Federated Learning Methods for Analytics of Big and Sensitive Distributed Data and Survey

Michal Staňo, Ladislav Hluchý, Martin Bobák, Peter Krammer and Viet Tran

March 5, 2023

Federated Learning Methods for Analytics of Big and Sensitive Distributed Data and Survey

1st Michal Staňo
Institute of Informatics Slovak Academy of Sciences
Bratislava, Slovakia
michal.stano@savba.sk
&
Faculty of Mathematics, Physics and Informatics
Comenius University
Bratislava, Slovakia
michal.stano38@fmphi.uniba.sk

2nd Ladislav Hluchý
Institute of Informatics Slovak Academy of Sciences
Bratislava, Slovakia
ladislav.hluchy@savba.sk

3rd Martin Bobák
Institute of Informatics Slovak Academy of Sciences
Bratislava, Slovakia
martin.bobak@savba.sk

4th Peter Krammer
Institute of Informatics Slovak Academy of Sciences
Bratislava, Slovakia
peter.krammer@savba.sk

5th Viet Tran
Institute of Informatics Slovak Academy of Sciences
Bratislava, Slovakia
viet.tran@savba.sk

***Abstract*—This article focuses on analytics of big distributed sensitive data on a federated learning base. The main current focus is on the most common use technology platforms: TensorFlow Federated, PySyft, Flower and IBM Federated Learning of the point of view edge computing usability. Training PyTorch models with differential privacy (DP) is more scalable than existing state-of-the-art methods. Differential privacy is a mathematically rigorous framework for quantifying the anonymisation of sensitive data. It's often used in analytics, with growing interest in the machine learning (ML) community. Training distributed data at the edge is interesting for privacy sensitivity and the transfer of huge data. Sensitivity and huge data is the main challenge in federated learning. Federated learning is a solution for protecting huge device data through model updates.**

Keywords — Federated Learning, Artificial Intelligence, TensorFlow Federated, Flower, Differential Privacy

I. Introduction

Nowadays, the fast-growing Artificial Intelligence sector demands big data and privacy. Our focus is on images of the airports. Subjects such as airports commonly have strict roles in sharing information with anybody. These rigid roles are to protect against privacy breaches and big data protection. It is, therefore, essential to find a way to use such privacy-sensitive data without needing to collect it in a centralised system. A solution to this challenge is *Federated Learning* [1], which grip advantage of large-scale clients that jointly train a central model. This paper aimed to survey to classify airport images according to federated learning in a distributed manner. The usability and robustness of the technology were essential to target. A distributed method runs on multiple clients (computers, mobile devices, institutes) and is networked to a central server. It is interesting to talk about open-source frameworks often used in *Federated Learning*.

Acknowledgements: This work was supported by the project VEGA 2/0131/23 (2023-2025) and by the Slovak Research and Development Agency under contract no. APVV APVV-20-0571 (Icontrol project).

In this paper, we focus on the results of models in IID and Non-IID modes. Specialised hardware for speeding up purpose computation are GPU, TPU, and FPGA. The evolution is dynamic and involved in the context of the responsible development of human-centric and trustworthy AI systems with the most notable document, "European Union Guideline on Ethics in Artificial Intelligence: Context andImplementation" as well as other worldwide data regulation and protection laws [2-5]. After four years of GDPR since 2018, there are two high contrasts: 1) the rise of distributed data analytics using AI/ML with the need for cross-organization data sharing and 2) the rise of data privacy protection and data security. First, several distributed ML architectures emerged to address the problem of enabling collaboration between different organisations without sharing their raw data. The most prominent is federated learning (FL). Regarding data privacy protection, numerous techniques are increasingly used in actual use cases. Among them, in addition to the classic methods focused on anonymising sensitive data (e.g. k-anonymity, l-diversity or t-closeness, among many others), differential privacy (DP) has experienced a significant boom in recent years and is present in numerous applications as mentioned in [6]. These transitions are more important and visible in landscaping than ever, and this will continue to be an innovation area over the coming years.

