Performance comparison of different federated learning aggregation algorithms

How does the performance of different federated learning aggregation algorithms compare to each other?

Roy Katz[1]

Supervisors: Marcel Reinders[1]**, Swier Garst**[1]

[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: Roy Katz
Final project course: CSE3000 Research Project
Thesis committee: Marcel Reinders, Swier Garst, Lydia Chen

Abstract

Federated learning enables the construction of machine learning models, while adhering to privacy constraints and without sharing data between different devices. It is achieved by creating a machine learning model on each device that contains data, and then combining these models through an aggregation algorithm without sharing the data. Federated learning is currently a hot topic, and a lot of research has gone into implementing accurate aggregation algorithms. The original algorithm is FedAvg, and since then many different algorithms have been introduced. In this paper, I will compare the performance of five different aggregation algorithms: FedAvg, FedProx, FedYogi, FedMedian and q-FedAvg. The algorithms are compared on different data sets, namely MNIST and a kinase inhibition data set, as well as on different data distributions and number of clients. The experiments indicate that among these five algorithms, FedYogi achieves the best performance, both in terms of highest final accuracy as well as in terms of convergence rate.

1 Introduction

Federated learning is a new machine learning technique that trains machine learning models while adhering to privacy restrictions [7]. Instead of sharing data between various parties, privacy is achieved by the parties working together to create a global model while keeping their local data. They create the global model through multiple communication rounds. In each round, the clients receive the current global model and train it using their local data to create a local model. These local models are then combined using an aggregation algorithm to create a global model [10].

An example of where federated learning is useful is the medical records kept in hospitals. In this scenario, different hospitals maintain records of their patients. However, due to the sensitive nature of the data, they cannot share the data with other hospitals. The challenge is that each hospital does not have sufficient data to create an accurate model on their own. Alternatively, federated learning can be used to create a model that reflects all the data. The hospitals can communicate with each other in either a centralised or a decentralised manner. In a centralised setting, there would be an external, powerful and reliable server acting as the manager. This manager communicates with all the hospitals and aggregates the different local models created in each hospital to create the global model. On the other hand, this manager can also be made in a decentralised approach through blockchain. In both cases, the learning of the model is divided into multiple different learning rounds. During each learning round, the manager selects a predetermined number of hospitals and sends them the global model. The hospitals then perform a decided number of epochs over their data to update the model and create a local model. Afterwards, they send the new model back to the manager who aggregates all the new models into a new global model.

Due to the importance of creating a model with privacy sensitive information, federated learning became a significant technology that has been extensively researched. The topic started with the paper by McMahan, et. al. in 2016 [10], which explains the need for federated learning, describes the algorithm behind it, and evaluates it. Since then, a substantial amount of additional research has been conducted. This research is summarised well in the survey by Li, et. al. [7]. The survey summarises all the different aspects of federated learning and outlines the research done in these areas. Moreover, it describes various open source systems that can be used to implement federated learning, such as FATE [6], TFF [1] and PySyft [16]. Further research has gone into evaluating a federated learning implementation, such as the framework FedEval created by Chai, et. al. [4]. This framework provides a means to evaluate a federated learning implementation.

In the paper by McMahan, et. al. [10], which introduced federated learning, the aggregation algorithm of FedAvg was proposed. In this algorithm, the different local models are averaged to create the global model. Since then, many different algorithms have been developed.

One problem with FedAvg is its inability to effectively handle a variable number of epochs. Due to different data distributions and different computational power of different devices, different clients will be able to run different number of epochs. FedAvg addresses this issue by instructing each device to run a specified number of epochs within a given time limitation. If a device does not finish in time, it is dropped from the current iteration. However, this approach was proven to be substandard in the paper by Wang, et. al. [14]. The paper proved that the model converges to a point that can be arbitrarily different from the true objective. Instead, the paper suggests the algorithm FedNova, which allows each client to have a variable number of epochs. To ensure fairness among clients, the local models are normalised. On the other hand, Li, et. al. [8] propose a different solution with their algorithm of FedProx. FedProx behaves in a similar manner to FedNova, but rather than normalising the local models, it uses a proximal term. This term determine the maximum allowed difference between the weights of the local model and the global model.

