IFC-Bank of Italy Workshop on "Data science in central banking: enhancing the access to and sharing of data"

17-19 October 2023

Exploring aggregation strategies for federated learning in national statistics[1]

Mauro Bruno, Erika Cerasti, Massimo De Cubellis and Francesco Pugliese,
Italian National Institute of Statistics - ISTAT

Rafik Chemli, Benjamin Santos and Julian Templeton,
Statistics Canada

Matjaz Jug,
Statistics Netherlands

[1] This contribution was prepared for the workshop. The views expressed are those of the authors and do not necessarily reflect the views of the Bank of Italy, the BIS, the IFC or the other central banks and institutions represented at the event.

Exploring Aggregation Strategies for Federated Learning in National Statistics

Authors: Benjamin Santos, Julian Templeton and Rafik Chemli (Statistics Canada), Erika Cerasti, Francesco Pugliese, Mauro Bruno and Massimo De Cubellis (ISTAT), Matjaz Jug (Statistics Netherlands)

ABSTRACT

Federated Learning (FL) can open opportunities for analytics to be generated on sensitive, distributed data. National Statistical Offices (NSOs) can hold a wealth of information which others could use to generate better statistics. By considering adopting FL, NSOs can explore the feasibility of collaborating with other NSOs to use Machine Learning (ML) models as a method to generate analytics from their distributed data sources. Within an FL environment, the aggregation strategy used by the ML model holder is a key component in ensuring that the ML model is accurately updated after each training round. This work explores a weighted Federated Averaging aggregation scheme which aims to highlight the potential benefits of positively weighing clients who contain more homogeneous data. The preliminary results exhibit that when clients with more homogeneous data are weighed higher, the global model can achieve better performance, but that a sufficiently complex FL task is needed for these effects to be more prevalent. Clients with more heterogeneous data exhibit generally worse performance in the simulations conducted. Furthermore, Homomorphic Encryption can be used as another defense within FL scenarios at high computational and memory costs. While shallow ML models can mitigate this issue to an extent, their performance can also degrade depending on the number of operations performed. Future studies will highlight the effects of data homogeneity within FL experiments on more complex and realistic environments.

Keywords: Federated Learning, Homomorphic Encryption, Privacy Enhancing Technologies

Introduction

In an era increasingly reliant on data, official statistics plays a pivotal role in the fabric of modern society. Published statistics provide essential insights into various aspects of life, including economic performance, population demographics, health trends, and environmental changes. The integrity, accuracy, and reliability of these statistics are crucial for maintaining public trust and enabling governments, businesses, and individuals to make evidence-based decisions that can impact society itself. The advent of big data has significantly impacted official statistics and National Statistical Offices (NSOs). The sheer volume, variety, and velocity of data being generated presents both unprecedented opportunities and formidable challenges. Big data offers a richer, more granular view of complex phenomena, potentially enhancing the depth and breadth of statistical analysis. However, this paradigm shift requires a re-evaluation and adaption of traditional statistical methodologies. The conventional approaches to data collection and analysis are being tested by the complexities inherent in big data, including issues related to data quality such as relevance, timeliness, and privacy protection.

Amongst these challenges, new methodologies, such as input privacy preservation, have become increasingly important. Input privacy refers to techniques and practices designed to protect the confidentiality and integrity of data sources, particularly when dealing with privately held information. As statistical agencies increasingly turn to privately held data to complement or enhance traditional data sources, the need to safeguard individual and organizational privacy becomes imperative. Input privacy approaches involve a range of techniques, from data anonymization and encryption to more sophisticated approaches such as Homomorphic Encryption (HE) and Secure Multi-Party Computations [1]. NSOs also use output privacy techniques such as differential privacy and statistical disclosure control. The proper adoption of these methodologies, following extensive research of their utility, is essential in ensuring that the analysis of privately held data is conducted responsibly, ethically, and effectively.

Although NSOs would benefit from holding more information from other parties, including other NSOs, this data may be private or legally protected, and its direct usage may be a point of conflict. Although legislation can exist to permit NSOs access to private datasets, avoiding forceful access can help further build trust when possible. NSOs themselves also have a duty to protect data, where laws can enforce this. To get insights from data shared with other organizations, Federated Learning (FL) is a potential solution since it trains models with an organization's data without sharing the data itself. Within the United Nations Privacy Enhancing Technologies (PET) Lab, ISTAT, Statistics Canada, and Statistics Netherlands have been researching the capabilities of FL. Within this work, we have explored different federated aggregation strategies, including weighted strategies which positively or negatively emphasize learning from homogeneous data sources.

This paper will first outline the various techniques being tested. This includes an overview of FL and HE, which can further enhance privacy when used with FL. The experiment methodology is then described, with the key results presented and discussed. Finally, future work this group can explore within the UN PET Lab is outlined.

Background Information

Understanding Federated Learning

FL is a Machine Learning (ML) methodology which enables ML models to be trained across distributed devices while keeping the training data stored locally on each device [2, 3]. Using FL, an organization can train a ML model held by some central authority without requiring the training data to be shared with the authority. This permits analytics to be derived from distributed private data sources. During the training process, the global model held by the central authority is sent to all clients (individuals or organizations) participating in training that model for the training round. The locally updated models are then sent to and aggregated by the central authority who stores the updated global model for future use or further training. Thus, the training process ensures that the private client data does not leave that client. The central authority's role consists of aggregating the updates from the clients to potentially improve the global model's performance [4]. If the central authority has an isolated testing set, the updated model can be tested before committing it as the main model. The updated models can also be version controlled to ensure that data poisoning attacks or concept drifts are appropriately handled.

The performance of the distributed trained model relies on the careful selection of its hyperparameters, the quality and quantity of the distributed training data, the number of clients participating in training, the ability to evaluate the model updates, and the aggregation strategy utilized to combine local models from each training round. There are a variety of aggregation strategies which can be used in FL, which each vary in complexity and utility. Some popular strategies include Federated Averaging (FedAvg) [5], Federated Adaptive Gradient (FedAdagrad) [6], Federated Adam (FedAdam) [7], and Federated Yogi (FedYogi) [6]. Each of these have pros and cons depending on the type of data being held by the clients. Within this paper we will explore strategies to weight FedAvg, which simply computes the average of each local model's weights, by assigning weights to each local model. This can allow models to be weighed higher based on different heuristics. We consider two scenarios:

1. Where models trained with more homogeneous data are weighed higher.
2. Where models are weighed higher when trained with more heterogeneous data.

The purpose of exploring these two opposite strategies is to understand the impact of prioritizing clients with more homogeneous data during the training rounds. If a client lacks homogeneous data in a round, it may be worth excluding the client for the round or using a subset of the data available for training. These two weighing schemes will be respectively referenced as HWFedAvg (homogeneously weighted FedAvg) and NWFedAvg (non-homogeneously weighted FedAvg). Full details on these schemes will be described later in the paper.

Although FL permits analytics previously impossible to be generated, not all privacy risks are mitigated [8]. In fact, locally trained client models can still be attacked if not appropriately protected. This can lead to privacy leaks such as the training set being reconstructed by the central authority. Fortunately, other PETs can be used alongside FL to further protect local models. Homomorphic Encryption (HE) is a PET that can provide additional protection at the cost of the total compute and memory needed [9].