II. Federated Learning

Federated Learning was tailored in 2017 by H. B. McMahan [1]. It is a new idea of Machine Learning. This name was created for the learning task, which solved a loose federation of clients. Clients of this idea are devices (mobile devices or institutes, airports, hospitals) which participated. These clients are controlled by one primer server.

A long-term goal of many research & development teams (covering databases, cryptography, and machine learning) is to learn and analyse from data distributed between many clients without publishing that data. Computational methods for data encryption date back to the 1980s [7], [8]. We will demonstrate the first known and public names of authors who focus on local data training by central orchestrator and protect sensitive data. Agrawal and Skrikant [9], Vaidya et al. [10]. Huge client data is stored by clients' sites (mobile devices) and not transferred. Use updates from the client's devices to achieve the target training model. In terms of FL functions, the following features are worth mentioning. *Decentralised data:* Federated Learning enables machine learning models to be trained on data distributed across multiple devices or servers without centralising the data [1]. *Privacy-preserving:* With Federated Learning, data remains on the user's device, which helps to ensure that sensitive data is not exposed to others. This makes Federated Learning an attractive approach for applications that require privacy and security [11]. *Collaborative learning:* Federated Learning enables multiple devices or servers to collaborate and contribute to the training of a machine learning model, even if they have different data [12]. *Efficient:* Federated Learning can reduce the amount of data that needs to be transmitted to a central server, reducing the computational and communication overheads [13]. *Iterative updates:* Federated Learning allows cumulative updates to be made to the machine learning model over time without requiring full retraining [1]. *Flexibility:* Federated Learning is a flexible approach used in various settings, including edge devices, mobile devices, and data centres [14-16].

Federated learning deployment involves deploying a federated learning system in a production environment, such as on mobile devices, edge devices, or cloud servers. This process typically involves several steps, including designing the system architecture, selecting appropriate algorithms and models, implementing the system, testing and validating the system, and finally, deploying the system to users. One of the critical challenges in federated learning deployment is ensuring the privacy and security of the user data, as well as the models being trained. This often requires advanced cryptographic techniques such as secure multi-party computation or homomorphic encryption to protect the privacy of the user data and models. Another essential consideration in federated learning deployment is optimising device communication, which can be a significant bottleneck in the training process. This can be achieved through techniques such as compressing model updates, reducing the number of communication rounds, or using differential privacy to reduce the amount of data that needs to be transmitted. Overall, successful deployment of a federated learning system requires a deep understanding of the system architecture, algorithms, and privacy and security considerations, as well as careful testing and validation to ensure that the system performs as expected in production environments [17].

We would like to introduce you to some of the most popular frameworks in federated learning:

TensorFlow Federated (TFF): TFF is an open-source library for TensorFlow that provides an API for federated learning. TFF allows developers to create and test federated models quickly and provides various optimisations for efficient model training.

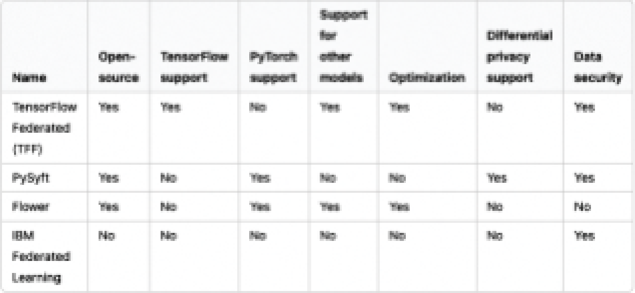
PySyft: PySyft is an open-source Python library that allows developers to create secure and decentralised machine learning applications. PySyft supports federated learning, differential privacy, and other techniques for processing sensitive data.

Flower: Flower (FL) is another open-source framework for federated learning. Flower provides an abstraction for federated learning and allows developers to create and manage federated models easily. Flower supports several different algorithms for optimising model training.

IBM Federated Learning: IBM Federated Learning is a federated learning platform provided by IBM. The platform allows developers to create and train models on decentralised data and provides tools for managing data security and privacy.

