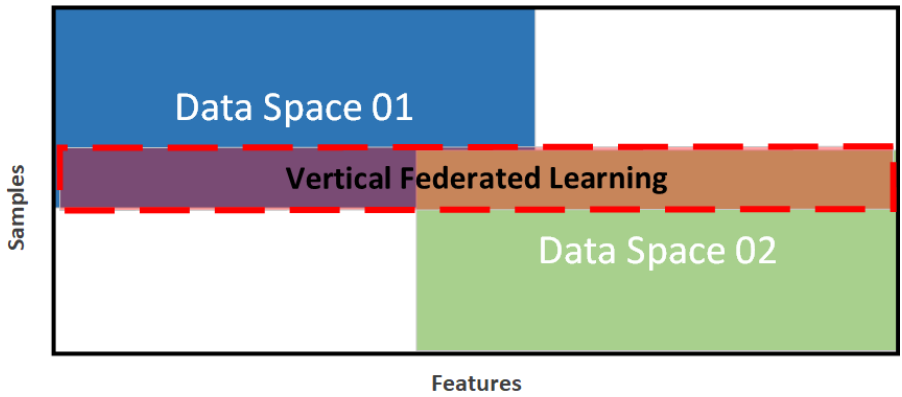

Figure 5.5 Vertical federated learning.

entities. For example, training a model on data from different features of the same patient (Figure 5.5).

- **Federated transfer learning:** This type of federated learning is focused on adapting a model pre-trained on one dataset to another related dataset (Figure 5.6).

However, the type of the federated learning approach used is not limited to the distribution of the data for the specific use case but depends on other characteristics such as the deployment constraints, the criticality of the data

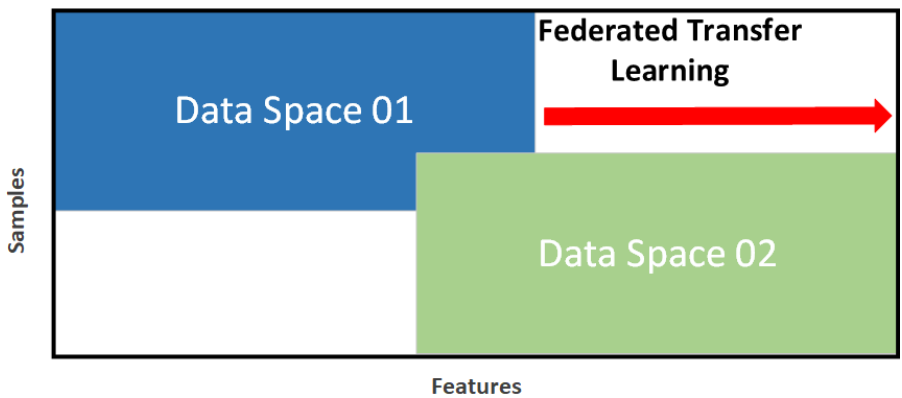

Figure 5.6 Federated transfer learning.

and infrastructure, and the nature of the task tackled. These preconditions orient the methodology and technique to adapt to the problem at hand and include the following approaches:

- **Multi-party federated learning:** This type of FL is similar to horizontal FL, but the data is under multiple parties' control. This type of federated learning is useful for the scenarios where data is not centralized but spread across multiple parties and each party wants to keep their data private.
- **Federated meta-learning:** This type of FL is focused on training a model that can adapt to new tasks or domains quickly by leveraging knowledge from previous tasks or domains.
- **Federated domain adaptation:** This type of FL is focused on adapting a model trained on one domain to work on another domain.
- **Federated few-shot learning**: This type of FL is focused on training a model that can learn to classify new classes with only a few examples.
- **Federated reinforcement learning:** This type of FL is focused on training a model using the reinforcement learning approach on the edge devices.

5.2.4 Model fusion algorithms

As mentioned before, the underlying core of the training procedure is the aggregation algorithm that undertakes the fusion of the distributed models into one optimized global model. Thus, the aggregation algorithm is a crucial component of FL as it determines the final performance of the global model. The most commonly used aggregation algorithm is federated averaging, which takes the average of the local models' weights to form the global model. However, there are other aggregation algorithms that can be used depending on the specific use case. For example, some algorithms weigh the contributions of the local models based on the quality of their data or the computational resources available on the device. These algorithms can help to mitigate the impact of data availability and device heterogeneity. Additionally, some algorithms use techniques such as differential privacy to protect the privacy of the data on the edge devices during the aggregation process. Overall, the choice of aggregation algorithm can have a significant impact on the performance and privacy of the final global model and should be carefully considered when implementing FL. Table 5.1 presents some of the common and state-of-the-art fusion algorithms that are widely used in different settings.