The creation of aggregation algorithms became one of the prominent research topics in federated learning. A great variety of aggregation algorithms have been developed to achieve the best machine learning model in different scenarios. One of these aggregation algorithms is FedMa, which tries to match the neurons between the local and global model better [13]. Three additional algorithms were introduced in the paper by Reddi, et. al. [12]. The paper introduced the class of aggregation algorithms called FedOpt. This class includes the algorithms of FedYogi, FedAdagrad and FedAdam [12]. The aggregation algorithms of the FedOpt class are built on top of FedAvg by changing the learning rate of the model throughout its creation. It is important to note that except of the FedOpt aggregation algorithm class, there is an unrelated FedOpt aggregation algorithm implemented by Asad, et. al. [3]. An alternative algorithm is FedMedian. As the name suggests, the idea behind this algorithm is to use the median of the weights rather than the mean [15]. Another aggrega-

tion algorithm is q-FedAvg proposed by Li, et. al. [9]. This algorithm claims to achieve the same performance as FedAvg while being more fair. The fairness is demonstrated by obtaining more similar accuracies across the testing data of each client. To achieve a higher fairness, the objective function is tweaked to favour low achieving clients.

The research question I will be answering is: How does the performance of different federated learning aggregation algorithms compare to each other? The choice of aggregation algorithm in federated learning significantly affects the resulting model's performance, therefore it is important to understand the strengths and weaknesses of different aggregation algorithms. Thus, I will compare the performance of different aggregation algorithms in different scenarios. The algorithms I will compare are FedAvg, FedProx, FedYogi, FedMedian and q-FedAvg.

The paper starts with an explanation of the experiment in section 2. Afterwards the results are shown and explained in section 3. This is followed by discussion of the ethical concerns of the research in section 4. Section 5 continues by discussing the validity and limitations of the results. Lastly section 6, concludes the findings of the research.

2 Methodology

In order to answer the research question, I implemented federated learning using the Flower framework [2]. Then I ran the implementation multiple times, with different aggregation algorithms, different data distributions between parties, different number of parties and with different data sets. Afterwards, I analysed the accuracy of the models and compared the performance of the different aggregation algorithms, as well as to the model created with classic machine learning (where all the data was pooled together into a single party).

The decision to use the Flower framework was made due to the framework's simplicity and flexibility. Due to the heavy time constraint of this project, it was important to choose a framework that can be quickly learned while having great capabilities. The Flower framework is exactly that. It allows for a quick implementation of federated learning, while giving a lot of flexibility in aggregation algorithms, scalable number of clients and different machine learning models.

I decided to focus on the accuracy of the global model rather than the different local models, as that is the model most closely resembling classic machine learning. However, I also compared the local accuracy of q-FedAvg and FedAvg. This is because the implementation of q-FedAvg aims to achieve a more fair local accuracy. I focused on accuracy itself, due to the main difference between different aggregation algorithms is the performance of the model they create. Naturally, other aspects of federated learning are also important to analyse, such as privacy and efficiency. Nevertheless, these aspects are often determined by the other parts of the implementation, other than the aggregation algorithm itself. The reason behind choosing accuracy as the performance metric, is due to its simplicity and ability to assess the model in all scenarios. Furthermore, the papers on the aggregation algorithms I compared, used accuracy to justify their respective algorithm.

As mentioned earlier, each aggregation algorithm was tested in multiple different scenarios and on two different data sets. The two different data sets I used are MNIST [5] and a kinase inhibition data set [11]. MNIST was chosen due to its simplicity and widespread use. As well as four of the five aggregation algorithms I am comparing, have used MNIST to justify their algorithm in their respective papers (only q-FedAvg did not) [8; 10; 12; 15]. On the other hand, the kinase inhibition data set was selected to demonstrate a real life use of federated learning. The data set was constructed by combining three different data sets, and as a result the data was collected from different parties.

Creating a federated learning model on the MNIST data set was done on six different scenarios. These scenarios include:

- IID scenario with equal distribution - where each party received an equal distribution of each class, with 500 instances of each class. The distribution can be seen in Figure 1.

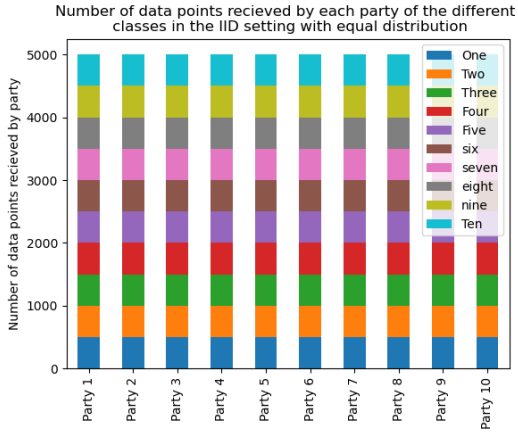

Figure 1: Data distribution scenario of IID and equal distribution

- Non-IID Scenario with equal distribution - where each party received an equal distribution of a total of 2 classes, with 2500 instances of each class. The distribution can be seen in Figure 2.

