Comparative Study of Federated Learning Frameworks NVFlare and Flower for Detecting Thermal Bridges in Urban Environments

Leonhard Duda, Khadijeh Alibabaei, Elena Vollmer, Leon Klug, Mishal Benz, Valentin Kozlov, Rebekka Volk, Markus Götz, Frank Schultmann, Achim Streit

Karlsruhe Institute of Technology

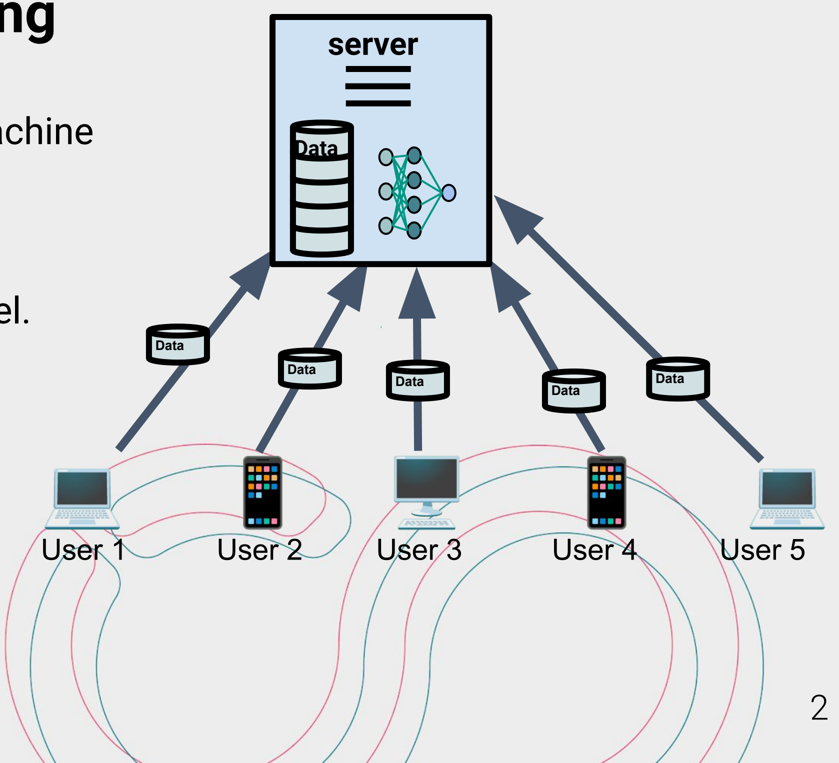
Funded by the European Union

22 | 04 | 2024 by K. Alibabaei

Centralized Learning in Machine Learning

- Refers to the traditional approach where all data is gathered and stored in a central location to train a machine learning model.

- Involves collecting and combining data from multiple sources into a single dataset before training the model.



Funded by
the European Union

Centralized Learning in Machine Learning: Challenges

- **Data Flow Management**: Manage the transfer of **large volumes** of diverse data quickly and accurately across different organizations.
- **Scalability**
- **Communication Overhead**
- **Intense competition within the industry.**
- **Data Privacy**: Ensuring compliance with strict data protection regulations, such as the GDPR[1] and EU AI ACT[2].

1. https://gdpr-info.eu/
2. https://artificialintelligenceact.eu/the-act/

Funded by the European Union

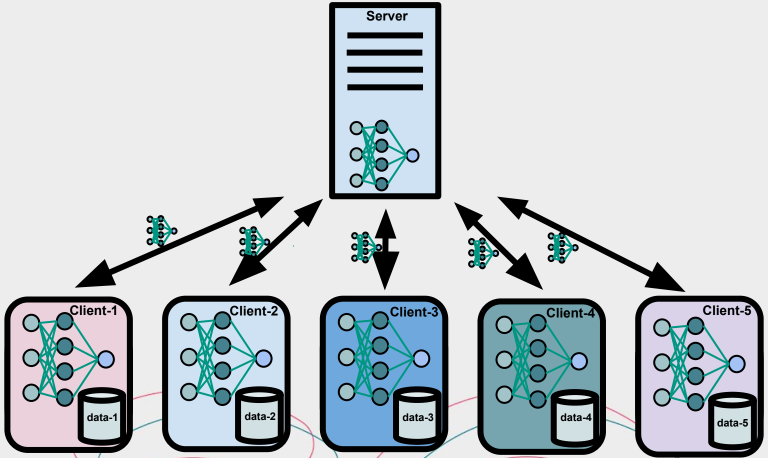
AI4 | eosc

Karlsruher Institut für Technologie

Federated Learning in Machine Learning

A method that facilitates multiple peers to collaboratively learn a common prediction model by exchanging model weights while keeping the sensitive data on the local devices

(Kairouz et al. (2021) and Khan et al. (2023))