Understanding Homomorphic Encryption

HE is a public-key encryption scheme which enables certain arithmetic functions to be calculated between encrypted data and another data source without needing to decrypt the encrypted data [10]. The outputs of these functions are also encrypted, where the plaintext is the correct result of the applied function. There are different types of HE algorithms, each offering varying degrees of functionality and security (at the cost of the corresponding complexity). Partially HE supports either addition or multiplication operations, but not both, to be applied with the encrypted data [11]. Somewhat HE supports a limited number of both addition and multiplication operations on encrypted data, with restrictions on the number of operations that can be performed [12]. Fully HE is the most robust form of HE which supports both addition and multiplication without the previously mentioned restrictions [13].

While we will explore HE in an FL scenario, it has many important uses. For example, it can securely perform computations on encrypted data in delegated computing scenarios (such as if companies need to send data to a competitor's cloud platform for aggregation by an NSO) [14]. Within Secure Multi-Party Computations, HE can enable multiple parties to jointly compute a function over their encrypted inputs [15]. Furthermore, it can allow privacy-preserving data analysis to be performed on encrypted data, which is important in sensitive domains such as healthcare. Other encryption approaches can help with generating data analytics, such as Functional Encryption, however within FL, HE best enhances a local model's privacy when sent for aggregation since its outputs remain encrypted. With HE, the central authority can send an encrypted model to be trained, then securely aggregate the encrypted local model weights to update the model. This ensures that the aggregation step remains secure for all clients. The corresponding memory and computational costs are high when used, but if feasible, the resulting analytics can be worth it.

Importance of Considering Data Homogeneity

A key challenge in FL is that client data cannot be observed directly by the central authority. Thus, there can be quality issues with the data being used for training. This can lead to data poisoning attacks, which attempt to reduce the model's performance. Within this paper we consider the scenarios in which client data is more or less homogeneous. When data is heterogeneously distributed, local models tend to diverge from the optimal global model. In these cases, by aggregating divergent local models, the performance of the global model may degrade. FedAvg does not address the issue of data heterogeneity when aggregating local models. To mitigate the negative impact of data heterogeneity and promote the convergence of the models, several algorithms have been proposed. These can be classified as either *client variance reduction* or *adaptive global model update* [16]. The former aim to improve client-side learning and use a naïve global updating strategy, while the latter focuses on improving server-side learning using a simple Stochastic Gradient Descent-based client learning strategy. This research includes FedYogi in its tests as an adaptive global model algorithm. These types of algorithms differ from FedAvg since they consider the previous training rounds before aggregating the models.

Experiment Methodology

Previously, we compared various aggregation strategies on a public Human Activity Recognition (HAR) dataset in a homogeneous data environment [17]. That work highlighted that FedAvg and FedYogi are effective federated aggregation strategies when applied to a task without hyperparameter tuning. In this work, we explore a weighted aggregation strategy, termed HWFedAvg, which weighs client models based on the homogeneity of their training data. Furthermore, we analyze the effectiveness of positively weighing client models when trained with heterogeneous datasets with the NWFedAvg strategy. The base simulation environment remains the same as the previous work we conducted, where the Flower library is used for the FL components [18].

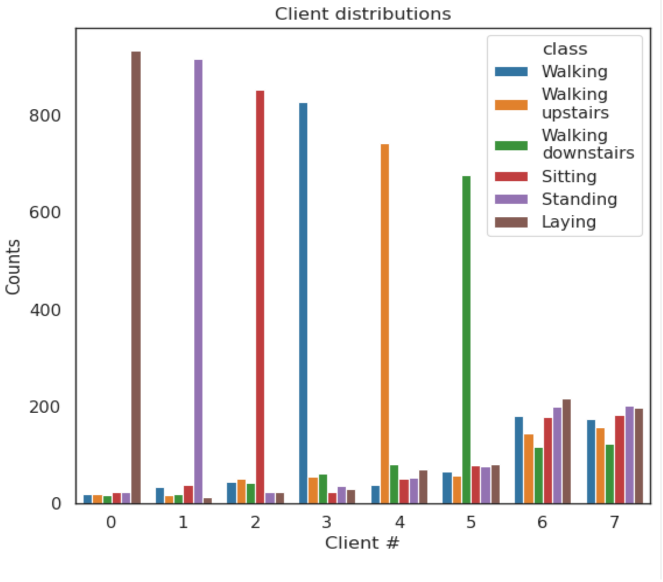
This research utilizes a simplistic HAR dataset comprising of accelerometer and gyroscope smartphone data. The ML model aims to ascertain human activity types from the following six categories: walking, walking upstairs, walking downstairs, sitting, standing, and laying. Within these tests, the HAR dataset has been partitioned among eight clients, aiming to establish a set of clients with a mix of heterogeneous and homogeneous local datasets. We have prepared the following four partitioning methodologies to distribute the data.

- Random: Allocation of samples across clients is randomized.
- Majority Even: The first k clients, where k is the number of classes in the classification task, predominantly holds samples from a unique class. Once all k classes have been uniquely assigned to the first k clients as a majority class, the remaining clients receive a homogeneous distribution of data. Each client holds an equal number of unique records.
- Majority: The same distribution of majority classes as described in Majority Even, however each client can hold a different total number of samples.
- Pick Two: The first $k/2$ clients each hold two unique majority classes (out of k total classes), representing heterogeneous datasets. The remaining clients hold homogeneous datasets with the remaining records. Each client holds an equal number of total records.

The experiments within this paper utilize eight clients, each receiving data using the ‘Majority Even’ method. It is important to note that each unique class is assigned as a majority class to only one client, where any remaining clients receives a balanced distribution of all classes. A visual representation of the data partitions from the Majority Even method is depicted in Figure 1 below. Note that clients zero through five each have one majority class and their datasets are heterogeneous, whereas clients six and seven have homogeneous datasets.

Human activity class distribution among clients using the 'Majority Even' splitting method

Figure 1



Two different sets of experiments will be presented in this paper. The first will be a comparison between FedAvg, HWFedAvg, NWFedAvg, and FedYogi. The second will be an exploration of different performance costs which can arise when applying HE. Specific details on these experiments are outlined below.

Experiment 1 – Understanding the Importance of Data Homogeneity

This experiment aims to understand the impact of clients holding homogeneous datasets and heterogeneous datasets when participating in training rounds. After utilizing the 'Majority Even' splitting algorithm to define the client datasets, two aggregation strategies are tested. The first gives a greater weight to clients with a homogeneous distribution, HWFedAvg (i.e. clients 6 and 7 in Figure 1), and the second prioritizes clients with a heterogenous distribution, NWFedAvg (i.e. clients from 1 to 5 in Figure 1). The details of the HWFedAvg and NWFedAvg aggregation strategies will be described below.

First, we will outline the key variables utilized by weighting algorithms. Variable i represents a single client, where there are eight clients within this experiment, whereas variable j denotes a single class, where six classes exist within the HAR dataset. Each client i will have an associated weight w_i, which is computed based on the data they hold at a given training round r. Every class j, for each client i, also contains a probability $\mathrm{p}_\mathrm{j}^\mathrm{i}$, which represents the probability of client i randomly selecting a sample of class j from their training set. The computation for this probability is presented below.

$$\mathrm{p}_\mathrm{j}^\mathrm{i} = \frac{\mathrm{N}_{\mathrm{i,j}}}{\mathrm{N}_\mathrm{i}}$$

Where, $\mathrm{N_{i,j}}$ is the total number of samples held by client i of class j to be used for training and $\mathrm{N_i}$ is the total number of samples held by client i for the training round.

