Federated Learning: A Comparison of Methods

How do different ML models compare to each other

Emīls Sīpols

Supervisor(s): Marcel Reinders, Swier Garst

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 25, 2023

Name of the student: Emīls Sīpols
Final project course: CSE3000 Research Project
Thesis committee: Marcel Reinders, Swier Garst, Lydia Chen

Abstract

Federated learning (FL) has emerged as a promising approach for training machine learning models using geographically distributed data. This paper presents a comprehensive comparative study of various machine learning models in the context of FL. The aim is to evaluate the efficacy of these models in different data distribution scenarios and provide practical insights for practitioners in the field. The findings highlight the performance and limitations of linear and non-linear models on MNIST and Kinase datasets.

1 Introduction

Federated learning (FL)[1] has emerged as a novel approach in machine learning that has garnered considerable attention in recent years. FL aims to facilitate model training using data that is geographically distributed, without having to collect all the data in a single centralized location. The strategy involves allowing nodes or devices to train their own models with their respective data, and then using an aggregation algorithm to combine the learned weights from each node to create a new, enhanced model. This is an iterative process that consists of multiple rounds of local training and model weight aggregation, enabling continual improvement of the model's performance over time.

FL provides numerous advantages, particularly in situations where data is scattered across multiple locations[2]. In the healthcare sector, for instance, centralizing medical data is challenging due to privacy concerns. The sensitive nature of medical data necessitates strict privacy regulations and safeguards, making it difficult to gather all the data in one central location. In such cases, FL enables the development of a robust and precise model while ensuring data confidentiality and privacy. Additionally, FL is beneficial in resource-intensive data collection scenarios since it allows the utilization of existing data sets to create new models. However, despite the numerous advantages of federated learning, it is important to acknowledge that there are challenges to overcome.

One of the primary challenges in federated learning is handling data heterogeneity and ensuring the scalability of the models. The distribution of data across different devices can be uneven, leading to variations in data characteristics and statistical properties. Models trained on a few devices may become biased towards those devices, resulting in limited generalizability and inaccurate predictions when applied to data from other sources[3]. Additionally, the scalability of federated learning systems poses a significant challenge, as the number of participating clients and their data can be vast. Overcoming these challenges is crucial for effectively leveraging the potential of federated learning in privacy-preserving and distributed machine learning applications.

When dealing with federated learning and heterogeneous data distributions, exploring different ML models becomes particularly interesting due to the varying ways in which models handle and capture the complexities of the data. Different ML models possess unique mathematical properties and assumptions that can impact their performance on heterogeneous data. Linear models, for example, assume linearity and may struggle to capture nonlinear relationships present in certain data distributions. Nonlinear models, on the other hand, offer more flexibility and can capture complex patterns more effectively.

Therefore, the objective of this research project is to investigate and evaluate the efficacy of various machine learning models in distinct scenarios within the context of federated learning. The study contributes to the expanding body of research on federated learning by presenting implementation examples of the considered scenarios. This research aims to enhance our understanding of the strengths and limitations of different models and provide valuable recommendations for practitioners in the field of federated learning.

2 Background

This section presents the relevant knowledge used to compare the performance of different machine learning models in both centralized and federated learning settings. It consists of three main components: data collection and preprocessing, model selection, and performance evaluation.

2.1 Data Collection and Preprocessing

In order to explore various scenarios in federated learning, this study goes beyond independent and identically distributed (IID) data and incorporates non-IID data distributions. While the IID scenario assumes that each client possesses a representative subset of the complete dataset with similar data distributions, introducing non-IID data distributions allows for the evaluation of machine learning models in more challenging scenarios[4].

By considering both IID and non-IID data distributions, along with different levels of intensity in class separations between clients, the study aims to provide a comprehensive assessment of machine learning models in federated learning scenarios. This approach encompasses a wide range of data distribution characteristics and challenges, contributing to a deeper understanding of the models' robustness and performance in real-world settings.