These frameworks differ in some aspects, such as support for different models and optimisations, so the choice depends on the project's specific requirements. However, they are certainly a good place to start working with federated learning. [18]

TABLE 1
Comparison of the state-of-the-art Federated Learning frameworks



Name	Open-source	TensorFlow support	PyTorch support	Support for other models	Optimization	Differential privacy support	Data security
TensorFlow Federated (TFF)	Yes	Yes	No	Yes	Yes	No	Yes
PySyft	Yes	No	Yes	No	No	Yes	Yes
Flower	Yes	No	Yes	Yes	Yes	No	No
IBM Federated Learning	No	No	No	No	No	No	Yes

The next option is support for federated learning in the MATLAB framework [23, 24]. MATLAB provides several functions and tools for implementing federated learning, such as the distributed function for distributed computing and the Machine Learning Toolbox for creating and training models. MATLAB also supports creating and managing connections between clients and a server in federated learning. We can use the Federated Learning Toolbox in Matlab to develop, simulate, and deploy federated learning algorithms. The Federated Learning Toolbox provides various tools and functionalities to create and evaluate federated learning workflows. With the Federated Learning Toolbox, we can:

- Create and manage a federated learning simulation environment.
- Define the communication and aggregation protocols for the federated learning workflow.
- Train machine learning models using the federated learning approach on distributed datasets.
- Evaluate the performance of the federated learning model and compare it with other models.
- Generate code for the federated learning algorithm and deploy it on different devices.

We can use the Federated Learning Toolbox in Matlab to implement various use cases, such as training machine learning models on distributed sensor data, collaborative machine learning for healthcare, and privacy-preserving machine learning for financial data. The toolbox provides various examples and tutorials to get started with federated learning in Matlab. [19]

In the case of Federated Learning, it's essential to talk about Orchestrators. Talking about orchestrators in Federated Learning is essential for several reasons:

Management and coordination of distributed learning: The orchestrator is a critical component of Federated Learning that ensures learning coordination among multiple clients and the server. The orchestrator provides even task allocation among clients and maintains a consistent state of the model.

Resource management: In distributed learning, it is essential to have resource management, such as computational resources and memory. The orchestrator can help automate resource management, simplifying the configuration and management of distributed learning.

Security: Federated Learning often involves sensitive data and models. The orchestrator can help ensure that only authorised clients can access this data and models.

Scalability: Federated Learning can train models on a large amount of data from many clients. The orchestrator can help with automated resource management and ensure that the system can scale as the number of clients and amount of data increases.

Implementation: Some many different orchestrators and frameworks can be used in Federated Learning. I want to discuss these other options, which can help choose the most appropriate one for a particular use case.

Overall, it is essential to talk about orchestrators in Federated Learning because these tools are critical for coordinating and managing distributed learning and ensuring security, scalability, and successful implementation of the Federated Learning system. [20]

III. Architectural solutions of Federated Learning Platforms

This scheme from Fig. 1. organises the schema linearly, with client libraries and server infrastructure on the left and components for privacy protection, security, and model management on the right. Communication infrastructure and federated learning algorithms are located in the middle, as they bridge client and server components. The privacy and data security component is further expanded to demonstrate specific techniques used to ensure the privacy and security of user data.

