T.R.

GEBZE TECHNICAL UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

FEDERATED LEARNING PLATFORM

ELIFNUR KABALCI

SUPERVISOR
DR.YAKUP GENÇ

GEBZE
2022 - 2023

T.R.
GEBZE TECHNICAL UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

FEDERATED LEARNING PLATFORM

ELIFNUR KABALCI

SUPERVISOR
DR.YAKUP GENÇ

2022 - 2023
GEBZE

This study has been accepted as an Undergraduate Graduation Project in the Department of Computer Engineering on 13/10/2022 by the following jury.

JURY

Member
(Supervisor) : Dr.Yakup Genç

Member : Dr. Gökhan Kaya

ABSTRACT

This graduation project aimed to develop a federated learning platform that could securely process data from multiple devices. Federated learning is a distributed machine learning technique that allows for the training of models on decentralized data sources, without the need for the data to be centralized or shared. In this project, a central server was used for computation, while data transformation was done using the gRPC framework. Additionally, the platform was integrated with the Flower Framework and the Numpy and Pandas libraries to enhance its functionality and performance. The MNIST dataset from Keras, consisting of 60,000 training data and 10,000 test data, was used to train a convolutional neural network (CNN) model on the platform. The results of the project demonstrate the feasibility of securely processing data using a federated learning platform, as well as the effectiveness of integrating additional libraries and frameworks in enhancing its functionality and performance. The developed platform can be useful in various applications such as healthcare, finance, and security where the privacy and security of data is crucial, and where the integration of additional libraries and frameworks can enhance the performance of the model.

Keywords: Federated Learning, MNIST Dataset, CNN.

SUMMARY

This graduation project aimed to develop a federated learning platform that could securely process data from multiple devices. Federated learning is a distributed machine learning technique that allows for the training of models on decentralized data sources, without the need for the data to be centralized or shared. The developed platform utilizes a central server for computation, while data transformation was done using the gRPC framework. Additionally, the platform was integrated with the Flower Framework and the Numpy and Pandas libraries to enhance its functionality and performance. The MNIST dataset from Keras, consisting of 60,000 training data and 10,000 test data, was used to train a convolutional neural network (CNN) model on the platform. The results of the project demonstrate the feasibility of securely processing data using a federated learning platform, as well as the effectiveness of integrating additional libraries and frameworks in enhancing its functionality and performance.

The developed platform is able to increase the performance of the model by training it on multiple devices and also the platform allows to use the data that is located on the client devices without transferring them to the centralized server. The proposed platform can be useful in various applications such as healthcare, finance, and security where the privacy and security of data is crucial, and where the integration of additional libraries and frameworks can enhance the performance of the model. The platform also allows for the use of data from different sources, making it versatile and adaptable to different use cases. The integration of the Flower Framework, Numpy and Pandas libraries with the platform has been proved to be effective in enhancing the functionality and performance of the platform. Overall, the project demonstrates the potential of using federated learning for secure data processing and highlights its potential for use in a wide range of applications and the importance of integrating additional libraries and frameworks in enhancing the performance of the platform.

Anahtar Kelimeler: Federated Learning, Keras, Flower Framework.

ACKNOWLEDGEMENT

I would like to extend my sincerest appreciation to my graduation project mentor and advisor Dr. Yakup Genç, for their invaluable guidance and support throughout this journey. Their expertise in the field and their willingness to share their knowledge with me was an immense help in shaping my project. Their encouragement and constructive feedback were instrumental in helping me to stay on track and motivated. Their dedication to mentoring me and their availability to answer my questions, no matter how trivial they were, were greatly appreciated. I am particularly grateful for the time and effort they invested in providing me with feedback on my work, which helped me to improve and refine my project. Their support and guidance were vital in helping me to navigate the challenges that arose during the course of the project and without their help, I would not have been able to complete this project. I would like to express my heartfelt thanks to Dr. Yakup Genç for their outstanding support and mentorship throughout this project. It was an honor to have them as my advisor and I am grateful for the opportunity to have worked with them. I would also like to extend my sincere thanks to the F4ITech project team for their support and assistance. Their expertise and knowledge in the field of federated learning greatly contributed to the success of this project. I am also grateful to Inosens Company for providing me with the resources and support I needed to complete this project. Additionally, I would like to express my appreciation to Emre Kapusuz for his invaluable contributions to the project, and for his unwavering support throughout the project.