2.2 Model Selection

A diverse set of machine learning models has been carefully chosen for this research to encompass both linear and non-linear approaches. The selected models include the Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Logistic Regression (LogReg), and Support Vector Machine (SVM).

The Multilayer Perceptron (MLP), CNN, and RNN are particularly well-suited for handling complex patterns and dependencies in data[5]. MLP is a feedforward neural network that can effectively capture non-linear relationships between features. CNNs excel at extracting spatial and hierarchical patterns from image data through the use of convolutional layers and pooling operations. RNNs, on the other hand, are designed to capture sequential dependencies in data, making them suitable for processing time series or text data.

In contrast, Logistic Regression and SVM provide efficient solutions for linearly separable problems. Logistic Regression is a linear model that employs a sigmoid function to estimate probabilities and make binary classifications. SVM, on the other hand, finds an optimal hyperplane that maximally separates classes in the feature space.

These selected models have a proven track record and have been widely used in various machine learning applications. They are also relevant to the field of federated learning, where different data distributions and challenges arise due to the distributed nature of the data.

By including both linear and non-linear models, this research aims to explore the strengths and weaknesses of each model in the context of federated learning. The evaluation and comparison of these models will provide valuable insights into their performance under different scenarios, facilitating the identification of suitable models for specific data distribution settings in federated learning.

2.3 Performance Evaluation

To evaluate the performance of the machine learning models, experiments will be conducted in both centralized and federated learning settings. In the centralized setting, the models will be trained on the complete dataset, while in the federated learning setting, the models will be trained on distributed data across multiple clients, with each client holding a fraction of the dataset. To evaluate the accuracy of the models, the primary metric of interest will be the accuracy over communication rounds.

3 Methodology and Experimental Setup

This section provides a detailed account of the approach that was employed to conduct the study. It encompasses various implementation aspects, including the techniques, methodologies, and tools utilized in the project. Additionally, it covers the selection and preparation of datasets, along with any preprocessing or cleaning steps performed on the data.

3.1 Frameworks

For the implementation of the linear models, the scikit-learn[6] library was utilized. Scikit-learn provides various linear models such as Logistic Regression and Support Vector Machine, along with functionalities for data preprocessing, feature selection, and evaluation metrics.

To handle the implementation of non-linear models, including Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN), the TensorFlow framework was leveraged. TensorFlow[7] is an open-source library for deep learning that supports the building, training, and evaluation of neural network models.

Additionally, the Flower framework[8] was employed to enable communication between the server and client nodes in the federated learning setting. Flower simplifies the development and coordination of federated learning systems, providing high-level abstractions and tools for client-server communication, model aggregation, and coordination among participating clients.

3.2 Datasets

The selection of appropriate datasets is essential in evaluating the performance of machine learning models. This subsection provides a description of the datasets used in the experiments and highlights their key characteristics.

MNIST Dataset

The MNIST dataset[9], widely used for image classification tasks, serves as the first dataset in our study. It consists of a large collection of handwritten digit images (0-9) and corresponding labels. Each image in the dataset is grayscale and has a dimension of 28x28 pixels. The MNIST dataset provides a benchmark for evaluating the models' performance in image recognition and classification tasks.

Kinase Dataset

The Kinase dataset used in this study is based on "molecular fingerprints" and focuses on capturing the structural elements of molecules numerically[10]. The data points are small molecules that are described by 8191 integer features, and the goal is to predict whether a molecule inhibits a specific type of protein known as FLT3, a kinase. The label associated with each molecule is binary, with values of either 0 or 1. In the case of the Kinase dataset, its complexity provides a more challenging task compared to the MNIST dataset. This increased difficulty level allows us to benchmark and compare the performance of different machine learning models in handling complex data relationships and addressing intricate prediction problems.

3.3 Federated Setting

This subsection presents the experimental configuration and setup for the federated learning setting. It outlines the experimental setup, the data distribution among clients and the specific ML model infrastructures.