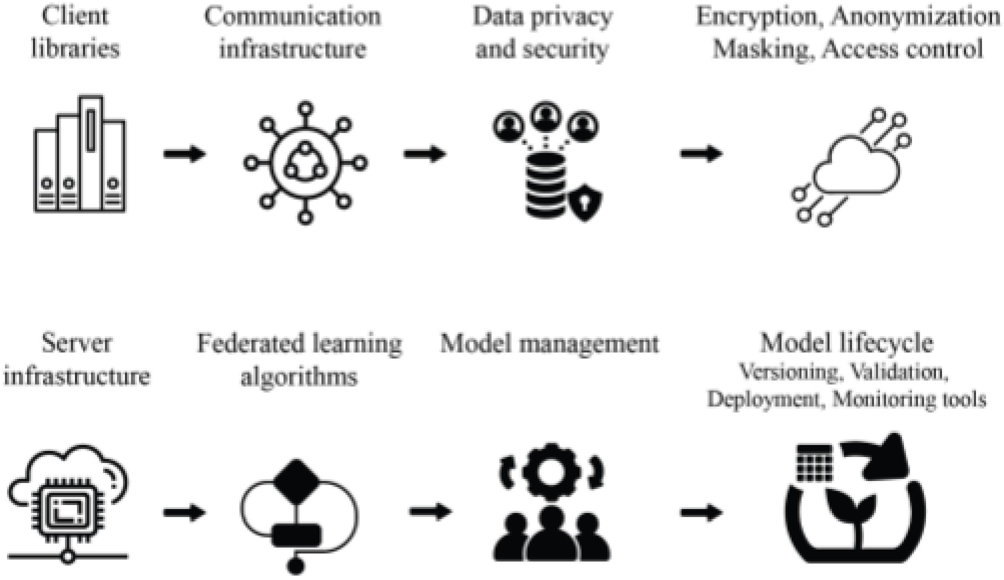

Client libraries
Communication infrastructure
Data privacy and security
Encryption, Anonymization Masking, Access control
Server infrastructure
Federated learning algorithms
Model management
Model lifecycle
Versioning, Validation, Deployment, Monitoring tools

Fig. 1. The architectural scheme of Federatel Learning Platforms

IV. Execution platform for federated learning

The proposed platform has to be built upon distributed computing which can handle large-scale data [26] and offer efficient communication among clients. The core challenge is to deal with data heterogeneity and data distribution with various policies. The data services also need to support various data formats for structured, semi-structured as well as unstructured data which are cleaned, transformed, and labeled for use in federated machine learning models. Machine learning computations are performed on the data federation by the above mentioned frameworks.

The next requirement is high scalability. The proposed approach exploits cloud technologies through which the platform can support large-scale federated machine learning. Cloud technologies allow the allocation of resources dynamically according to the current utilization of the service. It is suitable for federated machine learning where data size and model complexity varies according to the problem type and instance.

In the case, there is a huge number of clients, the computations need to be orchestrated which supports automated deployment, scaling, resiliency, and fault tolerance across clusters of clients. As the number of users increases, it becomes difficult to manually manage the application. Also, the clients are typically heterogeneous, they have different hardware configurations, network conditions, and data distribution. Thus, there is a strong demand to dynamically adjust computation taking into account the capabilities of clients.

Kubernetes is a popular open-source container orchestration system used for deploying, scaling, and managing containerised applications. In Federated Learning, Kubernetes can be used as an orchestration system to manage the resources required for training a machine learning model in a distributed manner. Kubernetes provides a way to manage the resources required for training a machine learning model, such as CPU, GPU, memory, and storage. With Kubernetes, you can easily provision and scale resources, manage the deployment of containers, and handle load balancing. In Federated Learning, Kubernetes can be used to manage the deployment of the Federated Learning server and the Federated Learning clients. The server manages the training process by coordinating the clients' training, aggregating the clients' updates, and updating the model. The clients perform the local training on their own data and send the updates to the server. Kubernetes can also help manage the communication between the server and the clients, such as network configurations, routing, and security. Kubernetes can provide features such as service discovery and load balancing to ensure that the communication between the server and clients is reliable and efficient. Kubernetes can be a powerful tool for managing the resources required for training a machine learning model in a Federated Learning setting, providing scalability, fault tolerance, and efficient resource utilisation. [20].

However, federated learning often requires distributed execution of clients located in different organisations because data cannot be moved out of the organisations. Therefore we are proposing a new execution platform based on the service mesh concept [21] for federated learning. For the implementation of the platform, we are using Consul [22] for the network communication layer and Nomad for workload management. The behaviour and functionalities of infrastructure based on Consul/Nomad is very similar to Kubernetes, and they can be distributed over multiple data centres/organisations. A prototype of the infrastructure has been deployed on two OpenStack sites located in Bratislava, Slovakia, and Santander, Spain.