Utilizing each normalized probability $\mathrm{p_j^i}$, each client will derive a locally held probability vector $\overrightarrow{\mathrm{p_\imath}}$, where $\overrightarrow{\mathrm{p_\imath}} = \{\mathrm{p_1^i}, \ \ldots, \ \mathrm{p}_k^{\mathrm{i}}\}$ for the k available classes. Similarly, a global vector $\vec{P} = \{P_1, \ \ldots, \ P_j\}$ is available to all clients, which contains the probabilities P_j of any class j being selected by any participating client. The equation for P_j described below.

$$P_j = \frac{\mathrm{N_j}}{\mathrm{N}}$$

Where $\mathrm{N} = \sum_i \mathrm{N}_i$ and $\mathrm{N_j}$ is the total number of samples of class j being used in round r.

We then begin constructing a set of weights with the probabilities which have been calculated. Let $\tilde{P}$ be the normalized product of the probabilities of picking a class j in the joint dataset (P_j) times the square probability for a client i to have this class j (p_i^j). This is presented below.

$$\widetilde{P_\imath} = \frac{\sum_j P_j \cdot \left(p_i^j\right)^2}{\sum_i \sum_j P_j \cdot \left(p_i^j\right)^2}$$

These squared probabilities $\widetilde{P_\imath}$ add importance to majority classes with respect to the minority classes. The normalized weights can then be computed as the inverse of these values, as outlined below.

$$w_i = \frac{1/\widetilde{P_\imath}}{\sum_i 1/\widetilde{P_\imath}}$$

These weights can then be scaled by considering sample size N_i, as follows.

$$\alpha_i = \frac{N_i}{N}$$

The final weights can then be computed through the equation below.

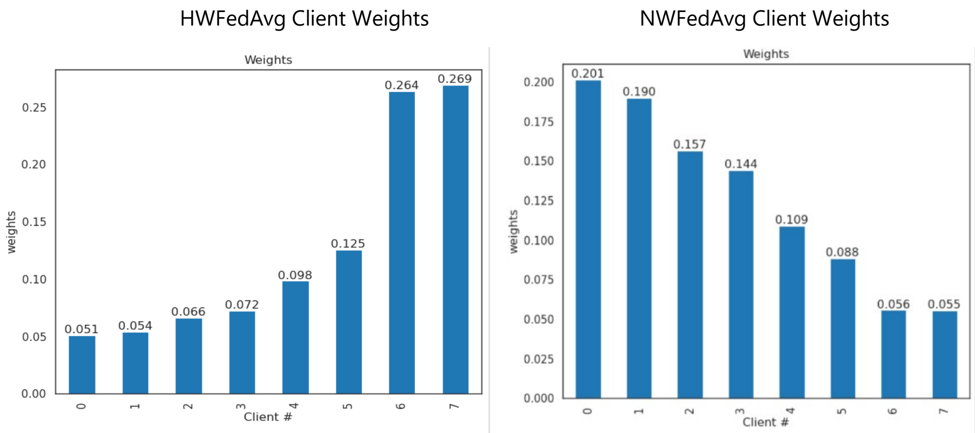
$$W_i = \frac{\alpha_i w_i}{\sum_i \alpha_i w_i}$$

Note that these weights represent adding value to more homogeneous data. Thus, the HWFedAvg strategy uses this equation whereas the NWFedAvg strategy will utilize the inverse of the weights.

With the HWFedAvg and NWFedAvg strategies better understood, examples of how clients weighted using them are presented in Figure 2 below.

Comparison between using HWFedAvg and NWFedAvg

Figure 2



These values are then used to ensure that clients are appropriately weighed on the selected weighting heuristic. We test both homogeneity and heterogeneity to better showcase the importance of data homogeneity in FL scenarios. Full results from tests utilizing these strategies will be presented and discussed later in this paper.

Experiment 2 – Exploring the effects of using Homomorphic Encryption

Previously, we explored using the Paillier public key cryptosystem to securely aggregate the local model weights when using the FedAvg strategy [17]. The Paillier cryptosystem is an additive HE scheme which the central authority can use to average all client model weights when the central authority has access to the public key [19]. Within the Paillier HE scheme, a set of ciphertexts {ct_1, ct_2, ..., ct_m} can be added, then decrypted, such that the ciphertext output ct is the encrypted equivalent to adding the corresponding plaintexts pt = $\text{pt}_1 + \text{pt}_2 + \cdots + \text{pt}_m$, where ct_i is the ciphertext of pt_i (i.e., ct_i=Enc(pt_i)). Now, we extend secure aggregation to the HWFedAvg and NWFedAvg strategies presented in this paper. To this end, we use the below property of the Paillier cryptosystem.

$$\text{Dec}(ct^k \bmod n^2) = k \ \times pt \bmod n.$$

This property is used to allow the multiplication of ciphertext $ct_1 = \text{Enc}(pt_1)$ by a plaintext pt_2 and its output decrypts to the multiplication of both plaintexts,

$\mathrm{Dec}(ct_1 \times pt_2) = pt_1 \times pt_2$. With this, we can leave the aggregation weights W_i as plaintext while encrypting the local model weights. Here, each W_i is exposed to the central authority. Alternatively, the CKKS HE scheme can be used to both multiply and add ciphertext [20]. Due to memory constraints when using CKKS and fully encrypting the local models with Paillier HE, this work encrypts only the 2 outermost layers for the two HE scenarios tested. In the first scenario, the local models have four layers: input-561x437, inner1-437x312, inner2-312,6, and output-6. The second scenario uses a shallow model with three layers: input-561x432, inner-432x6, output-6. Both models are derived from tuners and have high performance, where a shallow model is ideal due to the costly time and space complexity brought by using HE.

Experiment Results and Discussions

In this section we outline and discuss the results from running the two experiments detailed in the previous section. First, the impact of a client holding homogeneous data is analyzed by using traditional and weighted aggregation strategies. Next, the experiment results when combining HE with FL are presented.

Experiment 1 Results and Discussion

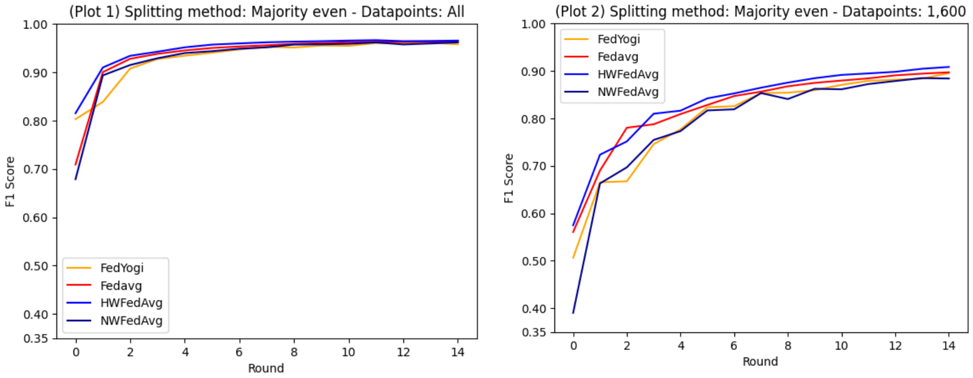
To evaluate the importance of data homogeneity, simulations will be run when using the HWFedAvg, NWFedAvg, FedAvg, and FedYogi aggregation strategies. While other strategies can also be experimented with, these results are preliminary and will be expanded upon with a more complex dataset. None of these strategies are optimized, and further optimizations can positively impact the performance of using a given strategy when applicable. These tests continue the tests from the previous PET Lab work on exploring FL aggregation strategies which indicate that FedAvg and FedYogi will outperform FedAdam and FedAdagrad on this task without proper tuning [17]. The outputs being analyzed are the F1 Scores for the general performance of the central model after each training round.