Experimental Setup

For the MNIST dataset, the experiments involved 10 clients, while for the kinase dataset, which was initially collected as three separate datasets, three clients were used. In each communication round, every client performs one local epoch of training. The performance of the federated experiments will be compared against a central implementation for evaluation and analysis.

Data Distribution

In the federated learning setting, different data distribution scenarios were considered to evaluate the robustness and generalization capabilities of the machine learning models. These scenarios encompass both IID data, as well as non-IID data with varying levels of label distribution among clients.

In the IID scenario, it is assumed that each client has a representative subset of the complete dataset, and the data distributions among all clients are similar.

To ensure the IID attribute while distributing the data samples across clients, a random assignment is performed. This assignment ensures that there is no overlap between the data samples assigned to each client, preserving the IID characteristic of the data. Figure 1 illustrates an example of this data distribution.

Figure 1: Example of MNIST IID label data distribution for all clients.

To assess the models' performance in more challenging and realistic scenarios, non-IID data distributions were introduced. These distributions include imbalanced distributions where certain clients have a significantly higher proportion of samples from specific classes.

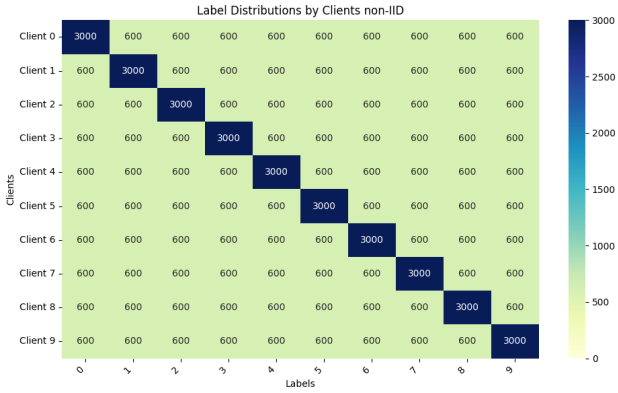

Figure 2: Example of MNIST non-IID softer skew label data distribution for all clients.

For the MNIST dataset non-IID cases, an imbalanced class distribution scenario is introduced. This deliberate skewing of the data distribution creates class imbalances across the clients. The first scenario can be seen in Figure 2. This case will be referred to as the non-IID case throughout the paper. The harsher scenario, that will be further referred to as aggressive non-IID, can be seen in Figure 3. The purpose of these scenarios is to assess the ability of federated learning models to handle the challenges posed by class imbalances commonly encountered in real-world datasets.

Models

For the MNIST dataset, LogReg makes use of the `LogisticRegression` class from the scikit-learn library, applying the l2 penalty. On the other hand, for the Kinases dataset, the logistic regression model utilizes the `SGDClassifier` class with a learning rate of 0.0001 and "log_loss" as the chosen loss function. The SVM implementation also relies on the `SGDClassifier` class, with the loss parameter set as "hinge" and a learning rate of 0.00001 for both datasets. For details about non-linear models, see Table 1.

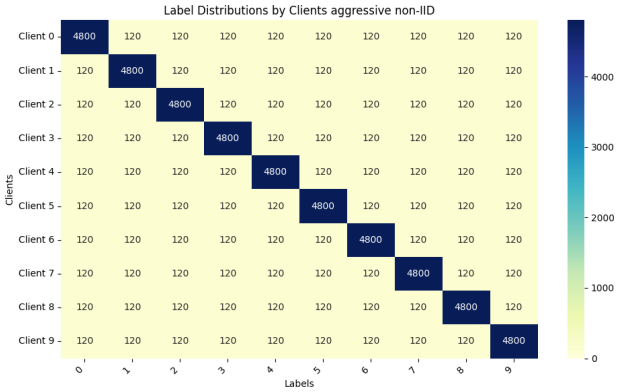

Figure 3: Example of MNIST non-IID harsher skewed label data distribution for all clients.