Table 5.1 Common fusion algorithms used in FL.

Algorithm	Year	Description	Benefits
FedAvg [1]	2017	An iterative model averaging FL framework	Reduces communication cost by locally computed updated aggregation
Zoo [8]	2018	Composable services to deploy ML models locally on edge	Reduces latency in data processing, and minimizes the raw data revealed
FedPer [9]	2019	Federated learning with personalization layers	Improves results with data heterogeneity, and communication cost
FedAsync [10]	2019	Asynchronous federated optimization framework	Improves flexibility and scalability and tolerates staleness
FedCS [11]	2019	Client selection for FL with heterogeneous resources	Improves performance and reduces training time
BlockFL [12]	2019	Blockchained federated architecture	Optimizes communication, computation, and latency
FedMa [13]	2020	Federated matched averaging algorithm for FL	Improves accuracy and communication cost
FedAT [14]	2020	Synchronous intra-tier training and asynchronous cross-tier training	Improves accuracy and reduces communication cost

5.3 Federated Learning vs. Traditional Centralized Learning

Federated learning is different from traditional centralized learning [15] in several ways. The most significant difference is that in traditional centralized learning, the data is collected and stored in a central location, where it is used to train the model. In contrast, federated learning keeps the data on the edge devices and trains the model locally on each device. This allows for the training of models on large amounts of data without the need to transfer it and also the ability to handle non-independent and identically distributed (IID) data. Additionally, federated learning preserves data privacy and security as the data never leaves the edge devices. This makes federated learning particularly well-suited for scenarios where data is sensitive or distributed across multiple devices. However, it is important to keep in mind that federated learning has its own set of challenges such as communication overhead, data availability, and model divergence.

Table 5.2 Comparison between federated and centralized learning.

Federated learning	**Traditional centralized learning**
Data remains on edge devices	Data is collected and stored in a central location
Model trained locally on each device	Model trained on centralized data
Suitable for non-IID data	Assumes data is IID
Preserves data privacy and security	Data privacy and security may be at risk
Requires communication between devices	No communication required between devices
Scales horizontally and vertically	Scales vertically
Suitable for sensitive or distributed data	Not suitable for sensitive or distributed data
Can handle many edge devices	Limited by the amount of data that can be centralized
Can have challenges such as communication overhead and model divergence	Fewer challenges than federated learning

5.3.1 Advantages and disadvantages of federated learning

By itself and as it is probably apparent, the federated learning approach is vast and, in its range, it encapsulates major advantages but also some drawbacks. As in all fields, the optimal deployment of federated learning is the fine line between the tradeoff of these advantages and drawback and strictly depends on the application of the methodology. For example, there might be some applications that require better model generalization but in expense of the communication efficiency of the network. Table 5.3 enumerates some of these advantages and disadvantages of federated learning in order to provide a better view of its utility.

5.3.2 Real-world examples of federated learning

5.3.2.1 Smart farming

In smart farming, federated learning can provide several benefits [16] by allowing for the training of models on data that is decentralized and spread across multiple devices or entities. The use case integrates IoT data from crops and animal care infrastructures, AR smart glasses, and other heterogeneous IoT devices, which can be difficult to source and gather in a central place to train a single AI model. By utilizing federated learning, it allows to train models on data that is distributed across great distances, making it possible to:

Table 5.3 Advantages and disadvantages of federated learning.

Advantages	Disadvantages
Collaborative learning: Allows multiple devices or entities to collaboratively train a model while keeping their data on-device. This allows for the training of models on large amounts of data without the need to transmit or centralize it.	**Data availability**: Data availability can be an issue in federated learning, as not all devices or entities may have access to the same data or may have data of different quality.
Data privacy and security: Allows for the training of models without compromising the privacy and security of the data. This is particularly important in scenarios where data is sensitive or distributed across multiple devices.	**Communication overhead:** Requires communication between the devices or entities, which can be a bottleneck, especially if the devices are located in different geographical locations.
Edge computing: Allows devices to train models locally, which can reduce the need for transmitting large amounts of data over the network. Additionally, it enables the training of models that can be deployed on resource-constrained devices, such as IoT sensors or mobile phones.	**Model divergence:** Can suffer from model divergence, where the local models may not converge to a common global model due to the non-IID data distribution on the devices.
Handling non-IID data: It is particularly well-suited for training models on non-IID data that is commonly found in the real-world scenarios.	**Latency:** Can suffer from latency issues, as it requires communication between the devices or entities to exchange model updates.
Scalability: It is highly scalable and can handle a large number of devices or entities.	**Complexity:** Can be complex to implement and requires a lot of communication and coordination between the devices or entities.