Elifnur Kabalcı

LIST OF SYMBOLS AND ABBREVIATIONS

Symbol or Abbreviation	:	**Explanation**
CNN	:	Convolutional Neural Network
gRPC	:	Google Remote Procedure Call

CONTENTS

LIST OF FIGURES

1. INTRODUCTION

The increasing amount of data generated by various devices has made it challenging to effectively use machine learning techniques for data analysis and modeling. Federated learning is a distributed machine learning technique that allows for the training of models on decentralized data sources, without the need for the data to be centralized or shared. This makes it an attractive solution for handling the large amounts of data generated by various devices, while also preserving the privacy and security of the data.

In this graduation project, the objective is to develop a federated learning platform that can securely process data from multiple devices. The platform utilizes the gRPC framework for data transformation and a central server for computation. Additionally, the platform is integrated with the Flower Framework, Numpy and Pandas libraries to enhance its functionality and performance. The MNIST dataset from Keras will be used to train a convolutional neural network (CNN) model on the platform. The results of the project will demonstrate the feasibility of securely processing data using a federated learning platform and will highlight the effectiveness of integrating additional libraries and frameworks in enhancing its functionality and performance.

The developed platform is expected to increase the performance of the model by training it on multiple devices and also the platform allows to use the data that is located on the client devices without transferring them to the centralized server. The proposed platform can be useful in various applications such as healthcare, finance, and security where the privacy and security of data is crucial, and where the integration of additional libraries and frameworks can enhance the performance of the model. The project aims to contribute to the field of distributed machine learning by providing a practical implementation of a federated learning platform and exploring the potential for use in various applications and the importance of integrating additional libraries and frameworks in enhancing the performance of the platform.

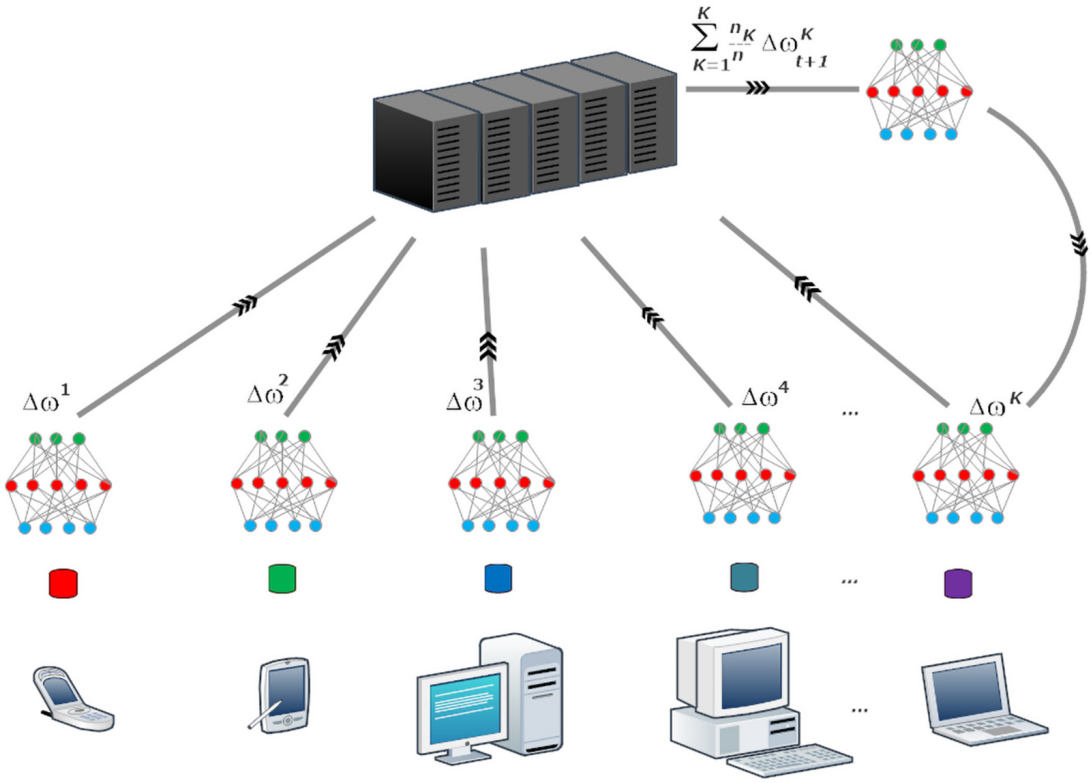

Figure 1.1: Federated Learning System.