Dataset: MNIST	**Dataset: Kinases**
Multi-Layer Perceptron (MLP)	**Multi-Layer Perceptron (MLP)**
Learning rate: 0.0001 Input: $(28, 28)$ Fully Connected: 784×128 (ReLU) Fully Connected: 128×256 (ReLU) Fully Connected: 256×10 (Softmax)	Learning rate: 0.0001 Input: $(8192,)$ Fully Connected: 8192×64 (ReLU) Dropout: 0.7 Fully Connected: 64×32 (ReLU) Dropout: 0.7 Fully Connected: 32×2 (Softmax)
Convolutional Neural Network (CNN)	**Convolutional Neural Network (CNN)**
Learning rate: 0.0001 Input: $(28, 28)$ Conv2D: 32 filters, kernel size $(3, 3)$ MaxPooling2D: pool size $(2, 2)$ Conv2D: 64 filters, kernel size $(3, 3)$ Fully Connected: 784×128 (ReLU) Fully Connected: 128×10 (Softmax)	Learning rate: 0.0001 Input: $(8192,)$ Conv1D: 32 filters, kernel size $(3, 3)$ MaxPooling1D: pool size 2 Conv1D: 64 filters, kernel size $(3, 3)$ MaxPooling1D: pool size 2 Fully Connected: 8192×64 (ReLU) Dropout: 0.7 Fully Connected: 64×32 (ReLU) Dropout: 0.7 Fully Connected: 32×2 (Softmax)
Recurrent Neural Network (RNN)	**Recurrent Neural Network (RNN)**
Learning rate: 0.00001 Input: $(28, 28)$ SimpleRNN: 784×128 Fully Connected: 128×10 (Softmax)	Learning rate: 0.00001 Input: $(8192,)$ SimpleRNN: 8192×128 Fully Connected: 128×2 (Sigmoid)

Table 1: Architecture of Models

4 Results

4.1 Logistic Regression

MNIST

The data obtained from the experiments is shown in Figure 4. Comparing the central implementation and the IID case, revealed a difference in accuracy between the two approaches. The centralized implementation achieved higher accuracy compared to the IID scenario, with a small margin of difference observed towards the later rounds of communication. This indicates that the decentralization introduced in the IID setting has a slight impact on the overall accuracy achieved, albeit not as large as with the non-IID cases.

The non-IID scenarios demonstrate a noticeable performance impact compared to the centralized approach, as can

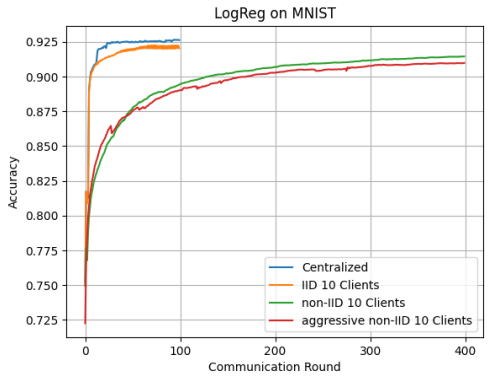

Figure 4: Accuracy over communication rounds for Logistic Regression on MNIST data.

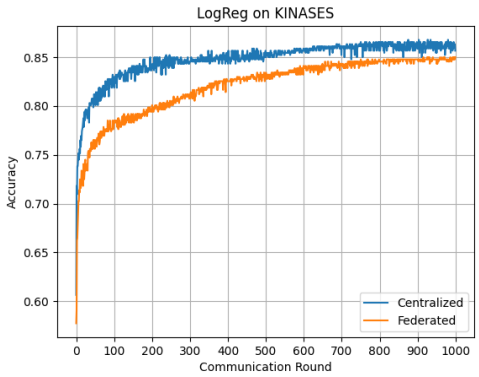

Figure 5: Accuracy over communication rounds for Logistic Regression on Kinases data.

be seen in Figure 4. It required more than four times the number of communication rounds needed for the centralized case to converge. Additionally, there is an accuracy difference of approximately 0.02 between the non-IID and central scenarios, indicating that the non-IID data distribution has indeed affected the model's performance.