- Formulate best practices for farming and livestock production in expanding the specific market by discovering weaknesses in the agricultural systems and providing insightful predictions to help end-users make informed decisions about their infrastructure's operations.
- Formulate rules and quantified metrics for optimum conditions in terms of (animal) behavior, psychiatry, food quality, nutrition, and agriculture environment by training models on the diverse data sources from different scenarios.
- Increase farm and livestock production by using AI-supported strategies that improve agricultural systems' sustainability, productivity, and risk.

- Provide feedback on how to ensure proper decision support by using the knowledge accumulated from the local models to improve the global model.

5.3.2.2 Smart, sustainable, and efficient buildings

In the use case of smart, sustainable, and efficient buildings, FL can provide several benefits [17]. By using IoT data in smart buildings to optimize the energy footprint and automate building management using AI-based solutions, FL can be used to train models on large amounts of data from multiple devices or entities, while keeping the data on-device. This allows for the training of models on large amounts of data without the need to transmit or centralize it, which can help to preserve the privacy and security of the data.

5.3.2.3 Industrial supply chains

In the context of the industrial supply chain use case, FL can provide significant benefits by improving the forecasting accuracy [18] for fulfilling the demand from retailers and agencies, who are attempting to satisfy the demand from their consumers. This is achieved by utilizing the abundance of product codes, complexity of certain manufacturing processes, and short lifetime of most products in the supply chain, which make production scheduling and market-oriented forecasting challenging. In this frame, FL allows for the collaborative training of models across different supply chains of the end-user, without the need to transfer or centralize the data. This can improve the forecasting accuracy by leveraging the knowledge and data from different product codes produced by the end-user. Additionally, the use of FL can protect the data privacy and resources of the end-user's infrastructure, by keeping the data on-device, and avoiding the need for centralizing and transferring it. Furthermore, by applying this technique to optimize the forecasting accuracy and using the heterogeneous data from different product codes, it can lead to the end-user's better decision making and better supplier−customer relationship.

5.3.2.4 Industrial infrastructures

In the use case of mixed reality and ML-supported maintenance and fault prediction of IoT-based critical infrastructure, the benefit of FL is its ability to predict the behavior of industrial devices, such as controllers, in order to identify potential defects and malfunctions. This enables the end-user to monitor

and prevent problems in the operation of each industrial infrastructure. The technique is applied to a large number of industrial devices that are divided and installed in decentralized optical switches. The use case makes use of the ability of federated learning to handle many edge devices, both horizontally by scaling to more devices such as small form-factor pluggable (SFP) modules or switches and vertically by applying a hierarchical model optimization. This allows for more efficient and accurate predictions and maintenance operations for the critical infrastructure.

5.3.2.5 Medical sector

Federated learning can bring several benefits to the medical sector [19], [20], particularly in a use case of a collection of hospitals across a large distance. Some of the benefits include:

- **Data privacy and security:** Allows for the training of models without compromising the privacy and security of the patients' data. This is particularly important in the medical sector where data is sensitive and regulated.
- **Handling non-IID data:** It is particularly well-suited for training models on non-IID data, which is commonly found in the medical sector. By training models on the local data from different hospitals, the models can learn from diverse patient populations, resulting in more robust models.
- **Edge computing:** Allows hospitals to train models locally, which can reduce the need for transmitting large amounts of data over the network. Additionally, it enables the training of models that can be deployed on resource-constrained devices, such as mobile devices used by clinicians and nurses.
- **Collaborative learning:** Allows multiple hospitals to collaboratively train a model while keeping their data on-device. This allows for the training of models on large amounts of data without the need to transmit or centralize it.
- **Scalability:** It is highly scalable and can handle a large number of hospitals across a large distance. This makes it suitable for large-scale healthcare studies and research.
- By using, hospitals can train models on their local data without sharing any sensitive information across the network, while still being able to build models that generalize well to different patient populations. This can lead to better diagnosis, treatment, and ultimately patient outcomes.

5.4 Implementing Federated Learning

Implementing federated learning requires a few key components. First, a centralized server is needed to aggregate the models trained on the local devices and distribute the updated global model back to the devices. Second, there should be a mechanism for the local devices to communicate with the central server and securely exchange model updates. Third, a mechanism for data partitioning is needed to ensure that the devices are training models on non-overlapping data. Fourth, a method for combining the local models into a global model, such as federated averaging, is necessary. Lastly, it is important to have a way to evaluate the performance of the model and monitor the FL process. Additionally, it is important to have a good understanding of the underlying deep learning model and the data that are being used. It is also important to consider the security and privacy aspects of the FL process, as well as the network infrastructure to ensure that the devices can communicate effectively with the central server.