Since the dataset is insufficiently complex to distinguish performance differences over many rounds, we further test the same FL experiments with only a subset of the data being used. This mimics a scenario in which the distributed clients contain less data a given set of training rounds and helps to better compare the aggregation strategies, and hence better clarifies the importance of considering data homogeneity for distributed clients. Specifically, we first test with all 7,726 records, then test with only 1,600 records. Although different limitations on the dataset size have been tested, reduced or greater sizes did not highlight a difference as strong as when 1,600 is used.

The F1 Scores for both the full and limited dataset tests are presented below in plots 1 and 2 of Figure 3, comparing the four aggregation strategies.

F1 score comparison between the full dataset (7,724 records) and a limited dataset (1,600 records) over all strategies

Figure 3



When working with the full dataset, plot 1 of Figure 3 showcases that all strategies can quickly converge in learning the target classification task. Each achieves a very high F1 Score within only two training rounds when using their complete datasets to train the model. Consequently, reducing the number of samples available to each client for training in plot 2 of Figure 3 reduces the rate of learning and increases the time for each strategy to begin achieving higher F1 Scores. This showcases that the original classification task is too simple when using the full dataset and the differences between aggregation strategies becomes more prevalent in more complex scenarios.

There are several key observations which these plots present regarding when data homogeneity is considered. First, both tests highlight that HWFedAvg achieves higher performance in the first round compared to all other strategies whereas NWFedAvg performs the worst in that same round. While the end results slowly begin to converge, this is an important observation which indicates that valuing homogeneous data from clients containing a mix of homogeneous and heterogeneous data distributions can result in quicker overall learning of the target task. While more complex tasks will help further highlight this, this can be important when in a scenario where clients cannot frequently participate in training rounds or where few clients with homogeneous data are able to training rounds. Prioritizing clients with homogeneous data may result in a high-quality central model at a faster rate. Furthermore, if a context drift occurs, this faster learning may help efficiently adapt the model.

Second, we note that strategies which do not consider the heterogeneity problem (i.e. FedAvg) or that attribute greater weight to homogeneous clients (HWFedAvg) provide better results in terms of F1 Score compared to strategies considering the heterogeneity problem (i.e. FedYogi) or which attribute greater weight to heterogeneous datasets (NWFedAvg). This is highlighted by FedAvg and HWFedAvg generally performing better than FedYogi and NWFedAvg in plots 1 and 2 of Figure 3. These findings may not always hold but help exhibit the challenges of working with clients holding heterogeneous datasets in FL scenarios. Figure 3 also showcases that HWFedAvg most closely overlaps with FedAvg, whereas NWFedAvg closely overlaps with FedYogi.

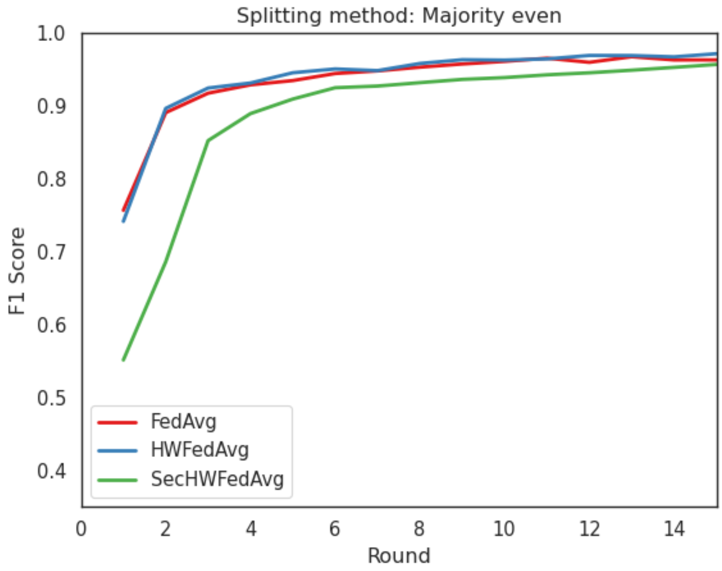
The results suggest that among the analyzed aggregation strategies, those favouring homogeneous dataset training tend to yield the best performance. This indicates that positively weighing clients with homogeneous datasets can result in better overall performance of the model in an FL environment.

Experiment 2 Results and Discussion

Within this experiment we have performed a suite of tests using HE on model architectures of varying complexity, where the full models are not encrypted due to memory constraints. These tests have highlighted that the secure versions of HWFedAvg and NWFedAvg perform well compared to the non-secure counterparts when only the output layer is encrypted. However, as Fig. 4 presents, performance degrades when encrypting more layers (in this scenario the 2 outermost layers are encrypted).

F1 score comparison between FedAvg, HWFedAvg, and the encrypted SecHWFedAvg (outermost two layers are encrypted with Paillier HE)

Figure 4



We hypothesize that this drop in performance is due to increased noise from the additional multiplication operations in the HE process. However, since the F1 Score of the secure aggregated model is approaching the non-secured versions, we intend to further explore this hypothesis in future work. Furthermore, we intend on expanding these tests to fully encrypt the models since both the larger and shallow models could not be fully encrypted due to RAM constraints at the time of testing. Finally, from the tests performed, we argue that in situations with a performance or memory bottleneck, a shallow model is best to use since it can be better secured with far less computational and memory cost than a larger model.

Conclusions and Next Steps

This paper has presented preliminary studies of the effects from applying custom weighting strategies to FedAvg in an FL environment. The simulations indicate that placing value on data homogeneity can result in better overall results whereas prioritizing heterogeneous data can reduce a global model's performance. These preliminary results, favouring a homogeneity-centric approach, warrant further exploration. Future research endeavours will involve more complex classification tasks and utilize benchmarking datasets to substantiate and extend these findings. We expect that in future experiments, the performance of more complex strategies, such as FedYogi, will improve. Furthermore, we hypothesize that a more statistically significant difference will be found when comparing HWFedAvg and NWFedAvg (with HWFedAvg performing better). Within the UN PET Lab, we will also be performing these scenarios in realistic deployments with mock data being used to test the infrastructure requirements needed to consider using FL.

To conclude, this paper has explored weighted FL aggregation strategies, named HWFedAvg and NWFedAvg, with the aim of better understanding the complexity of homogeneous and heterogeneous data scenarios in FL. Simulations have been run to explore the feasibility and utility of the weighting strategies. Paillier HE has also been applied to further test the effects of encrypting model weights within FL scenarios. A performance drop has been observed alongside the increased compute requirements. These tests indicate that data homogeneity is important to consider, but further work needs to be done on more complex tasks to derive more concrete results.

Disclaimer

The content of this article represents the position of the authors and may not necessarily represent that of Statistics Canada, ISTAT, and Statistics Netherlands.