Although the non-IID models show a longer convergence time and a noticeable difference in accuracy compared to the centralized scenario, they still exhibit good final accuracy overall. It is important to note that there is a distinction between the non-IID and aggressive non-IID cases, indicating that the level of label skewness affects the model's performance. While the logistic regression model can effectively adapt and learn from the skewed data distribution, it is evident that there is still a difference between the performance of the central and federated approaches.

Kinase

Both the centralized and federated cases achieved results can be seen in Figure 5. The centralized approach exhibited slightly higher accuracy compared to the federated scenario, although the difference between the two is minimal. This suggests that the model is capable of handling the federated scenario reasonably well.

An interesting observation from the accuracy graph is that the accuracy values fluctuate over the communication rounds, creating a "jumpy" pattern. Despite these fluctuations, the general trend shows an increase in accuracy over time. This behavior could be due to the complexity of the Kinase dataset.

4.2 Support Vector Machine

MNIST

In the non-IID cases, Figure 6, the accuracy values exhibit fluctuations over the communication rounds, with no noticeable improvement in the average accuracy over time. On the other hand, the IID case, although starting with a lower accuracy compared to the non-IID cases, shows a converging trend towards higher accuracy as the communication rounds progress. Eventually, the IID case surpasses both non-IID cases in terms of accuracy.

Among the non-IID cases, it is worth noting that the more skewed distribution demonstrates a noticeable difference in performance compared to the softer skew. The softer skew scenario performs relatively better, implying that a less skewed data distribution among clients contributes to improved SVM performance.

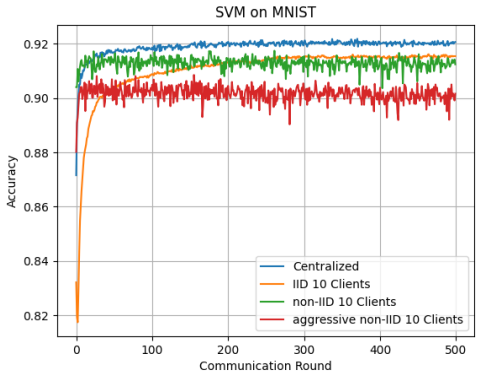

Figure 6: Accuracy over communication rounds for SVM on MNIST data.

Kinase

The SVM model achieves performance that is similar to logistic regression on the Kinase dataset, Figure 7. This observation is expected, since both SVM and logistic regression are linear models that aim to separate data points based on a linear decision boundary.

The similarity in performance between the central and federated scenarios using SVM and logistic regression on the Kinase dataset suggests that the linear relationship captured by these models is sufficient for making accurate predictions in this context. However, it should be noted that there is still a difference between the accuracy achieved in the central and federated cases, indicating the impact of the decentralization process on the overall performance. Nonetheless, both SVM and logistic regression models demonstrate their effectiveness in handling the complexity of the dataset.

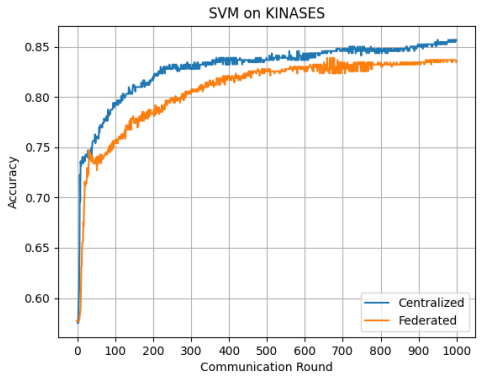

Figure 7: Accuracy over communication rounds for SVM on Kinases data.

4.3 MLP

MNIST

The data obtained, seen in Figure 8, demonstrates a notable difference is the higher accuracy achieved by the MLP model compared to the linear models. This higher accuracy can be attributed to the MLP model's ability to learn and model complex non-linear relationships within the dataset.