Funded by
the European Union

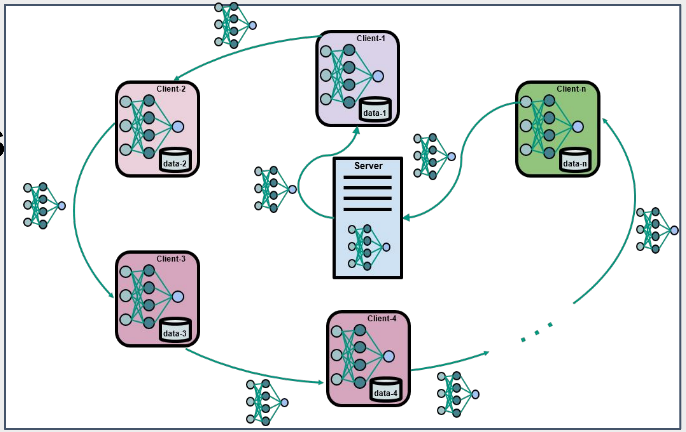
Model Aggregation

Model Aggregation in FL is a further development of distributed learning that is specifically tailored to the challenges of **unbalanced** and **non-independent**, **non-identically distributed data (non-IID)**.

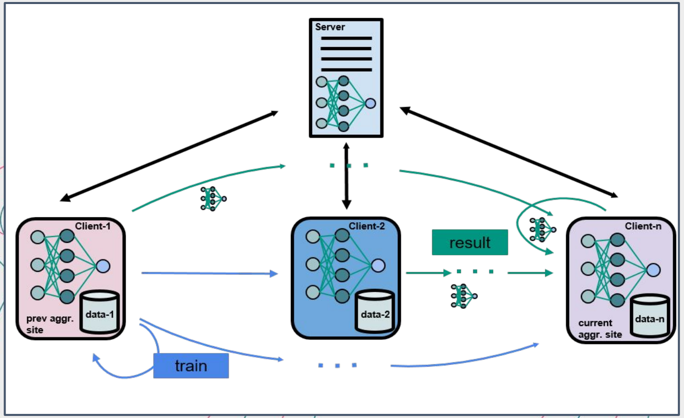
- **FedAvg**: Local weights are collected and aggregated again after local training, using weighted average.
- **FedProx**: Loss function added to penalize the local weights of clients deviating from the global model.
- **FedOpt**: Added option of using a specified Optimizer and Learning Rate Scheduler when updating the global model (like SGD to aggregate the weights of the model).
- **Scaffold**: Added correction term to the model weights after each epoch of local training to prevent them from deviating too much from the global weights

Workflow in FL: Communication Strategies

- **Scatter and gather**: global model parameters are distributed to client devices for local training; updated parameters are then aggregated.
- **Cyclic Weight Transfer** (Chang, K., et al. (2018)): the server selects a subset of clients. Training is done following a predetermined sequential order set by the server.
- **Swarm Learning** (Warnat-Herresthal, S. (2021)): a decentralized subset of FL where orchestration and aggregation is performed by the clients



Centralized cyclic Learning

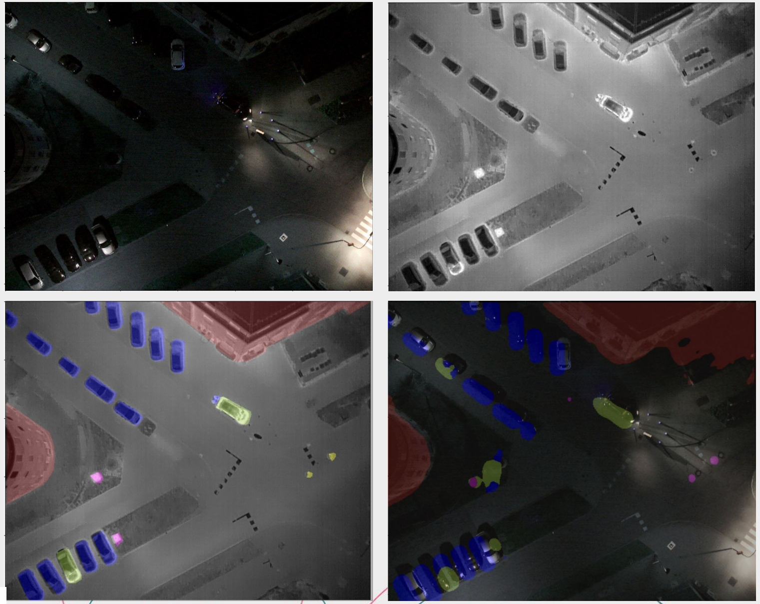


Swarm Learning

Funded by the European Union

Detecting Thermal Bridges in Urban Environments

- Identifying thermal anomalies in urban environments to improve the efficiency of energy-related systems.
- U-Net with ResNet-152 backbone
- Images of Karlsruhe and Munich
  - 700 images from Munich
  - 93 images from Karlsruhe
-



Example of thermal urban feature segmentation (I): combined RGB (top left) and TIR (top right) inputs, manual segmentation mask (bottom left), and U-Net model prediction (bottom right) Vollmer, E. (2023).