References

[1] United Nations. (2023). *United Nations Guide on Privacy-Enhancing Technologies for Official Statistics*. United Nations Committee of Experts on Big Data and Data Science for Official Statistics, New York. Website: https://unstats.un.org/bigdata

[2] Liu, J., Huang, J., Zhou, Y., Li, X., Ji, S., Xiong, H., & Dou, D. (2022). From distributed machine learning to federated learning: A survey. *Knowledge and Information Systems*, *64*(4), 885-917.

[3] Konečný, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., & Bacon, D. (2016). Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*.

[4] Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, *37*(3), 50-60.

[5] McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017, April). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics* (pp. 1273-1282). PMLR.

[6] Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., ... & McMahan, H. B. (2020). Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*.

[7] Ju, L., Zhang, T., Toor, S., & Hellander, A. (2023). Accelerating Fair Federated Learning: Adaptive Federated Adam. *arXiv preprint arXiv:2301.09357*.

[8] Gosselin, R., Vieu, L., Loukil, F., & Benoit, A. (2022). Privacy and security in federated learning: A survey. *Applied Sciences*, *12*(19), 9901.

[9] Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., ... & Zhao, S. (2021). Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, *14*(1–2), 1-210.

[10] Gentry, C. (2009). *A fully homomorphic encryption scheme*. Stanford university.

[11] Morris, L. (2013). Analysis of partially and fully homomorphic encryption. *Rochester Institute of Technology*, *10*, 1-5.

[12] Gentry, C., Sahai, A., & Waters, B. (2013). Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Advances in Cryptology–CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I* (pp. 75-92). Springer Berlin Heidelberg.

[13] Gentry, C. (2009, May). Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing* (pp. 169-178).

[14] Yang, Y., Huang, X., Liu, X., Cheng, H., Weng, J., Luo, X., & Chang, V. (2019). A comprehensive survey on secure outsourced computation and its applications. *IEEE Access*, *7*, 159426-159465.

[15] Acar, A., Aksu, H., Uluagac, A. S., & Conti, M. (2018). A survey on homomorphic encryption schemes: Theory and implementation. *ACM Computing Surveys (Csur)*, *51*(4), 1-35.

[16] Nguyen, H., Phan, L., Warrier, H., & Gupta, Y. (2022). Federated Learning for Non-IID Data via Client Variance Reduction and Adaptive Server Update. *arXiv preprint arXiv:2207.08391*.

[17] Santos, B., Templeton, J., Chemli, R., Molladavoudi, S., Pugliese, F., Cerasti, E., ... & Jug, M. (2023, September). Insights into privacy-preserving federated machine learning from the perspective of a national statistical office. In *Conference of European Statistics*.

[18] Beutel, D. J., Topal, T., Mathur, A., Qiu, X., Fernandez-Marques, J., Gao, Y., ... & Lane, N. D. (2020). Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390*.

[19] Paillier, P. (1999, April). Public-key cryptosystems based on composite degree residuosity classes. In *International conference on the theory and applications of cryptographic techniques* (pp. 223-238). Berlin, Heidelberg: Springer Berlin Heidelberg.

[20] Cheon, J. H., Kim, A., Kim, M., & Song, Y. (2017). Homomorphic encryption for arithmetic of approximate numbers. In *Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I 23* (pp. 409-437). Springer International Publishing.

3rd IFC Workshop - Federated Learning Data Science in Central Banking

Aggregation strategies for Federated Learning

Erika Cerasti

- Benjamin Santos, Julian Templeton, Rafik Chemli, Saeid Molladavoudi (Statistics Canada)
- **Erika Cerasti**, Francesco Pugliese, Massimo De Cubellis (ISTAT)
- Matjaz Jug (Statistics Netherlands)

17-19 October 2023
Bank of Italy - Rome

International Context

PETs
***privacy-enhancing technologies*:**
Methodologies and approaches
to mitigate privacy risks when using
sensitive or confidential data

NSOs can build greater trust with the public and unlock new opportunities associated with more accurate and complete data collection.

https://unstats.un.org/bigdata/task-teams/privacy/index.cshtml

Are PETs highly reliable and can they be used for accessing sensitive data (health records, tax records or credit card data) by NSOs?

International Context

UNCEBD : UN Committee of Experts on Big Data and Data Science for Official Statistics

Elements to accelerate the adoption of PETs in the NSO community:

- Experimentation (PET Lab)
- Outreach & Training
- Support Services

UN PET Lab

Created to facilitate experimentation on pilot projects and effective collaboration on "real world" use case.

<u>Objectives</u>: Develop principles, policies and open standards for data sharing, taking full account of data privacy, confidentiality and security issues.

Secure Multi-Party Computation

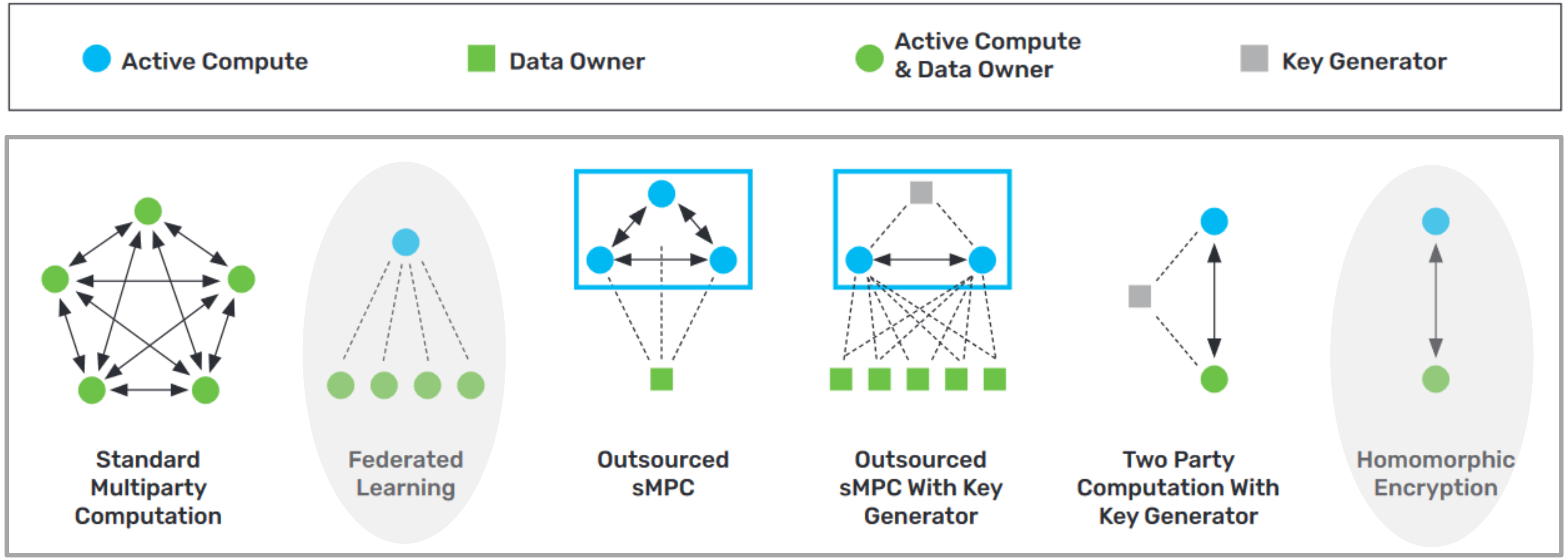
Secure multi-party computation – sMPC

Two or more (mutually distrusting) parties wish to compute an agreed-on function on data that they provide to that computation but are unwilling to disclose to others.