Furthermore, the MLP model achieves this higher accuracy with fewer communication rounds compared to the linear models. This indicates the MLP model's efficiency in leveraging its deeper architecture and multiple layers of neurons to capture intricate patterns and improve classification accuracy more rapidly. Additionally, the MLP model also exhibits the ability to generalize better in the non-IID case.

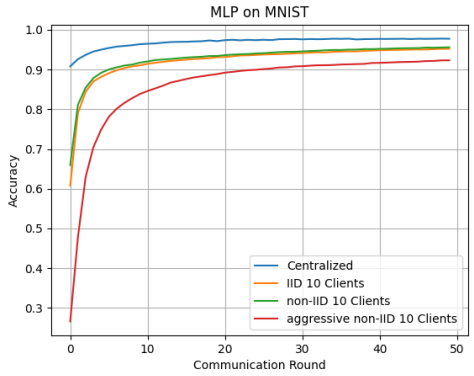

Figure 8: Accuracy over communication rounds for MLP on MNIST data.

Kinase

The MLP model converges in a similar manner to the logistic regression and SVM models on the Kinase dataset, Figure 9, however, there are a few notable differences. Firstly, the MLP model achieves convergence in fewer communication rounds compared to the other models. Additionally, the accuracy of the MLP model exhibits fewer fluctuations compared to the other models during the communication rounds. This suggests that the MLP model is able to achieve more stable and consistent predictions throughout the training process.

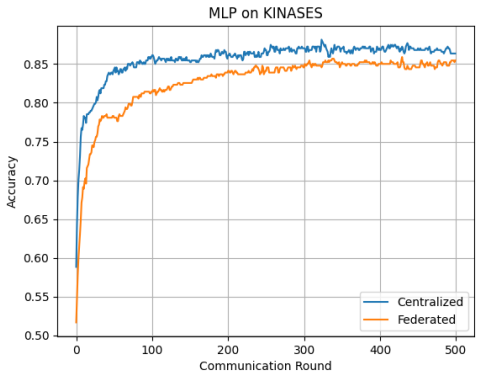

Figure 9: Accuracy over communication rounds for MLP on Kinases data.

4.4 CNN

MNIST

The results obtained from the CNN models, Figure 10, on the MNIST dataset exhibit similar patterns to those observed with the MLP model. However, there is one notable difference: the CNN models achieve higher accuracy across all data distribution scenarios compared to the MLP model. This suggests that the CNN models benefit from the spatial relationships present in the image data, enabling them to capture more intricate patterns and generalize better.

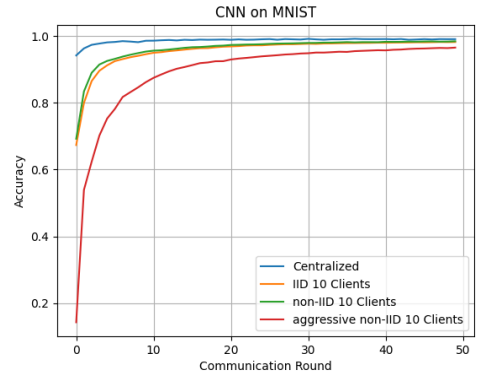

Figure 10: Accuracy over communication rounds for CNN on MNIST data.

Kinase

Performance of the CNN model on the Kinase dataset, Figure 11, exhibits similar behavior to the MLP model. However, there are some notable differences. The fluctuations in accuracy observed in the CNN model have a larger amplitude compared to the MLP model, indicating a higher degree of variability in performance over the communication rounds.

Like the previous models, the CNN model shows that the centralized case consistently outperforms the federated scenario in terms of accuracy. This suggests that the centralization of data and model parameters contributes to better performance in this particular dataset.

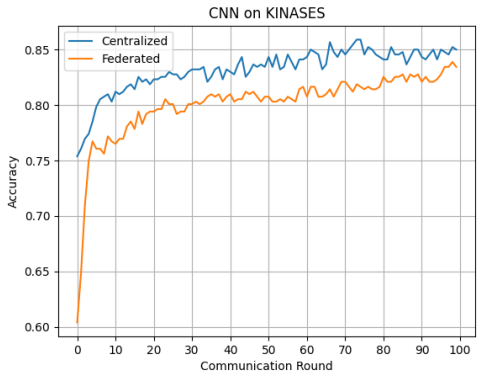

Figure 11: Accuracy over communication rounds for CNN on Kinases data.