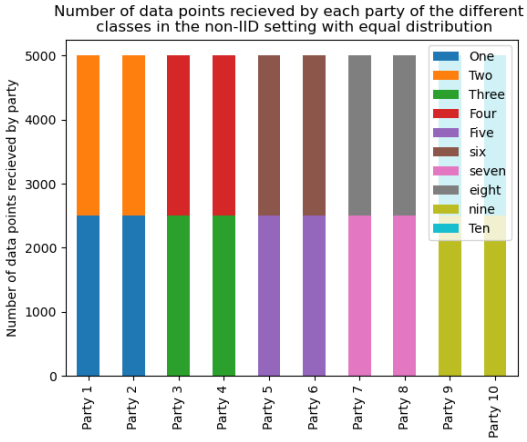

Figure 2: Data distribution scenario of non-IID and equal distribution

- IID scenario with different distribution - Where every party received all the different classes of the MNIST data set with equal distribution, but half of the parties received four times as much data. In order to maintain a practical scenario, the parties with less data ran 4 epochs instead of 1. The distribution can be seen in Figure 3.

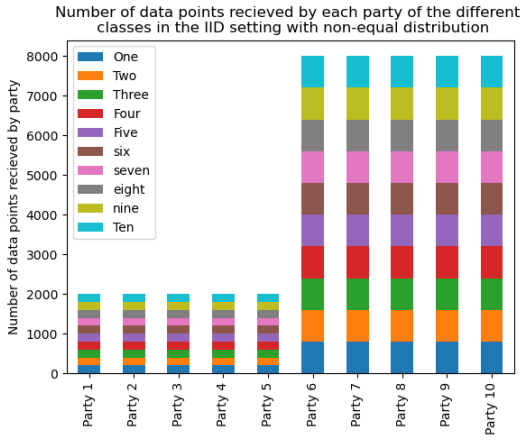

Figure 3: Data distribution scenario of IID and non-equal distribution

- Non-IID Scenario with different distribution - Where each party received an equal distribution of 2 classes, but half the parties received 4 times as much data. In order to maintain a practical scenario, the parties with less data ran 4 epochs instead of 1. The distribution can be seen in Figure 4.

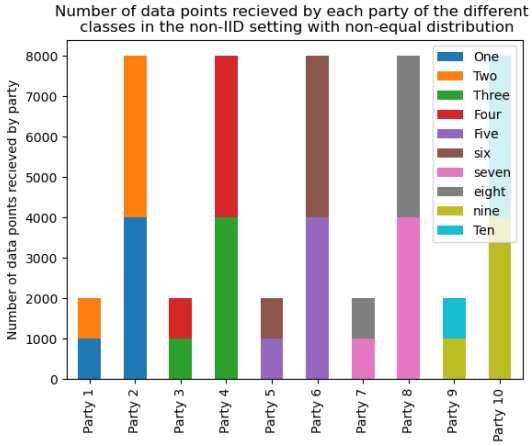

Figure 4: Data distribution scenario of non-IID and non-equal distribution

- IID Scenario with exceedingly number of parties - An IID scenario with 50 parties instead of 10, where in each communication round 7 parties were contacted. The data was distributed in such a way that each client received $\frac{1}{50}$ of all the data in an equal distribution. The distribution of data was done in the same fashion as in scenario 1.
- Non-IID Scenario with exceedingly number of parties - A non-IID scenario with 50 parties, where in each communication round 7 parties were contacted. The data is distributed in a manner such that each client received $\frac{1}{10}$ of all the data of two different classes. It was done in the same fashion as scenario 2.

Creating a federated learning model on the kinase inhibition data set involved a single scenario. The scenario was that the data was distributed between three different parties in the same fashion as the data was originally collected. This scenario analysed federated learning in a more realistic setting.

To ensure the value of the data collected in this research for other researchers, I followed the most common type of federated learning in research. The implemented federated learning communicated in a centralised manner, where all the communication and aggregation were carried out by a trustworthy third party. Both data sets were horizontal data sets, meaning that data sets of each party shared the same vector space. The machine learning model that was used was a neural network. The architecture of the neural network created can be found in Appendix A.2. The federated learning was a cross silo federated learning, meaning there was a small number of parties compared to the large amount of data. More specific settings include:

- In the first 4 scenarios of MNIST there were a total of 10 clients.
- The learning rate of the model was set at 0.0001.
- There was one epoch per communication round. This was the case in all scenarios except of the ones with non-equal distribution. In those scenarios the clients with less data ran 4 epochs.
- There were a total of 500 Communication rounds.
- The aggregation algorithms that were compared are FedAvg, FedProx, FedYogi, FedMedian and q-FedAvg.