Model FL Frameworks

- **Flower**:
  - is a flexible, easy-to-use and easily understood open-source FL framework.
  - The server is provided by the AI4EOSC project as a tool (Secure personalized federated learning within the AI4EOSC platform ,2 Oct 2024, 14:30, Judith Sainz-Pardo Diaz)
- **NVIDIA Federated Learning Application Runtime Environment (NVFlare)**:
  - NVFlare is a business-ready FL framework by Nvidia.
  - Plan to be add to the Platform provided by the AI4EOSC project

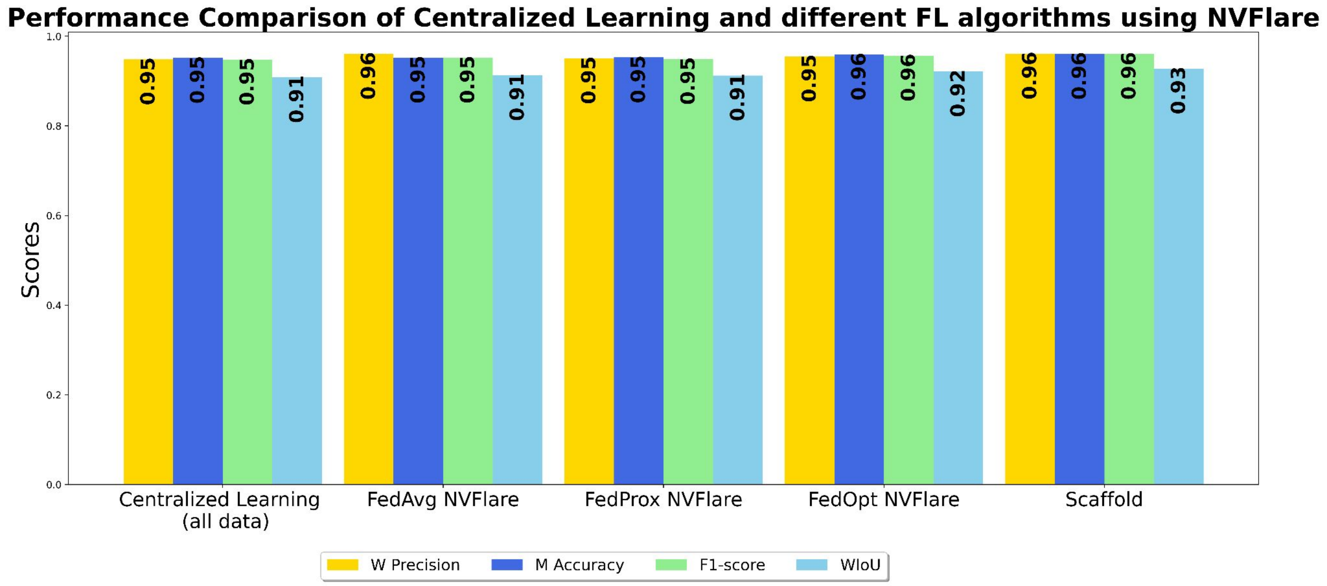
Flower: A Friendly Federated Learning Framework