4.5 RNN

MNIST

The RNN models on the MNIST dataset exhibit similarities with the MLP and CNN models in terms of characteristics and final achieved accuracy, Figure 12. Both models showcase the ability to learn and make accurate predictions on the handwritten digit images. However, there are notable differences observed between the two models.

One significant difference is the convergence speed. The other non-linear models tend to converge faster compared to the RNN models. This discrepancy can be attributed to the inherent architecture differences between them. While MLP and CNN models process images as vectors, RNN models handle sequential data, treating each row of pixels as a time step. The sequential nature of the RNN models introduces additional complexity, resulting in a slower convergence rate.

Another difference lies in the performance of the non-IID data distribution scenario. The non-IID case in the RNN models exhibits a larger performance gap compared to the IID case, whereas the MLP models showed a smaller difference.

Kinase

The data of the RNN models on the Kinase dataset reveal some distinct characteristics compared to other models, Figure 13. The RNN model achieved the lowest accuracy among all the models evaluated on this dataset. Furthermore, the accuracy exhibited noticeable fluctuations over the communication rounds, and the model did not seem to converge to a stable performance.

The challenges encountered by the RNN model on the Kinase dataset can be attributed to the dataset's unique characteristics. The Kinase dataset comprises a larger number of features compared to the MNIST dataset. The presence of a high-dimensional feature space can pose challenges for RNN

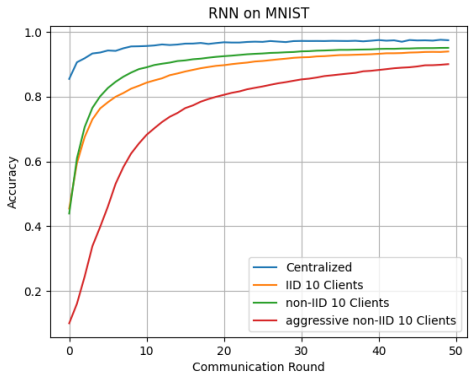

Figure 12: Accuracy over communication rounds for RNN on MNIST data.

models, particularly in terms of the exploding or vanishing gradient problem.

The exploding or vanishing gradient problem arises when gradients during the backpropagation process either become too large or too small, hindering the model's ability to learn effectively. This problem can be exacerbated in datasets with a large number of features, potentially leading to unstable training and difficulty in finding an optimal solution.

It is interesting to note that despite both the central and federated cases possibly being affected by the gradient problem, there still exists a difference between their performances. The RNN model exhibits lower accuracy compared to the centralized case. This discrepancy suggests that factors beyond the gradient problem may contribute to the observed differences in performance between the two scenarios.

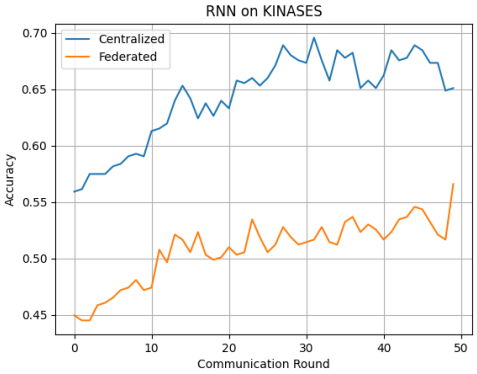

Figure 13: Accuracy over communication rounds for RNN on Kinases data.

5 Responsible Research

Efforts have been made to ensure the reproducibility of the methods employed in this study. The code implementations of various machine learning models, such as MLP, CNN, RNN, Logistic Regression, and Support Vector Machine,

have been made publicly available[1]. These implementations, along with the provided datasets, allow others to access and replicate the experiments conducted in this study. It is crucial to acknowledge that reproducibility is an ongoing process. As new techniques and frameworks emerge, modifications may be necessary to replicate the results.