5.4.1 Tools and frameworks available

Since its introduction, federated learning has continuously been explored and integrated into a variety of commercial and industrial applications. To support the migration from conventional deep learning, a lot of diverse frameworks have been proposed and used to both deploy or experiment with the FL methodology. Table 5.4 enumerates some of the most used frameworks that exist today.

5.4.2 Challenges

Despite the many potential benefits of federated learning, there are still some challenges that need to be addressed before it can be widely adopted. These

Table 5.4 Available federated learning frameworks and tools.

Framework	**Type**
Tensorflow federated **[21]**	Research
FATE **[22]**	Production
Flower **[23]**	Production/research
PySyft **[24]**	Production/research
IBM federated **[25]**	Production/research
Leaf **[26]**	Research
OpenFL **[27]**	Production/research

include issues related to data privacy and security, as well as the need for robust methods for aggregating updates from multiple devices. Additionally, it requires the design of efficient algorithms to handle the high-dimensional and non-IID data across the devices, and more. Therefore, it is an active area of research and development, with many ongoing efforts aimed at addressing these challenges and making the approach more practical and widely applicable. Common challenges in the federated learning domain include problems that derive by its innate nature, such as:

- **Untrusted sources:** O ne of the challenges is the presence of untrusted sources, which can be devices or entities that may not have the same level of security or data privacy as the other participants. This can lead to potential breaches of security or privacy and can compromise the integrity of the model.
- **Adversarial attacks:** FL is also vulnerable to adversarial attacks, where an attacker may attempt to manipulate the local models or the global model, leading to a decrease in the accuracy of the model.
- **IID and non-IID data processing:** FL requires the data distributed across the devices or entities to be identically independently distributed (IID), which is not always the case. In scenarios where data is non-IID, the local models may not converge to a common global model, leading to a decrease in the accuracy of the model.
- **Synchronization problems:** FL requires coordination and communication between the devices or entities, and synchronization problems can occur if the devices or entities are not able to communicate or coordinate effectively.
- **Small number of participants:** FL requires a large number of devices or entities to participate in order to effectively train a model. If the number of participants is small, the model may not be able to effectively learn from the data.
- **System infiltration:** In FL, since the data is distributed across multiple devices or entities, it can be vulnerable to infiltration by malicious actors who can attempt to access the data or manipulate the models.

5.5 Conclusion

Federated learning is a novel methodology created on the basis of distributed training of AI models, heavily oriented at keeping the distributed data private while also optimizing the models and the resources used. It is particularly

useful in the industrial and critical infrastructure domain, as it allows for the integration and optimization of AI in these systems without compromising their integrity. FL offers several advantages in terms of deployment, scalability, and security; however, it also poses some challenges in terms of implementation, communication, and model optimization, especially when considering the distribution of the distributed resources. It is a status quo in today's AI applications. The chapter focuses on introducing the basics of the federated learning methodology, the application of FL in decentralized critical infrastructures, outlining the advantages and disadvantages and different techniques used in the field. It provides practical examples of FL's deployment in various infrastructures such as power production facilities, agricultural sensor networks, and smart homes and more while also summarizing the currently available sources.

Acknowledgements

This work has received funding from the European Union's Horizon 2020 research and innovation program under Grant Agreement No. 957406 (TERMINET).

References

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, February 2016. W364W7352

[2] "Federated Learning: Collaborative Machine Learning without Centralized Training Data – Google AI Blog," [Online]. Available: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html.W364W7352

[3] J. K. Konečnťy, H. B. Mcmahan and D. Ramage, "Federated Optimization:Distributed Optimization Beyond the Datacenter," November 2015. W364W7352

[4] I. Siniosoglou, V. Argyriou, S. Bibi, T. Lagkas and P. Sarigiannidis, "Unsupervised Ethical Equity Evaluation of Adversarial Federated Networks," ACM International Conference Proceeding Series, August 2021. W364W7352

[5] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh and D. Bacon, "Federated Learning: Strategies for Improving Communication Efficiency," October 2016. W364W7352

[6] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y. C. Liang, Q. Yang, D. Niyato and C. Miao, "Federated Learning in Mobile Edge Networks: A Comprehensive Survey," IEEE Communications Surveys and Tutorials, vol. 22, no. 3, pp. 2031-2063, July 2020. W364W7352