Funded by the European Union

AI4 eosc

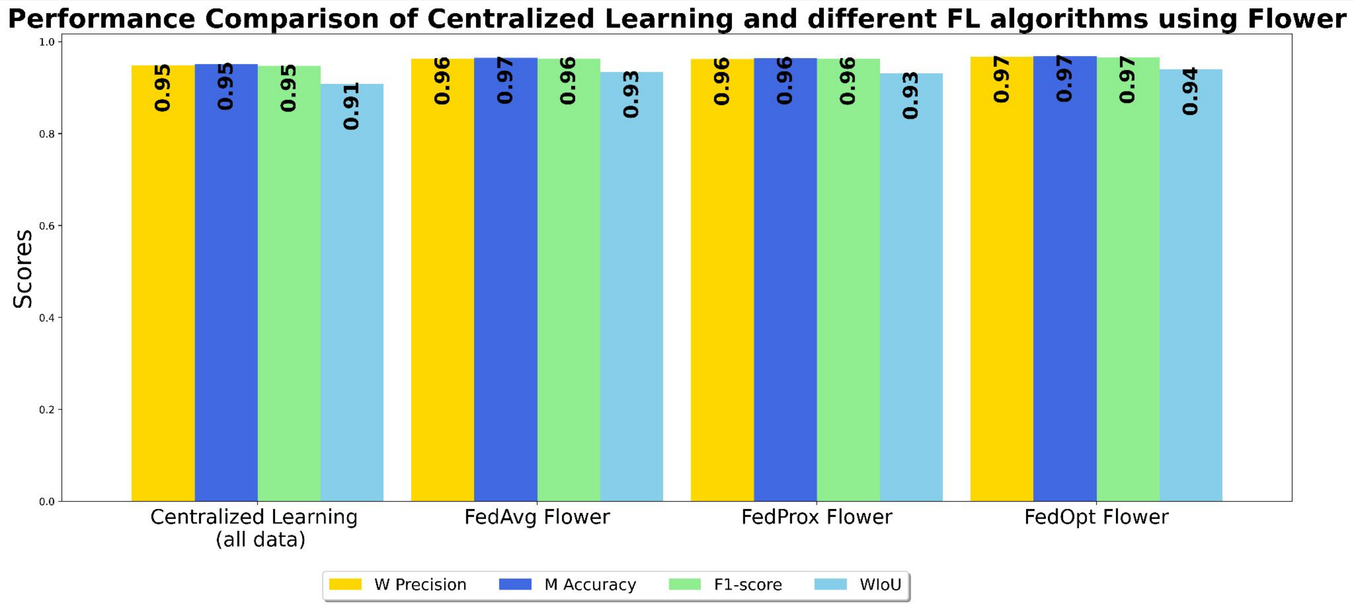
Karlsruher Institut für Technologie

Scatter & Gather using different algorithms in NVFlare



Funded by the European Union

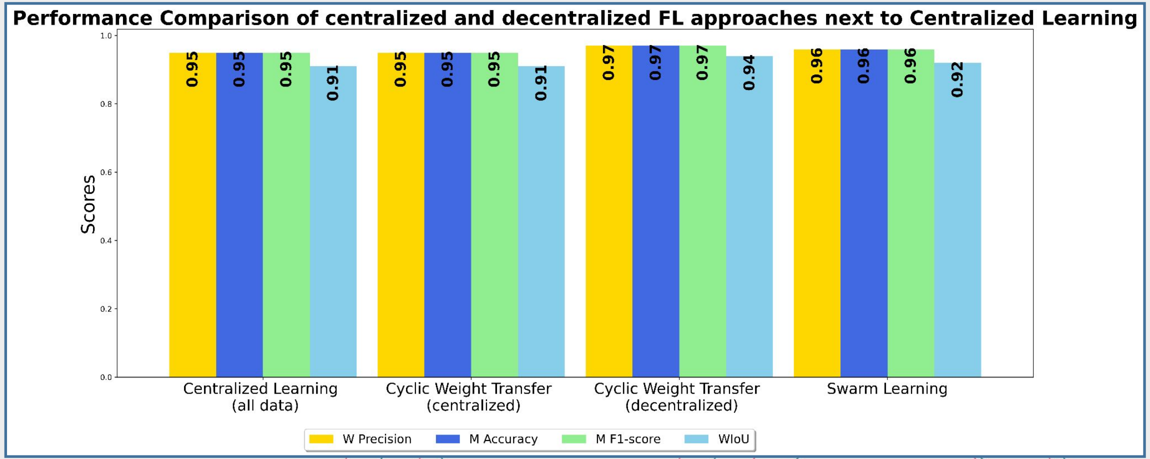
Scatter & Gather using different algorithms in Flower



Funded by
the European Union

Cyclic Weight Transfer and Swarm Learning

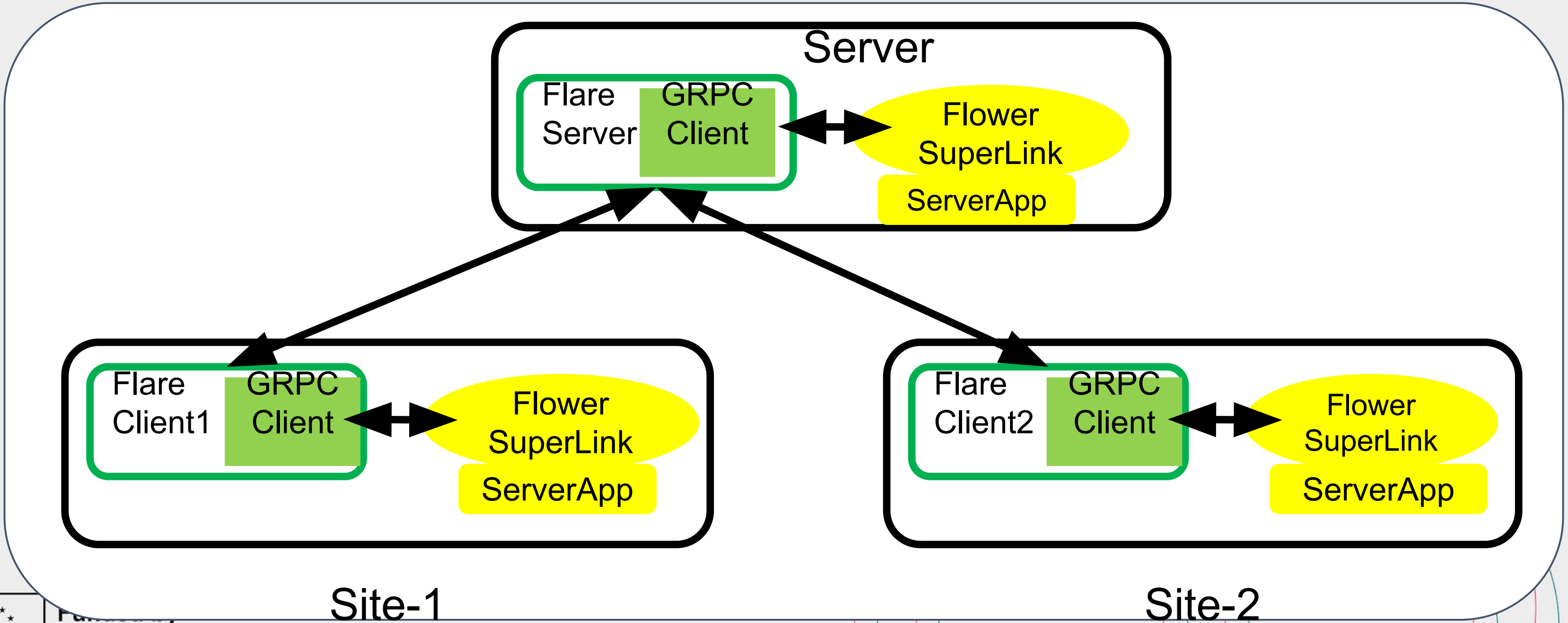
- ➢ Using Cyclic Weight Transfer as an approach, the order of the clients have a big impact on the overall results
- ➢ Using Decentralised FL is removed some communication overhead and so is faster
- ➢ Looking at the metrics during training, Swarm Learning is more stable than Cyclic weight transfer learning