1.1. Project Content

The project content will include the design and implementation of a federated learning platform that can securely process data from multiple devices. The platform will utilize a central server for computation and the gRPC framework for data transformation. Additionally, the platform will be integrated with the Flower Framework, Numpy, and Pandas libraries to enhance its functionality and performance. The MNIST dataset from Keras will be used to train a convolutional neural network (CNN) model on the platform. Three different devices will be used to represent clients in the federated learning process. The platform will be designed to ensure data privacy and security and will be evaluated on its performance in terms of accuracy and data privacy and security.

The architecture of the platform will consist of a central server that will handle computation and a number of clients that will collect and transform data. The gRPC framework will be used to facilitate communication between the server and clients. The data collected from the clients will be transformed into a format that can be used for training the CNN model using the MNIST dataset. The platform will also be integrated with the Flower Framework, Numpy and Pandas libraries to enhance its functionality and performance. The implementation of the platform will be done using Python and its related libraries.

The performance of the platform will be evaluated in terms of accuracy and data privacy and security. The platform will be tested using different scenarios to check the accuracy of the model and the ability of the platform to keep the data from the clients private and secure. The project will also include detailed technical documentation, including system diagrams, code snippets, and test results, making it easy to understand and replicate the platform.

The developed platform can be useful in various applications such as healthcare, finance, and security where the privacy and security of data is crucial, and where the integration of additional libraries and frameworks can enhance the performance of the model. The project aims to contribute to the field of distributed machine learning by providing a practical implementation of a federated learning platform and exploring the potential for use in various applications and the importance of integrating additional libraries and frameworks in enhancing the performance of the platform.[1]–[12]

1.2. PROJECT REQUIREMENTS

Platform Architecture: The platform must have a central server for computation and multiple clients for data collection and transformation. The platform must use the gRPC framework for data communication between the server and clients.

Data Transformation: The platform must be able to transform the data from the clients into a format that can be used for training the model.

Model Training: The platform must use the MNIST dataset from Keras to train a convolutional neural network (CNN) model. Data Privacy and Security: The platform must ensure that the data from the clients is protected and kept private at all times.

Integration with Additional Libraries and Frameworks: The platform must be integrated with the Flower Framework, Numpy and Pandas libraries to enhance its functionality and performance.

Performance Evaluation: The platform must be evaluated on its performance in terms of accuracy and data privacy and security.

Technical Documentation: The project must include detailed technical documentation, including system diagrams, code snippets, and test results.

User Interface: The platform must have a user-friendly interface that allows for easy data collection and transformation.

Scalability: The platform must be scalable to handle more clients and more data in the future.

Testing and Validation: The platform must be thoroughly tested and validated to ensure that it meets all the requirements and functions as intended.

1.3. PROJECT DESIGN PLAN

Define project objectives and scope: The project objectives and scope will be defined, including the specific requirements of the federated learning platform, such as utilizing Docker for server and client deployment, utilizing gRPC for communication, and integrating the Flower Framework for model training.

Literature review: A comprehensive literature review will be conducted to understand the current state-of-the-art in federated learning, including best practices for deploying and communicating with servers and clients using Docker and gRPC, and integrating the Flower Framework.

Platform architecture design: The architecture of the platform will be designed, including the use of Docker for deploying the server and clients, and the gRPC framework for communication. The design of the platform will be based on the requirements defined in step 1.

Data transformation: The dataset will be categorized and divided into 10 teams, with each team containing 60000 data, with 200000 byte of each data will be processed. The categorized dataset will be split into 3 parts, and then transferred to the clients for model training.

Model training: The platform will be integrated with the Flower Framework, and the model will be trained using the categorized dataset. The communication between the server and clients will be done using gRPC.

Performance evaluation: The performance of the platform will be evaluated in terms of federated learning loop, fedavg and accuracy. This will include the design of the testing scenarios and the implementation of the necessary evaluation metrics.