I chose to compare FedAvg and FedProx as they are both some of the most commonly used federated learning algorithms, due to their simplicity and good performance. Furthermore, I decided to analyse FedYogi, as the FedOpt class of aggregation algorithms have been shown to perform well. FedYogi was specifically chosen, because it has shown the best performance among the FedOpt algorithms. I chose to compare FedMedian, due to its uniqueness in using a median rather than a mean, as well as its lack of performance documentation. This lack of documentation is a result of the algorithm being developed for distributed learning rather than federated learning. Lastly, I decided to compare q-FedAvg as instead of aiming to perform better, it aims for a more fair performance.

3 Results

Figures 5-8 show the accuracy of the aggregation algorithms per communication round in the four different scenarios of 10 parties in MNIST. Unlike the research done by Wang, et. al. [14], I noticed that there was minimal to non-existent impact on having a non-equal distribution. Although in their paper they emphasized the impact of non-equal distribution and variations in the number of epochs have on the overall model. They justify it by explaining that the model does not converge to the true objective. However, Figures 5 and 6,

as well as Figures 7 and 8, are almost identical, which contradicts their statement. The difference between our results could have been caused by the use of a simulation in this research and the difficulty in simulating a realistic scenario. As the federated learning was simulated on a single device, the number of epochs ran on each device was set in advance. In order to simulate a more practical environment, in the scenarios where some devices had 4 times the amount of data compared to other devices, the devices with less data ran 4 times the number of epochs. While this simulated a more practical scenario, it did not take into account possible differences in devices computation ability or specific parties not finishing the required epochs in time. As a result, the simulation in this research did not simulate a completely realistic scenario.

There is a lot of valuable information to be taken out of Figures 5 and 6. These figures show that on IID data sets, federated learning achieves nearly identical accuracy compared to classic machine learning. In particular, the FedYogi aggregation algorithm does achieve the same accuracy. Although it converges in a slightly slower manner. On the other hand, FedAvg, FedProx, q-FedAvg and FedMedian all achieve slightly lower accuracy and convergence rate compared to classic machine learning.

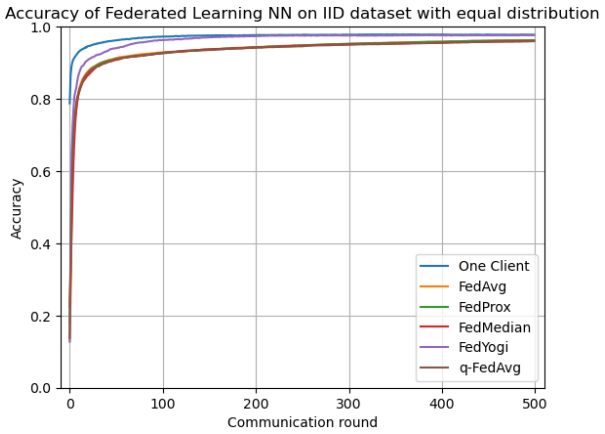

Figure 5: Accuracy on MNIST on IID data set of equal distribution

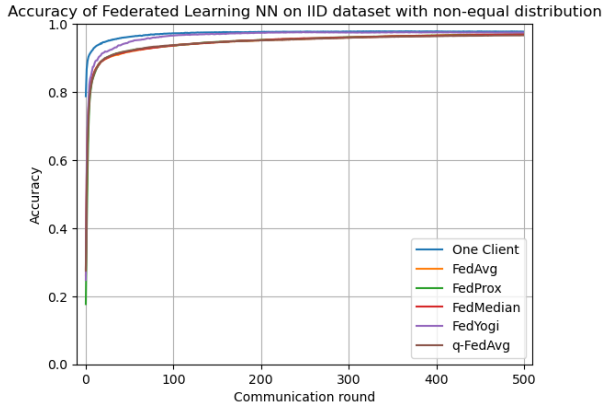

Figure 6: Accuracy on MNIST on IID data set of non-equal distribution

Figures 7 and 8 illustrate the accuracy of the models when the data distribution is non-IID. In this scenario, there is a notable difference in accuracy between the different aggregation algorithms. FedYogi does achieve the highest accuracy, with a considerable margin. Nevertheless, the difference in accuracy between FedYogi and classic machine learning is substantial. Additionally, there is a visible instability of the FedYogi model in both of these scenarios. After FedYogi, the algorithms of FedAvg, FedProx and q-FedAvg achieve very similar accuracy, but with a large decrease in performance when compared to FedYogi. Finally, there is another large drop in accuracy when comparing FedMedian.