Federated Learning

sMPC protocols use frequent communication among the compute parties.

- available network bandwidth
- network latency between parties



Active Compute
Data Owner
Active Compute & Data Owner
Key Generator
Standard Multiparty Computation
Federated Learning
Outsourced sMPC
Outsourced sMPC With Key Generator
Two Party Computation With Key Generator
Homomorphic Encryption

Federated Learning

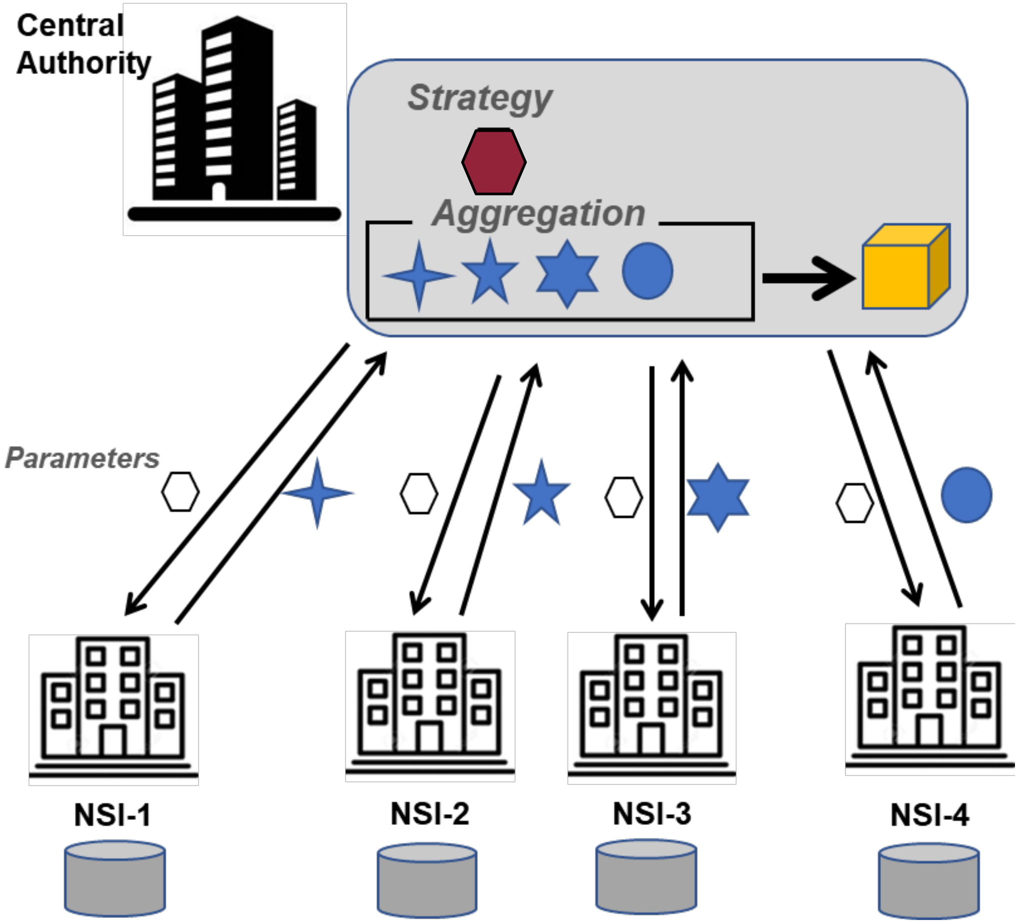
Decentralized / Distributed Computation

FL allows a centralized ML model to be trained on data residing on distributed client devices.
After training the model with data locally, a client will send the weights or gradients back to the server to be aggregated.

This allows analytics to be derived without collecting data.

Aggregation strategy

Performance of the trained models can reach similar performance of centralized approaches but a careful selection of the hyperparameters and the aggregation strategy is important.



Federated Learning and Homomorphic Encryption

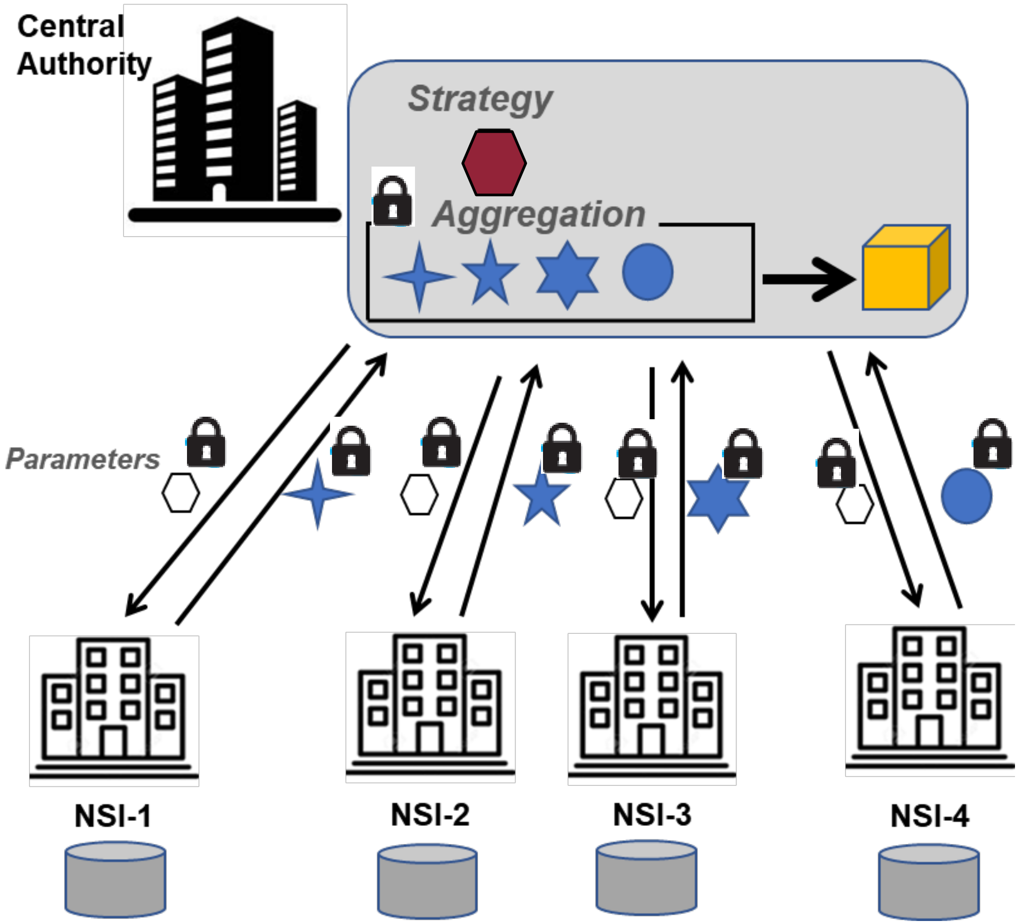
Decentralized / Distributed Computation

Locally trained client models can still be attacked, which can be better protected by using other PETs in conjunction with FL

- Example: Homomorphic Encryption (HE)

HE is a cryptographic technology allowing for the direct computation (addition and multiplication) on encrypted data.

It adds more computational complexity and can result in a high computational overhead.