[7] I. Siniosoglou, P. Sarigiannidis, V. Argyriou, T. Lagkas, S. K. Goudos and M. Poveda, "Federated Intrusion Detection in NG-IoT Healthcare Systems: An Adversarial Approach," IEEE International Conference on Communications, June 2021. W364W7352

[8] J. Zhao, R. Mortier, J. Crowcroft and L. Wang, "Privacy-Preserving Machine Learning Based Data Analytics on Edge Devices," AIES 2018 - Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, pp. 341-346, December 2018. W364W7352

[9] M. G. Arivazhagan, V. Aggarwal, A. K. Singh and S. Choudhary, "Federated Learning with Personalization Layers," December 2019. W364W7352

[10] C. Xie, O. Koyejo and I. Gupta, "Asynchronous Federated Optimization," March 2019. W364W7352

[11] T. Nishio and R. Yonetani, "Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge," IEEE International Conference on Communications, Vols. 2019-May, April 2018. W364W7352

[12] H. Kim, J. Park, M. Bennis and S. L. Kim, "Blockchained on-device federated learning," IEEE Communications Letters, vol. 24, no. 6, pp. 1279-1283, June 2020. W364W7352

[13] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos and Y. Khazaeni, "Federated Learning with Matched Averaging," February 2020. W364W7352

[14] Z. Chai, Y. Chen, A. Anwar, L. Zhao, Y. Cheng and H. Rangwala, "FedAT: A High-Performance and Communication-Efficient Federated Learning System with Asynchronous Tiers," International Conference for High Performance Computing, Networking, Storage and Analysis, SC, October 2020. W364W7352

[15] M. Asad, A. Moustafa and T. Ito, "Federated Learning Versus Classical Machine Learning: A Convergence Comparison," July 2021. W364W7352

[16] T. Manoj, K. Makkithaya and V. G. Narendra, "A Federated Learning-Based Crop Yield Prediction for Agricultural Production Risk Management," 2022 IEEE Delhi Section Conference, DELCON 2022, 2022. W364W7352

[17] U. M. Aïvodji, S. Gambs and A. Martin, "IOTFLA : AA secured and privacy-preserving smart home architecture implementing federated learning," Proceedings - 2019 IEEE Symposium on Security and Privacy Workshops, SPW 2019, pp. 175-180, May 2019. W364W7352

[18] T. Li, A. K. Sahu, A. Talwalkar and V. Smith, "Federated Learning: Challenges, Methods, and Future Directions," IEEE Signal Processing Magazine, vol. 37, no. 3, pp. 50-60, August 2019. W364W7352

[19] M. Joshi, A. Pal and M. Sankarasubbu, "Federated Learning for Healthcare Domain - Pipeline, Applications and Challenges," ACM Transactions on Computing for Healthcare, vol. 3, no. 4, pp. 1-36, November 2022. W364W7352

[20] N. Rieke, J. Hancox, W. Li, F. Milletarì, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein, S. Ourselin, M. Sheller, R. M. Summers, A. Trask, D. Xu, M. Baust and M. J. Cardoso, "The future of digital health with federated learning," npj Digital Medicine 2020 3:1, vol. 3, no. 1, pp. 1-7, September 2020. W364W7352

[21] Federated Learning | TensorFlow Federated. W364W7352

[22] "Fate," [Online]. Available: https://fate.fedai.org/.W364W7352

[23] "Flower: A Friendly Federated Learning Framework," [Online]. Available: https://flower.dev/.W364W7352

[24] "PySyft - OpenMined Blog," [Online]. Available: https://blog.openmined.org/tag/pysyft/.W364W7352

[25] "IBM Federated Learning - IBM Documentation," [Online]. Available: https://www.ibm.com/docs/en/cloud-paks/cp-data/4.0?topic=models-federated-learning-tech-preview.W364W7352

[26] "LEAF - A Benchmark for Federated Settings," [Online]. Available: https://leaf.cmu.edu/.W364W7352

[27] "OpenFL - Creative expression for desktop, mobile, web and console platforms," [Online]. Available: https://www.openfl.org/.W364W7352

[28] L. Li, Y. Fan, M. Tse and K. Y. Lin, "A review of applications in federated learning," Computers & Industrial Engineering, vol. 149, p. 106854, November 2020. W364W7352

[29] I. Siniosoglou, P. Sarigiannidis, Y. Spyridis, A. Khadka, G. Efstathopoulos and T. Lagkas, "Synthetic Traffic Signs Dataset for Traffic Sign Detection & Recognition in Distributed Smart Systems," Proceedings - 17th Annual International Conference on Distributed Computing in Sensor Systems, DCOS 2021, pp. 302-308, 2021. W364W7352