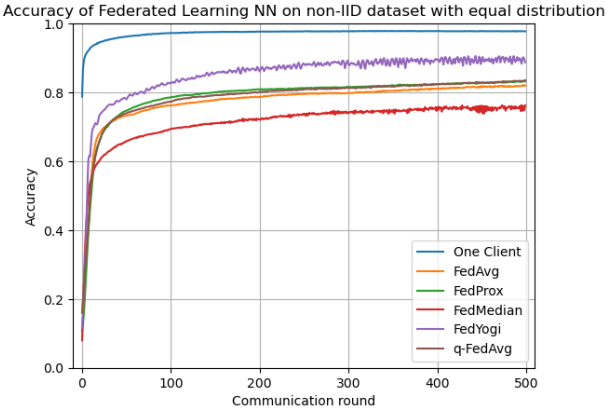

Figure 7: Accuracy on MNIST on non-IID data set of equal distribution

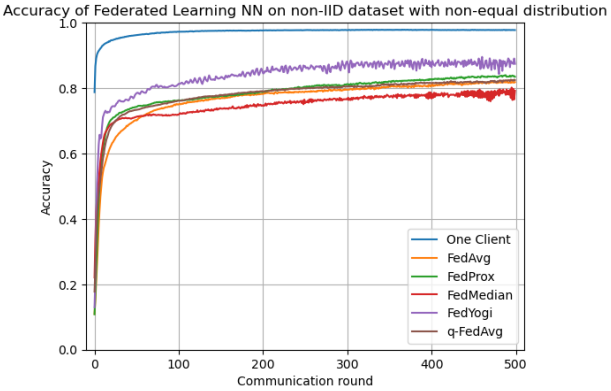

Figure 8: Accuracy on MNIST on non-IID data set of non-equal distribution

After experimenting with federated learning with 10 clients, I proceeded to evaluate the algorithms in scenarios with 50 clients. Due to memory limitation of Flower simulation, only 7 parties were contacted in each communication round, and the neural network architecture was slightly smaller. The neural network architecture can be found in Appendix A.2. Figure 9 presents the accuracy of the models in a scenario with 50 clients, where the data is distributed equally in an IID fashion. In this scenario FedYogi achieves the highest accuracy, but this accuracy is lower than classic machine

learning. Subsequently, FedMedian, FedAvg, FedProx and q-FedAvg all achieve a similar accuracy, which is substantially lower than FedYogi. Furthermore, all four algorithms do have different convergence rates.

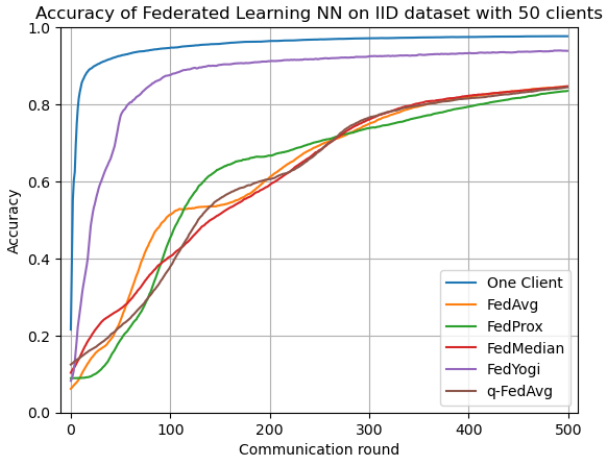

Figure 9: Accuracy of MNIST on IID data set with 50 clients

Figure 10 shows the accuracy of the different aggregation algorithms with 50 clients and non-IID data. As only 7 parties were contacted in each round, there were rounds where some classes were not present, resulting in instability of all the accuracies. Also in this case FedYogi achieved the highest accuracy, which is followed by FedProx, q-FedAvg and FedAvg, and lastly FedMedian achieved the worse.