Personal Insights: Comparing Flower and NVFlare

Feature	NVFlare	Flower
Ease of Use	hard, depending on the model and which ml framework is used	Easier due to different architecture (client-file and server-file)
Privacy features	Offers more secure communication methods and features	Less features, HE not implemented
Experiment tracking	Offers proprietary functionality for MLFlow and TensorBoard	Experiment tracking needs to be implemented manually
Workflows	Offers “decentralized” workflows (Swarm, Cyclic)	No such workflows
Algorithms	“Just” a few implemented algorithms	No Scaffold, but a variety of different algorithms
support	Direct connect with NVflare developer and direct support from them	Did not try
orchestration	we can monitor the progress of a submitted job and client and server status from Admin console	No

NVFlare and Flower Collaboration [17]



Conclusions

- In our case of two distributed datasets Federated Learning can keep up with traditional Centralized Learning
- In our case of two unequal distributed dataset (7:1 ratio), Scaffold performs best when using a Scatter & Gather workflow
- When privacy is not a priority, Flower is the better solution as it's easier to setup and to use. Otherwise NVFlare offers more features (DP, HE, Provisioning)
- With the new collaboration between these two framework, some features can now be shared across each other.

Funded by the European Union

References

Behera, S., & Prathuri, J. R. (2020). Application of Homomorphic Encryption in Machine Learning. In 2020 2nd PhD Colloquium on Ethically Driven Innovation and Technology for Society (PhD EDITS) (pp. 1-2). Bangalore, India. https://doi.org/10.1109/PhDEDITS51180.2020.9315305

Chang, K., Balachandar, N., et al. (2018). Distributed deep learning networks among institutions for medical imaging. *Journal of the American Medical Informatics Association, 25*(8), 945-954. https://doi.org/10.1093/jamia/ocy017

Fung, C., Yoon, C.J., Beschastnikh, I., (2018). Mitigating sybils in federated learning poisoning. arXiv preprint arXiv:1808.04866 .

Geiping, J., Bauermeister, H., Dröge, H., & Moeller, M. (2020). Inverting Gradients -- How easy is it to break privacy in federated learning? [Preprint]. arXiv. https://arxiv.org/abs/2003.14053

Holger R. Roth, et.al. (2022). NVIDIA FLARE: Federated Learning from Simulation to Real-World. arXiv. https://arxiv.org/abs/2210.13291

Jagielski, M., Oprea, A., Biggio, et.al. (2018). Manipulating machine learning: Poisoning attacks and countermeasures for regression learning, in: 2018 IEEE Symposium on Security and Privacy (SP), IEEE. pp. 19–35.

References

Kairouz, P., McMahan, H. B., Avent, et al. (2021). Advances and Open Problems in Federated Learning. https://ieeexplore.ieee.org/document/9464278

Khan, M., Glavin, F. G., & Nickles, M. (2023). Federated Learning as a Privacy Solution - An Overview. Procedia Computer Science, 217, 316-325. https://doi.org/10.1016/j.procs.2022.12.227

Narayanan, A., & Shmatikov, V. (2008). Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (SP 2008)* (pp. 111-125). Oakland, CA, USA. https://doi.org/10.1109/SP.2008.33

Melis, L., Song, C., De Cristofaro, E., & Shmatikov, V. (2019). Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)* (pp. 691-706). San Francisco, CA, USA. https://doi.org/10.1109/SP.2019.00029

Muto, R., et.al. (2022). Predicting oxygen requirements in patients with coronavirus disease 2019 using an artificial intelligence-clinician model based on local non-image data. *Frontiers in Medicine (Lausanne)*, 9, 1042067. https://doi.org/10.3389/fmed.2022.1042067