V. Challenge Task for Federated Learning

Federated learning methods and techniques have already been successfully applied in many domains and tasks. However, some specific tasks still present a challenge due to specific characteristics. One such domain is in air transport domain. More specific, it is the task of determining visibility at airports, while this task must be solved at every airport in Europe - for legislative and security reasons. Visibility or observation-ability at the airport significantly affects the safety of aircraft landing and take-off. Aspects such as rain, fog, or heavy snow can significantly affect safety. This task is defined in more detail in the article [25]. However, airports are among the critical infrastructure in which data are sensitive and must therefore be protected. However, many airports are dealing with the same problem and need a model to estimate visibility level. However, not all airports have at their disposal sufficiently representative historical data about weather and surrounding influences. The use of federated learning therefore appears to be a logical choice in this task due to the non-sharing of data and the advantage of one jointly trained model on a large number of data covering many different and even extreme situations. However, the problem is caused by the individual characteristics of each airport. Such individual characteristics include: geographic location, altitude, but especially the location of objects around the airport, which are used as a calibration for estimating visibility. Individual calibration objects have different distances, directions, height, colour or optical properties. The example of the visibility of individual calibration objects are shown in Fig. 2. Also, different conditions in terms of visibility prevail during the day and at night.

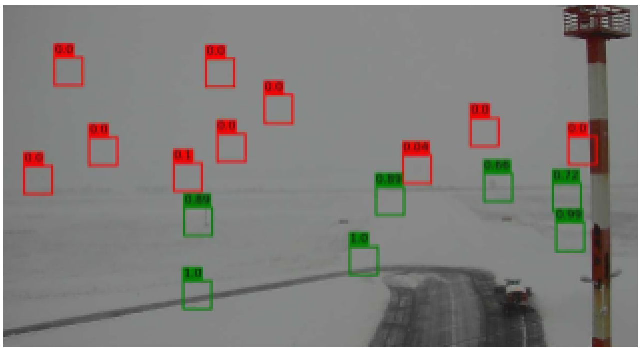


Fig. 2: An example of an image from the airport during the winter season, which shows the visibility of individual calibration objects. The final visibility level for specific direction is determined based on calibration objects visibility.

There are still open and unsolved tasks in the field of federated learning, for which it is necessary to design new methods and approaches. The task of determining visibility at airports is one of the hot candidates to be solved, thus increasing safety in air transport.

VI. Conclusion

The choice of a federated learning framework for classifying airport images will depend on several factors, such as the specific requirements of the project and the available resources. However, TensorFlow Federated (TFF) is a good framework to consider for this type of project. TFF has a wide range of features and tools that can be used for image classification tasks, including pre-trained models and optimisers for efficient training. TFF also has good support for distributed computing, which can be helpful when training models on large datasets. Another framework to consider is PySyft, which also strongly supports image classification and offers additional features such as differential privacy to help ensure data security and privacy. Ultimately, the best choice of framework for the project will depend on a range of factors, and it may be worthwhile to evaluate several options before making a final decision.

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,” *in Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pp. 1273–1282, 2017.

[2] European Commission. EU guidelines on ethics in artificial intelligence: Context and implementation. 2019. URL: https://www.europarl.europa.eu/thinktank/en/document.html?reference=EPRS_BRI(2019)640163 . Accessed 23.11.2022.

[3] General Data Protection Regulation. 2016. https://eur-lex.europa.eu/eli/reg/2016/679/oj . Accessed 14.11.2022.

[4] California Consumer Privacy Act. 2018. https://oag.ca.gov/privacy/ccpa . Accessed 11.11.2022.

[5] Japan’s data protection law, the Act on the Protection of Personal Information, 2019 ttps://www.ppc.go.jp/files/pdf/Act_on_the_Protection_of_Personal_Information.pdf . Accessed 11.11.2022.

[6] Open Mined Explorers Study Group - Use Cases of Differential Privacy. https://blog.openmined.org/use-cases-of-differential-privacy/ Posted on April 30th, 2020.

[7] R. L. Rivest, L. Adleman, and M. L. Dertouzos, “On data banks and privacy homomorphisms,” *Foundations of Secure Computation, Academia Press*, pp. 169–179, 1978.