Additionally, detailed descriptions of the experimental setup, including the number of clients, data distributions, and training configurations, have been provided to facilitate the replication of the research. These details aim to encourage future researchers to validate the findings, explore alternative approaches, and contribute to the advancement of federated learning research.

6 Discussions and Conclusions

This study compared the performance of different machine learning models in the context of federated learning using two datasets: MNIST for image classification and Kinase for predicting molecular inhibition of FLT3 kinase. The findings provide valuable insights into the strengths and limitations of these models in the federated learning setting.

Linear models, such as Logistic Regression and Support Vector Machines, performed well on both datasets, achieving high accuracy and steady convergence, particularly in the case of IID data. However, they faced challenges in handling non-IID data distributions and class imbalances, resulting in decreased performance and slower convergence in terms of communication rounds.

Non-linear models, including Multi-Layer Perceptron, Convolutional Neural Networks, and Recurrent Neural Networks, demonstrated superior performance on the MNIST dataset compared to linear models. These models effectively captured complex patterns in the image data, leading to higher accuracy. Additionally, MLP and CNN showed promise in handling non-IID data distributions, achieving comparable or better accuracy than the IID scenario. However, RNN struggled with the Kinase dataset, highlighting the challenges of high-dimensional molecular data and sequential modeling.

One limitation of this study is the restricted time allocated, which resulted in a limited exploration of machine learning models and datasets within the federated learning setting. A more extensive incorporation of additional models and datasets would have provided a more comprehensive understanding of their performance. Furthermore, due to computational resource constraints, the study was unable to conduct experiments on separate machines. This limitation hindered the exploration of various scenarios with different numbers of clients, varying communication rounds, and diverse computational power across clients.

An additional important consideration is that the observed performance of the non-IID cases for SVM model on the MNIST dataset, Figure 6, particularly the fluctuation of accuracy, may be influenced by the specific implementation details of the `SGDClassifier` from scikit-learn that was used for SVM implementation in this study. This observation is supported by the fact that when the same model was used with a logistic regression loss function, similar fluctuating results were obtained for the non-IID cases on MNIST, while the `LogisticRegression` model exhibited different behavior, as seen in Figure 4. It is worth noting that the `SVM` implementation in scikit-learn, which could potentially handle data skew similarly to the `LogisticRegression` model, does not support manual weight settings. As a result, it is not suitable for federated learning scenarios. These findings highlight the importance of considering the underlying implementation details and model choices when interpreting the results.

Therefore, an important avenue for future research is to conduct experiments on separate machines, as originally intended for federated learning, while also expanding the current implementation with new models and datasets. Furthermore, exploring new implementations and keeping the codebase up to date with the latest advancements in federated learning frameworks and libraries is crucial. This ensures compatibility with evolving standards and allows for the integration of new techniques and algorithms as they emerge. Regular updates and maintenance of the codebase guarantee reproducibility and facilitate the adoption of the research findings by the community.

[1] https://github.com/emilssipols11/FedML

References

[1] Brendan McMahan, Eider Moore, and Daniel Ramage. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54, pages 1273–1282, 2017.

[2] Qiang Yang, Yang Liu, Tianyi Chen, and Yu Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):12, 2019.

[3] Tianqing Li, Ananda Krishna Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[4] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. 2018.

[5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[6] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[7] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg,

Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: Large-scale machine learning on heterogeneous distributed systems, 2016.

[8] Federated Learning Community. Flower: A friendly federated learning research framework. PyPI, 2023.

[9] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]*, 2, 2010.

[10] Benjamin Merget, Samo Turk, Sameh Eid, Friedrich Rippmann, and Simone Fulle. Profiling prediction of kinase inhibitors: Toward the virtual assay. *Journal of Medicinal Chemistry*, 60(1):474–485, 2017. PMID: 27966949.