References

Li, T., Hu, S., Beirami, A., & Smith, V. (2021). Ditto: Fair and Robust Federated Learning Through Personalization. arXiv. https://arxiv.org/abs/2012.04221

Liu, Y., et al. (2023). Vertical Federated Learning: Concepts, Advances, and Challenges. *IEEE Transactions on Knowledge & Data Engineering*, 01(01), 1-20. https://doi.org/10.1109/TKDE.2024.3352628

Lu, S., Li, R., Liu, W., Guan, C., & Yang, X. (2023). Top-k sparsification with secure aggregation for privacy-preserving federated learning. *Computers & Security*, 124, 102993. https://doi.org/10.1016/j.cose.2022.102993

Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S., & McMahan, H. B. (2021). Adaptive Federated Optimization. arXiv. https://arxiv.org/abs/2003.00295

Sai Praneeth Karimireddy, et al. (2021). SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. *arXiv.* https://arxiv.org/abs/1910.06378

Tian, L., Kumar Sahu, A., Talwalkar, A. S., & Smith, V. (2020). Federated Learning: Challenges, Methods, and Future Directions. IEEE Signal Processing Magazine, 37.

References

Olaf Ronneberger, Philipp Fischer, & Thomas Brox. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv: https://arxiv.org/abs/1505.04597

Tian Li, Anit Kumar Sahu, et al. (2020). Federated Optimization in Heterogeneous Networks. *arXiv*. https://arxiv.org/abs/1812.06127

Truong, N., Sun, K., Wang, S., Guitton, F., & Guo, Y. (2021). Privacy preservation in federated learning: An insightful survey from the GDPR perspective. Computers & Security, 110, 102402. https://doi.org/10.1016/j.cose.2021.102402

Zapechnikov, S. (2022). Secure multi-party computations for privacy-preserving machine learning. Procedia Computer Science, 213, 523-527. https://doi.org/10.1016/j.procs.2022.11.100

Vollmer, E. (2023). UAV-based thermography: Using AI with multispectral data. Vortrag gehalten auf ANERIS Workshops on AI Basics for Image Processing (2023), Online, 28. November–7. Dezember 2023. DOI: 10.5445/IR/1000166038

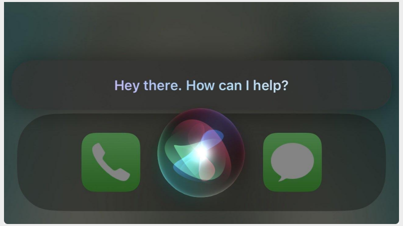
Warnat-Herresthal, S., Schultze, H., Shastry, K. L., et al. (2021). Swarm Learning for decentralized and confidential clinical machine learning. *Nature, 594*, 265–270. https://doi.org/10.1038/s41586-021-03583-3

Funded by
the European Union

More Examples of successful applications of FL

- Apple has employed federated learning to improve Siri's voice recognition capabilities while maintaining user privacy[1].

- Predicting oxygen requirements for COVID-19 patients in the ER using chest X-rays and health recorde (Muto, R., et.al. (2022)).



Source: Holger R. Roth, et.al (2023)

1. https://www.technologyreview.com/2019/12/11/131629/apple-ai-personalizes-siri-federated-learning/

Funded by the European Union

Results:

Pull request for our implementation of Scaffold and FedProx [15]

Tensorflow support

With community contributions, we add FedOpt, FedProx and Scaffold algorithms using Tensorflow to create parity with Pytorch. You can them here.

Changelog of the new release 2.5 from 9th September 2024 [16]

Categories Federated Learning

Federated Learning can be categorized as (Khan et al. (2023))**:**

- **Data distribution**
  - **Cross devices**: the model is decentralized across the edge devices and is trained using the local data on each device.
  - **Cross silos**: where the clients are a typically smaller number of organizations, institutions, or other data silos.
- **Architecture**
  - **Centralized Federated Learning**: server coordinates the training
  - **Decentralized Federated Learning**: the communication is peer to peer
- **Learning model**
  - **Horizontal Federated Learning**: each party has the same feature space but different data samples.
  - **Vertical Federated Learning**: datasets of each party share the same samples/users while holding different features (Liu, Y., et al. (2023)).

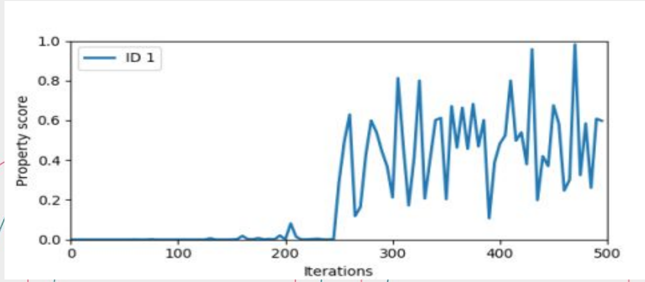
Possible Issues with Federated Learning!

Reconstruction attack (Truong et al. (2021)) :

- The original training data samples can be reconstructed from the model weights.

- membership tracing i.e., to check if a given data point belongs to a training dataset, or when a participant whose local data has a certain property, joined collaborative training.