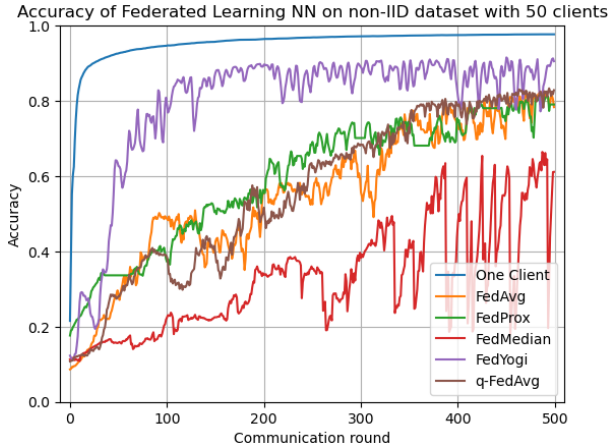

Figure 10: Accuracy of MNIST on non-IID data set with 50 clients

Lastly, I tested the aggregation algorithms on a more realistic scenario with the kinase inhibition data set. Figure 11 displays the accuracy in this more realistic scenario. In this scenario, FedAvg, FedProx, q-FedAvg and FedYogi exhibit similar performance, achieving accuracy comparable to that of a single client. On the other hand, FedMedian performs notably worse. An additional observation from the data is that all the plots show a slight decrease after reaching their respective maximas. The decrease can be explained by overfitting on the training data.

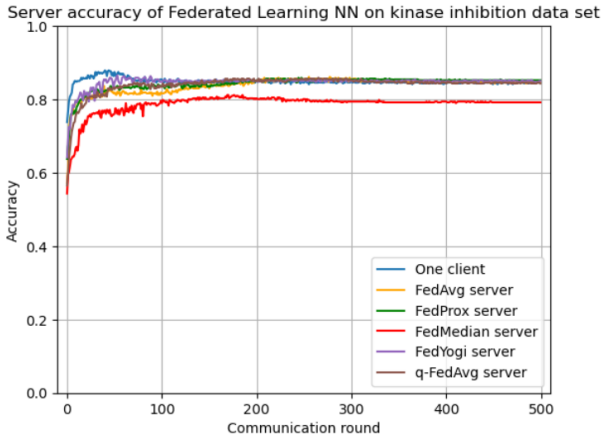

Figure 11: Accuracy on Kinase inhibition data set

The local accuracies of q-FedAvg and FedAvg were compared and the results are shown in Figure 12. This comparison is interesting because q-FedAVg aims to achieve more fair results. According to the definition of fairness by Li, et. al. [9], fairness is evaluated by analysing the similarity of accuracies in different clients. From the graph, it can be observed that FedAvg achieves the same or slightly higher fairness than q-FedAvg. The difference in fairness between FedAvg and q-FedAvg is minor, therefore the graph does not prove nor disprove the statement of Li, et.al [9] that q-FedAvg achieves a higher fairness than FedAvg. Furthermore, q-FedAvg and FedAvg have achieved a very similar accuracy in all seven scenarios. This coincides with the paper's statement that the global accuracy of FedAvg and q-FedAvg is comparable [9]. The local accuracies of FedProx, Fedmedian and FedYogi can be found in Appendix A.1

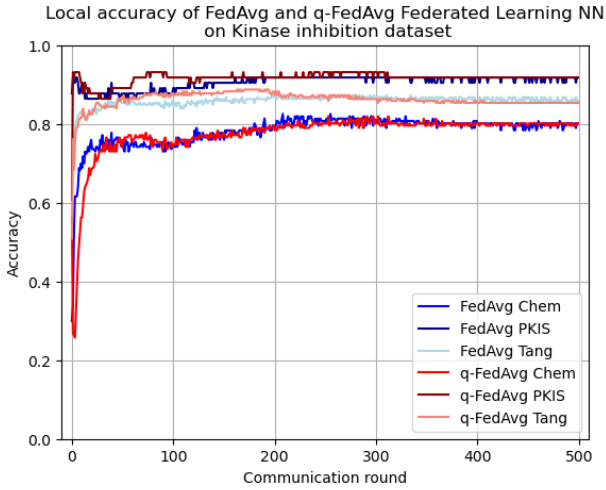

Figure 12: Local accuracy on kinase inhibition dataset of FedAvg and q-FedAvg

In all seven scenarios, FedYogi achieved the highest accu-

racy or shared the highest accuracy among the aggregation algorithms. However, it exhibit slight instability in the non-IID cases. My hypothesis is that due to the algorithm changing the learning rate, it allowed for a more diverse exploration of the search space. This ability to explore different regions of the search space may enable the model to avoid getting stuck in local maximas.

After FedYogi, the algorithms of FedProx, q-FedAvg and FedAvg consistently achieved the next best performance, with their accuracies being very similar across all seven scenarios. Conversely, FedMedian had the lowest or shared lowest performance in all seven scenarios.