Aggregation strategies

Human Activity Recognition public DATASET

30 volunteers (19-48 years) - 6 class of human activities

Recordings:

Accelerometer and gyroscope data collected by smartphones.
(3-axial linear acceleration and 3-axial angular velocity, rate of 50Hz)

Human activity classes

D. Anguita et al ,ESANN 2013

Work objectives

- Explore different federated aggregation strategies
- Explore the role of dataset **heterogeneity**
- Apply HE to model weights

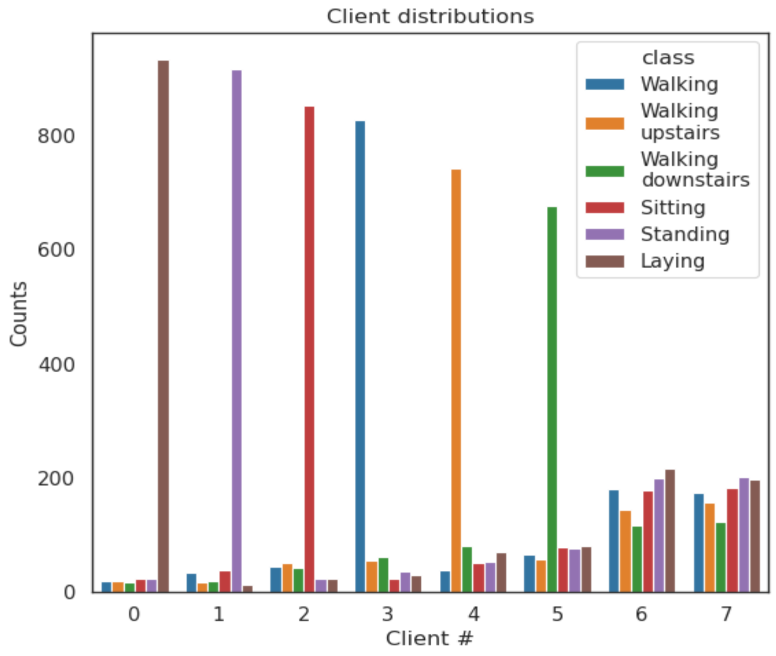
Federated aggregation Strategies

- FedAvg
- **Weighted FedAvg (WFedAvg)**
- FedAdagrad
- FedAdam
- FedYogi

Data Splitting Methodologies

Splitting methods - heterogeneous datasets – 8 clients

- ***Random***: Samples are randomly distributed among clients
- ***Majority even:*** Each client has one majority class, same number of records
- ***Majority:*** Each client has one majority class, different number of records
- ***Pick two***: Each client has two majority classes, same number of records



Note that each class can only be assigned as a majority class once, where remaining clients without a majority class are given a distribution of all classes.

Weighted Federated Averaging Strategy

- Index i runs on clients, i.e, 1 to 8 clients
- Index j runs on classes, i.e, 1 to 6 classes
- Properties on i are related to clients: weights w_i, Gini coefficient G_i, entropy H_i
- Properties on j are related to classes: probabilities for picking particular class j for client i.

A vector of probabilities p_i is local for a client i: $\vec{p_i} = p^i_j \rightarrow \{p^i_1, \ldots, p^i_6\}$ with $p^i_j = \frac{\text{count class } j \text{ client } i}{N_i}$ with N_i the total number of samples for client i. p^i_j is normalized.

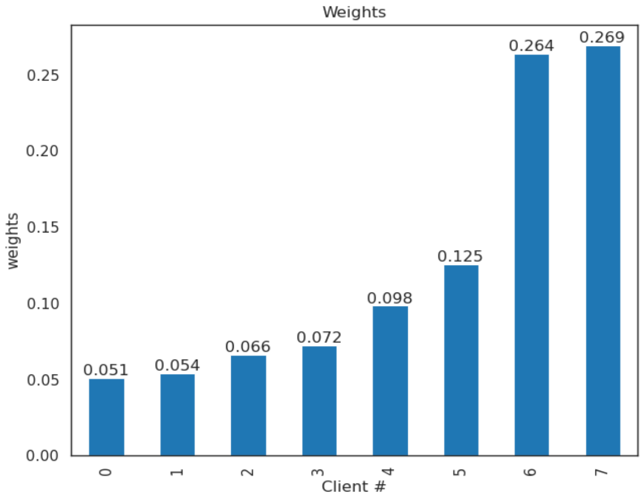
The vector P_j is global (common to all clients) $\vec{P} = P_j \rightarrow \{P_1, \ldots, P_6\}$ with $P_j = \frac{\text{count class j for all clients}}{N}$ with $N = \sum_i N_i$ the total number of samples for all clients. P_j is normalized.

A set of weights could be constructed by multiplying a vector of a local property times a set of local vectors for each client. For instance, let $\tilde{P}$ be the normalized product of the probabilities of picking a class j in the joint dataset times the square probability for a client i to have this class j.

$$\tilde{P}_i = \frac{\sum_j P_j \cdot (p^j_i)^2}{\sum_i \sum_j P_j \cdot (p^j_i)^2}$$

Setting this squared probabilities adds more importance to majority classes respect to the minority classes. Then the normalized weights can be computed as the inverse of this value:

$$w_i = \frac{1/\tilde{P}_i}{\sum_i 1/\tilde{P}_i}$$



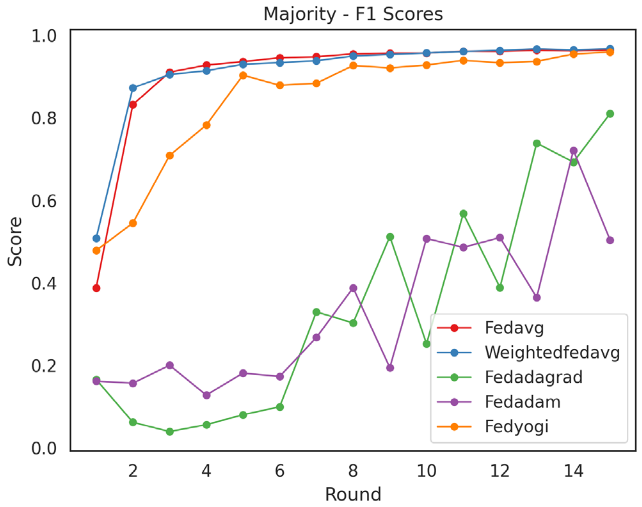
Experiment Results

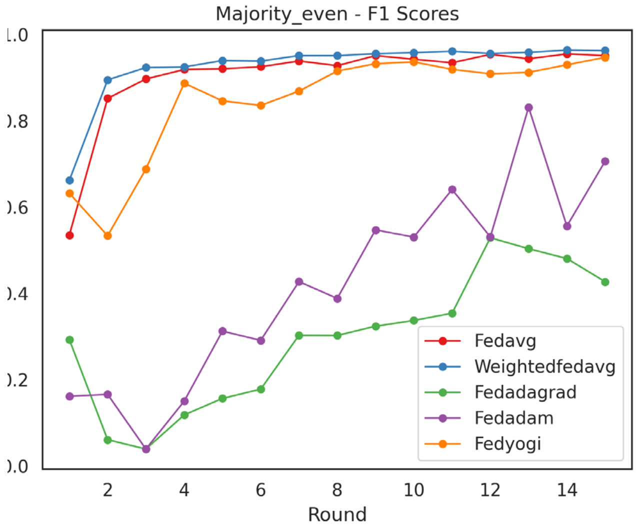
- FedAvg, WFedAvg and FedYogi perform better than Fedadagrad and Fedadam
  - Fedadam and Fedadagrad would do better if properly optimized