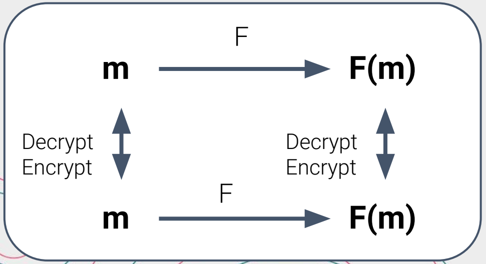
Reconstructing an input image using the gradient.. On the left: Image extracted from the validation dataset. In the middle: Reconstruction generated by a ResNet-18 model trained on ImageNet Right: Reconstruction from a trained ResNet-152. **Geiping, J. et.al, (2020)**



Inferring that a participant whose local data has the property of interest has joined the training. **Melis, L. et.al, (2019)**

Solutions

- **Data Anonymization :** a technique to hide or remove sensitive attributes, such as personally identifiable information (PII) (Narayanan, A.& Shmatikov, V. (2008)).
- **Differential Privacy (DP)**[1]:
  - It provides a formal definition of privacy by introducing noise to query responses to prevent the disclosure of sensitive information.
- **Homomorphic Encryption (HE)** (Behera et al. (2020)): allows computations to be performed on encrypted data.

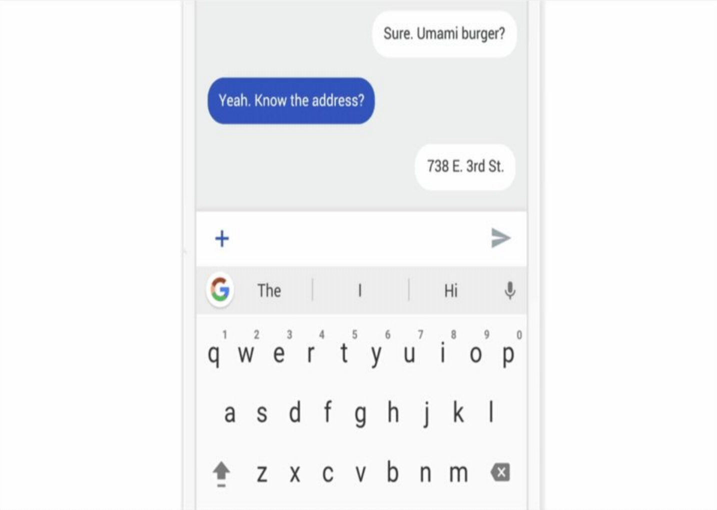


1. https://github.com/google/differential-privacy

Funded by
the European Union

Examples of successful applications of FL

Google already used FL in Gboard Android:
When Gboard suggests a query, your phone stores context and interactions locally. Federated Learning uses this to improve Gboard's suggestions.

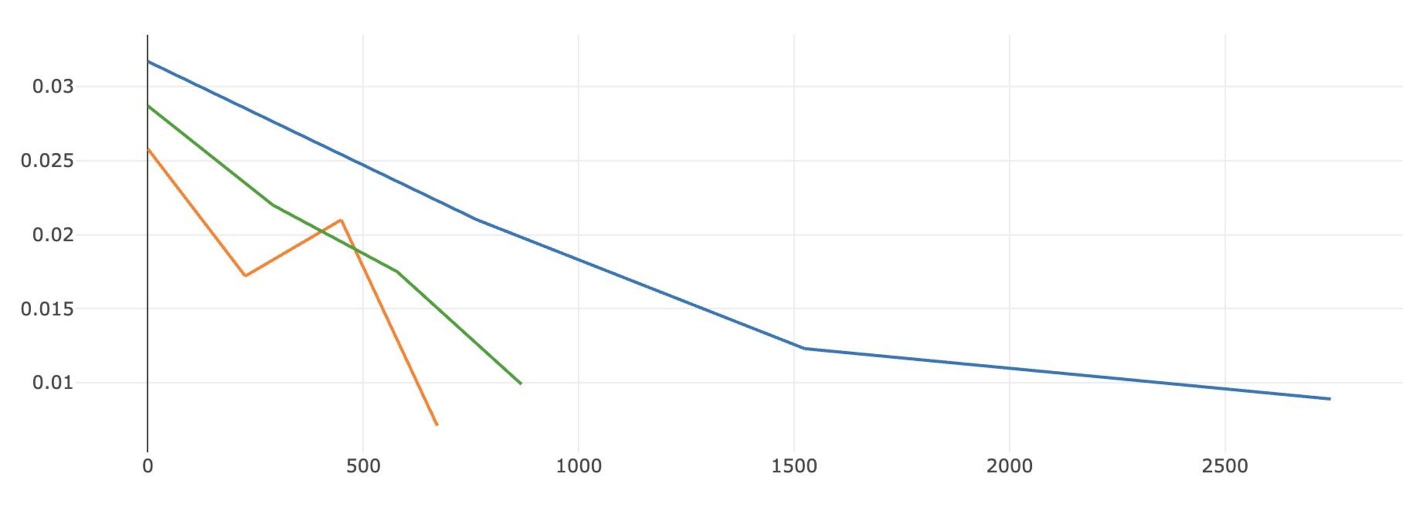


https://research.google/blog/federated-learning-collaborative-machine-learning-without-centralized-training-data/