4 Responsible Research

When conducting this research, ethical methods were followed to ensure the integrity of this study. The data sets used are open source material, that are available to be used for further research. Similarly, frameworks used are open source, that are available online. Throughout the research process, the Netherlands Code of Conduct for Research Integrity was followed. All the data constructed in the experiments was constructed by following the information in the methodology. The methodology was written with a lot of detail, to ensure it can be used to recreate the experiments. The code for the experiment can be found online[1]. It is important to note that not all data created was afterwards used. Specifically the data collected of FedYogi and FedMedian when running 50 clients and non-IID distribution. Instead of using the original data, it was simulated again. This was because in the original simulation, the computer ran out of memory, and as a result many communication rounds did not run correctly. To compare the accuracies in a more fair manner, the simulation was rerun.

5 Discussion

The research done in this paper provides useful insight when selecting an aggregation algorithm. It is evident that FedYogi consistently achieved the best performance in the seven scenarios. On the other hand, FedMedian consistently had the lowest accuracy.

Upon reviewing the papers introducing the five aggregation algorithms, it is evident that the accuracy achieved in this research on the MNIST IID dataset is on par with the papers for FedMedian and FedProx. However, there was a large drop of accuracy in the results from this research compared to the results in the papers of FedAvg and FedYogi. One potential explanation is the difference in implementation of the neural network. As the neural networks used in their experiment were considerably larger.

While the reliability of the results is improved by having on par accuracy with other research, the ability to effectively compare the aggregation algorithms can always be improved. Each federated learning scenario is different and each scenario may require different aggregation algorithms for optimal performance. By expanding the analysis to include a winder range of scenarios, researchers can compare aggregation algorithms better.

[1] https://github.com/roykatz10/RP_FL/tree/main

Further explorations of scenarios with more parties or more parties running in each communication round would be valuable. This scenarios are very important to explore, due to the use of federated learning in creating machine learning models of private data from our phones. The memory limitation of the Flower framework in simulating on one device more than 50 clients, or more than 7 clients per communication round, posed challenges in conducting such experiments in this research.

Another scenario that could have been explored further is a different neural network implementation. The neural network used in this research was relatively small. Similarly to having more clients, when having a larger neural network there were problems in terms of memory. A solution that can work in both cases is using more devices when simulating the federated learning.

Other than exploring more scenarios, it is also important to explore more data sets. While both data sets used in this research effectively test the performance of federated learning, conducting tests with additional data sets would improve our ability to compare the algorithms.

Lastly, in order to thoroughly evaluate the performance of the aggregation algorithms, they should be tested on a broader range of data distributions. The non-IID distribution used, tested a very extreme scenario, which stressed the impact of a non-IID distribution on the performance of the algorithms. Nevertheless, it is important to explore a more diverse data distributions to gain a comprehensive understanding of the algorithms' performance.

Apart from comparing the algorithms in more scenarios, a possible future work is to compare additional algorithms. Some other algorithms worth exploring include FedNova and FedMa, which were mentioned earlier. I decided not to compare FedMa due to memory constraints. Since I used a small neural network, the nodes between the different local models would not switch as much. Furthermore, FedNova is not compared, because it aims to improve scenarios where different clients run a different number of epochs due to varying device capabilities and data distribution. However, when simulating such a scenario, it did not have a significant impact on the overall model's performance. Similarly to testing additional scenarios, five algorithms were compared due to the time limitation of the project.

6 Conclusion

With the increased use of federated learning, the ability to create an accurate federated learning model becomes more important. The choice of aggregation algorithm has a great impact on the overall model, thus selecting the correct aggregation algorithm is essential. I compared the algorithms of FedAvg, FedProx, FedYogi, q-FedAvg and FedMedian. By comparing them with several different scenarios and on two different data sets, it is clear that FedYogi consistently outperformed the other algorithms in terms of overall accuracy as well as convergence rate. While FedYogi achieved the highest accuracy, it is important to consider the main goal of federated learning in specific scenarios. For example, if fairness is critical, fair-based aggregation algorithms, such as q-FedAvg

should be considered.

In terms of performance, FedAvg, q-FedAvg and FedProx achieved a very similar performance levels, all falling below that of FedYogi. Lastly, FedMedian consistently achieved the lowest performance. From the results, it can be stated that FedYogi achieves the highest performance in the tested scenarios. That being said, it is important to remember that there are different scenarios where different aggregation algorithms may work better. Thus, it is important to compare the performance of these algorithms in more scenarios. Possible additional scenarios include testing on other data sets, testing on a greater variety of neural networks, testing with a different number of clients, testing with different data distributions, as well as testing additional aggregation algorithms.