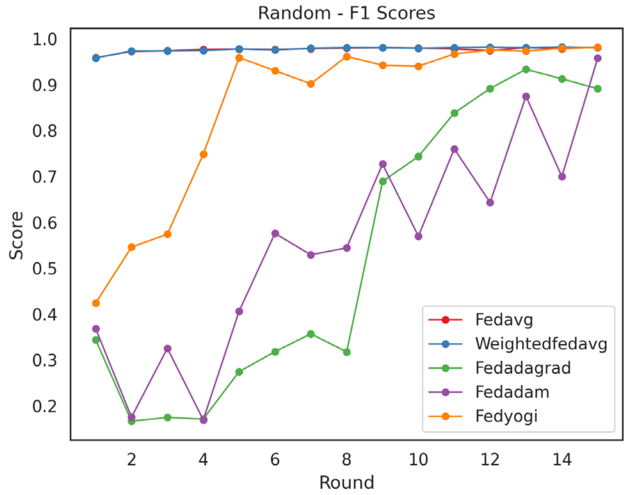
- FedYogi seem to reach convergence a bit later than Fedavg and WFedAvg

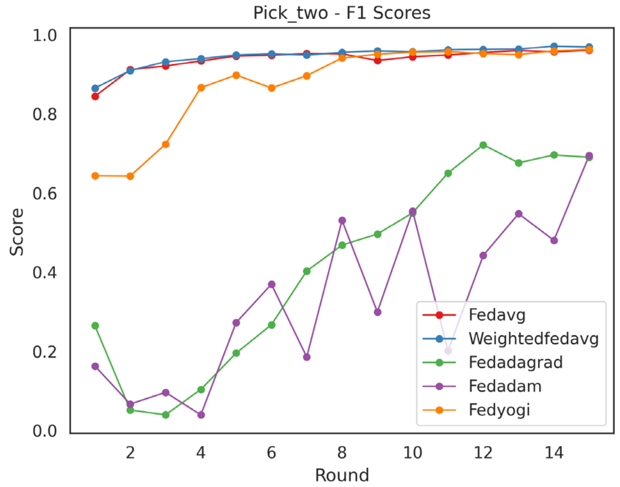
- FedAvg and WFedAvg show a very similar behavior (for random split they are expected to be identical)

- The convergence is reached after few rounds

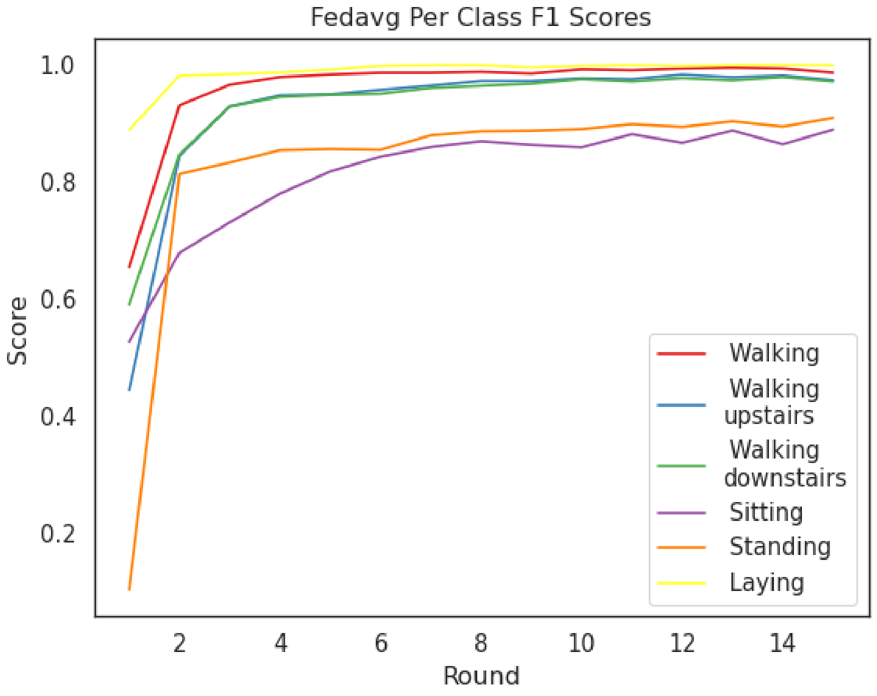


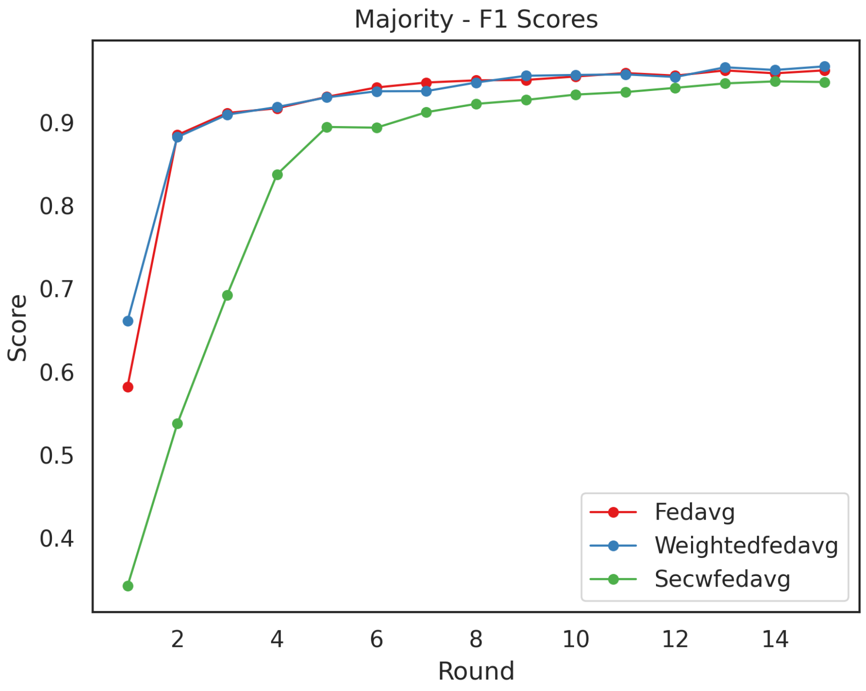






Experiment Results





- FedAvg Performance for each activity class
- For two classes learning is slower

- Performance of FedAvg and WFedAvg compared to FedAvg with Homomorphic Encritption
- HE makes the convergence slower

Conclusions

- Homomorphic Encryption can add significant time and communication costs, scaling with the amount of encrypted weights/gradients
- Heterogeneity of the training dataset seems to not affect performance in this example:
  - The best models are FedAvg and WFedAvg, which are very similar
  - Adaptive methods (FedYogi) do not show a better performance with heterogenous data
  - It seems that the simpler aggregation technique (FedAvg) work better for this dataset

These results are preliminary

- The task is too easy with the selected dataset. Convergence is reached too quickly
- Next steps are to explore different hyperparameters and use a more complex dataset

Preliminary results indicate faster overall training and faster learning of different classes when using Weighed Federated Averaging (can lead to less communication cost if learning is quicker)

UN PET repository: **https://unstats.un.org/wiki/display/UGTTOPPT/Case+study+repository**

Thanks for your attention

For further information please write to *erika.cerasti@istat.it*