Funded by the European Union

AI4 | eosc

Cyclic Weight Transfer and Swarm Learning



AI4 | eosc

From Centralized Learning to Federated Learning

- Adjust the code to make use of the Federated Learning Features of NVFlare
    - For each FL approach adjustments to the code were necessary
    - some algorithms needed to be implemented manually from scratch for Tensorflow
- Try the implemented approaches with the simulator
    - Run the simulation of two clients and a server on one HPC system
- Go from simulation to real world environment
    - Deploy one client on HoreKa, one on HAICORE and set up a server on the bwCloud
    - Write the batch scripts for the usage of the clients
- Train a model for each approach in the real world setup and track the metrics
- Try out Flower and adjust the initial Centralized Learning code
    - Try the same algorithms used in NVFlare for a comparison
- Evaluate and compare the results

Funded by
the European Union

AI4 eosc

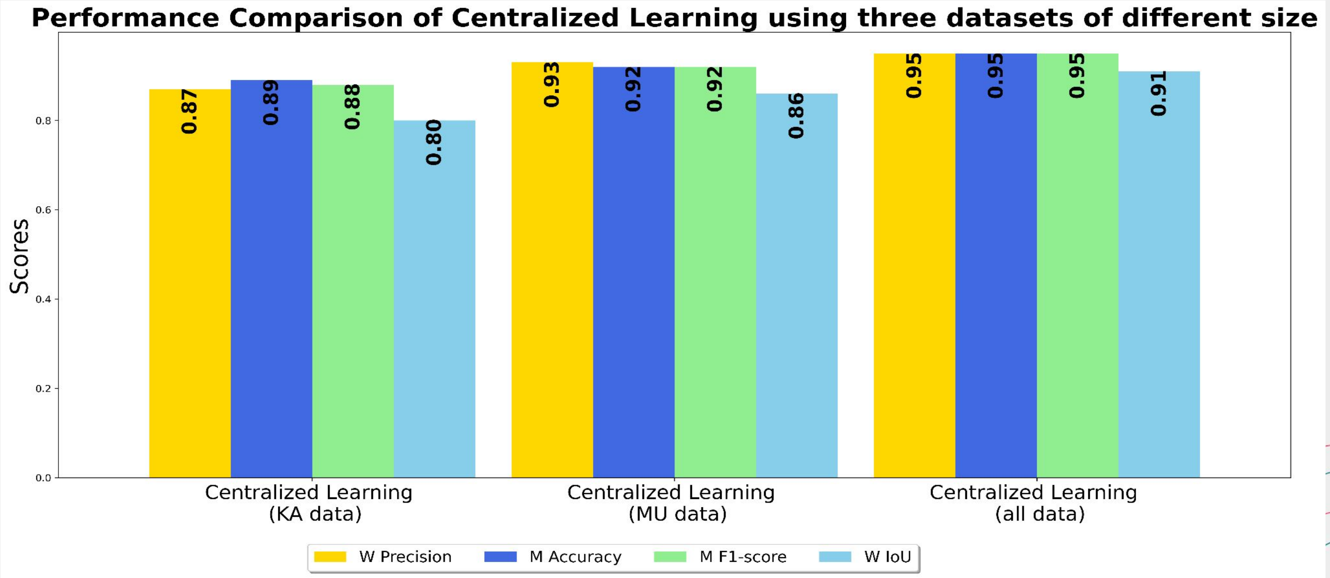
Detecting Thermal Bridges in Urban Environments

- ➢ Semantic segmentation model
- ➢
- ➢ 8001 annotations

Class	# Annotations	# Pixels (*10^3)
Background	-	37 063.96
Building	1404	9 087.95
Car (cold)	2531	601.90
Car (warm)	1034	325.60
Manhole round	1536	50.51
Manhole square	358	12.79
Miscellaneous	81	8.38
Person	275	7.64
Street Lamp	782	27.18

Funded by
the European Union

AI4 eosc


Performance Comparison of Centralized Learning using three datasets of different size
Scores
0.0
0.2
0.4
0.6
0.8
0.87
0.89
0.88
0.80
0.93
0.92
0.92
0.86
0.95
0.95
0.95
0.91
Centralized Learning (KA data)
Centralized Learning (MU data)
Centralized Learning (all data)
W Precision
M Accuracy
M F1-score
W IoU

Parameters for training CL against FL

Parameters	Centralized Learning	Federated Learning
Devices	1 Client (HoreKa)	2 Clients (HoreKa & HAICORE) and 1 server (bwCloud)
Rounds of training	1	4
(Local) Epochs	35	9
Batch Size	8	8
Optimizer	Adam	Adam
Learning Rate	0.01	0.01
Loss function	Focal Loss	Focal Loss