A Appendix

A.1 Local Accuracies

Figures 13-15 show the local accuracies for FedProx, FedYogi and FedMedian on the kinase inhibition data set.

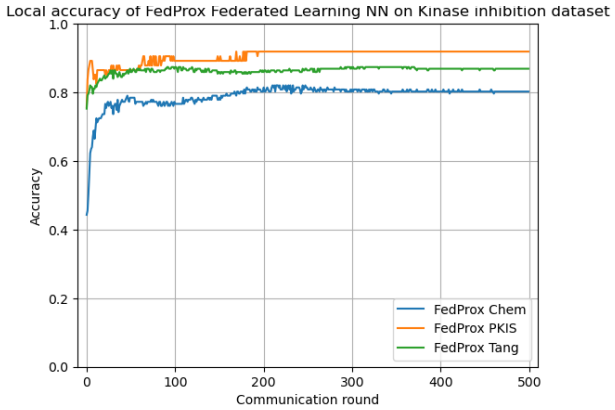

Figure 13: Local accuracy on kinase inhibition dataset of FedProx

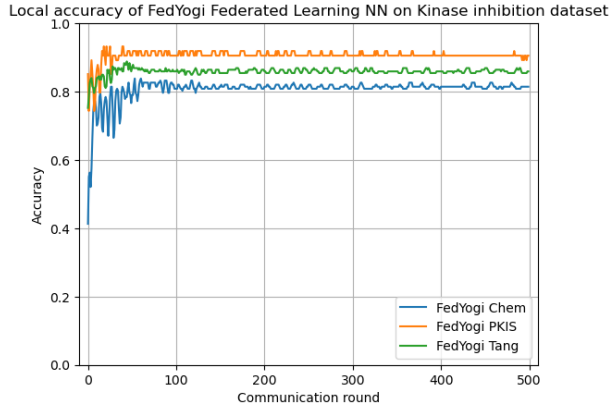

Figure 14: Local accuracy on kinase inhibition dataset of FedYogi

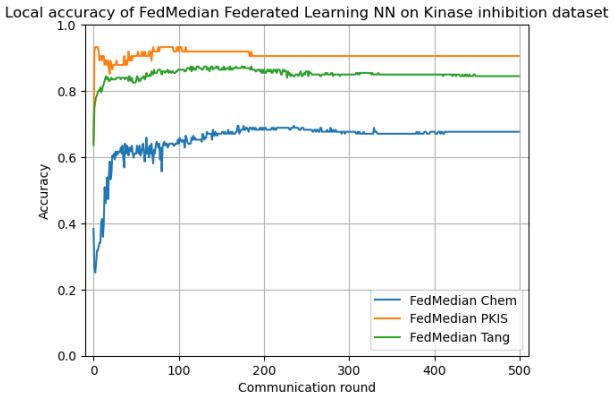

Figure 15: Local accuracy on kinase inhibition dataset of Fedmedian

A.2 Neural Network Implementation

Table 1 shows the architecture of the neural networks used.

	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
MNIST 10 clients	784× 128	ReLU	128× 256	ReLU	256×10
MNIST 50 clients	784× 128	ReLU	128×64	ReLU	64×10
Kinase Inhibition Dataset	8192× 128	ReLU	128× 256	ReLU	256×2

Table 1: The neural network architecture

References

[1] 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, Dandelion Mané, 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 Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Adap GmbH. Flower.

[3] Muhammad Asad, Ahmed Moustafa, and Takayuki Ito. Fedopt: Towards communication efficiency and privacy preservation in federated learning. *Applied Sciences*, 10(8), 2020.

[4] Di Chai, Leye Wang, Liu Yang, Junxue Zhang, Kai Chen, and Qiang Yang. Fedeval: A holistic evaluation framework for federated learning, 2022.

[5] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[6] FADAI. Fate.

[7] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):3347–3366, apr 2023.

[8] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks, 2020.

[9] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning, 2020.

[10] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data, 2023.

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

[12] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H. Brendan McMahan. Adaptive federated optimization, 2021.

[13] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging, 2020.

[14] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization, 2020.

[15] Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates, 2021.

[16] Alexander Ziller, Andrew Trask, Antonio Lopardo, Benjamin Szymkow, Bobby Wagner, Emma Bluemke, Jean-Mickael Nounahon, Jonathan Passerat-Palmbach, Kritika Prakash, Nick Rose, Théo Ryffel, Zarreen Naowal Reza, and G. Kaissis. Pysyft: A library for easy federated learning. 2021.