[8] A. C. Yao, “Protocols for secure computations,” *in Symposium on Foundations of Computer Science*, pp. 160–164, 1982.

[9] R. Agrawal and R. Srikant, “Privacy-preserving data mining,” in *ACM SIGMOD International Conference on Management of Data*, pp. 439–450, 2000.

[10] J. Vaidya, H. Yu, and X. Jiang, “Privacy-preserving SVM classification,” *Knowl. Inf. Syst.*, vol. 14, no. 2, pp. 161–178, 2008.

[11] J. So, B. Guler, and A. S. Avestimehr, “Byzantine-resilient secure federated learning,” *in IEEE Journal on Selected Areas in Communication*, Series on Machine Learning for Communications and Networks, pp. 2168–2180, 2020.

[12] P. Vanhaesebrouck, A. Bellet, and M. Tommasi, “Decentralized collaborative learning of personalized models over networks,” in AISTATS, 2017.

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

[14] Y. Qu, S.R. Pokhrel, S. Garg, L. Gao, Y. Xiang, A blockchained federated learning framework for cognitive computing in industry 4.0 networks. IEEE Trans. Ind. Inform. https://doi.org/10. 1109/TII.2020.3007817

[15] M. Song et al., Analyzing user-level privacy attack against federated learning. IEEE J. Sel. Areas Commun. 38(10), 2430–2444 (2020). https://doi.org/10.1109/JSAC.2020.3000372

[16] X.Wu, Z.Wang, J. Zhao, Y. Zhang, Y.Wu, FedBC: blockchain-based decentralized federated learning, in 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China (2020), pp. 217–221. https://doi.org/10.1109/ICAICA50127.2020.9182705

[17] K. A. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. M. Kiddon, J. Konecný, S. Mazzocchi, B. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, and J. Roselander, “Towards federated learning at scale: System design,” *in SysML 2019*, 2019. [Online]. Available: https://arxiv. org/abs/1902.01046.

[18] Q. Yang,Y. Liu, T. Chen,Y. Tong, Federatedmachine learning: concept and applications. ACM Trans. Intell. Syst. Technol. (TIST) 10(2), 1–19 (2019)

[19] Train Network Using Federated Learning. https://www.mathworks.com/help/deeplearning/ug/train-network-using-federated-learning.html#responsive_offcanvas

[20] K. Hightower, B. Burns, and J. Beda. 2017. Kubernetes: Up and Running: Dive into the Future of Infrastructure. O’Reilly Media.

[21] A. Koschel, M. Bertram, R. Bischof, K. Schulze, M. Schaaf and I. Astrova, "A Look at Service Meshes," 2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA), Chania Crete, Greece, 2021, pp. 1-8.

[22] Sabharwal, Navin & Pandey, Sarvesh & Pandey, Piyush. (2021). Infrastructure-as-Code Automation Using Terraform, Packer, Vault, Nomad and Consul: Hands-on Deployment, Configuration, and Best Practices. 10.1007/978-1-4842-7129-2.

[23] McMahan, H. Brendan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. "Communication-Efficient Learning of Deep Networks from Decentralized Data." Preprint, submitted. February, 2017. https://arxiv.org/abs/1602.05629.

[24] Train Network Using Federated Learning in Matlab, Mathworks, https://nl.mathworks.com/help/deeplearning/ug/train-network-using-federated-learning.html

[25] Pavlove, F. - Lúčny, A. - Malkin-Ondik, I.- Krammer, P. - Kvassay, M. - Hluchý, L.: Efficient deep learning methods for automated visibility at airports. In 2022 Cybernetics & Informatics (K&I) : 31st International Conference. - Danvers, US : IEEE, 2022, p. 1-7. ISBN 978-1-6654-8775-7.

[26] Cushing, R., Valkering, O., Belloum, A., Madougou, S., Bobak, M., Habala, O., Tran, V., Meizner, J., Nowakowski, P., Graziani, M. and Mueller, H., 2021. Process data infrastructure and data services.In Computing and informatics, 2020, vol. 39, no. 4, p. 724-756. ISSN 1335-9150. https://doi.org/10.31577/CAI_2020_4_724