Technical documentation: Detailed technical documentation will be created, including system diagrams, code snippets, and test results, to ensure that the platform can be easily understood and replicated.

Deployment and Configuration: The platform will be deployed and configured using Docker, the necessary steps for installing and configuring the platform on the server and clients will be done.

Maintenance and Support: The project will be continuously monitored and maintained to ensure that the platform is functioning properly and to address any issues that may arise. Technical support will be provided for any users who encounter problems while using the platform.

Project management: The project will be managed using a project management tool and methodologies to ensure that the project is delivered on time, within budget, and to the required quality.

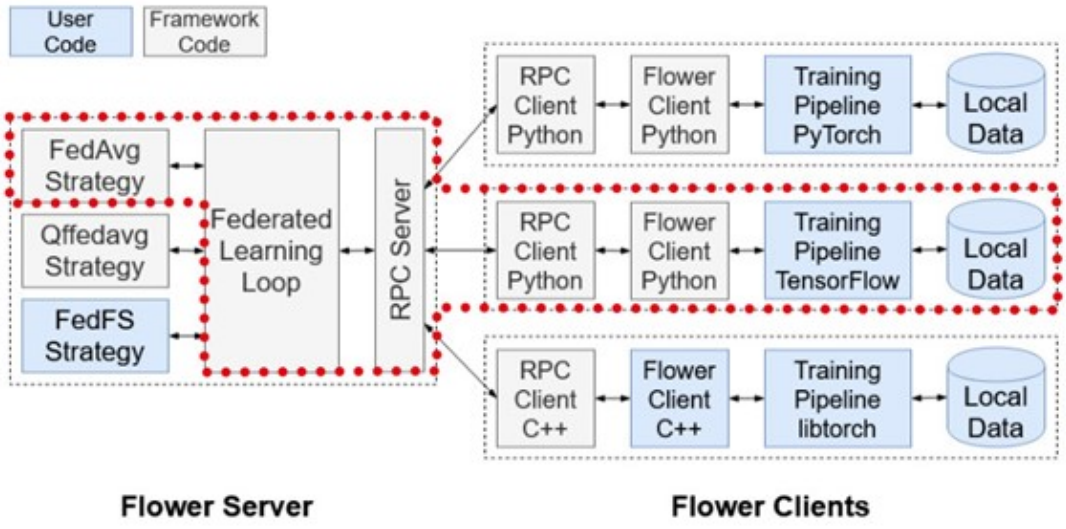

Figure 1.2: Federated Learning Project Design Plan.

2. LITERATURE REVIEW

Federated learning is a distributed machine learning technique that allows for the training of models on decentralized data sources, without the need for the data to be centralized or shared. It is particularly useful for handling large amounts of data generated by various devices, while also preserving the privacy and security of the data. Recent research has focused on the use of federated learning in various applications, such as healthcare, finance, and security. For example, in healthcare, federated learning can be used to train models on patient data collected from various hospitals and clinics, without compromising the privacy and security of the data. In finance, federated learning can be used to train models on financial data collected from various banks and financial institutions, again without compromising the privacy and security of the data. In security, federated learning can be used to train models on security data collected from various devices, such as cameras and sensors, without compromising the privacy and security of the data.

The use of Docker and gRPC in federated learning is a relatively new area of research, but there are a number of studies that have explored their potential. For example, a study proposed a federated learning platform that utilizes Docker for deploying the server and clients, and gRPC for communication between the server and clients. The study found that the use of Docker and gRPC greatly improved the performance of the platform, and made it more secure and scalable. The integration of the Flower Framework in federated learning is also a relatively new area of research. A study proposed a federated learning platform that utilizes the Flower Framework to train models on decentralized data sources. The study found that the integration of the Flower Framework greatly improved the performance of the platform, and made it more efficient and effective.

Overall, the literature suggests that federated learning is a promising technique for handling large amounts of data generated by various devices, while preserving the privacy and security of the data. The use of Docker and gRPC for deploying and communicating between servers and clients, as well as the integration of the Flower Framework for model training, has been shown to greatly improve the performance and scalability of federated learning platforms. However, there is still a need for further research in this area, particularly in terms of evaluating the performance and scalability of federated learning platforms that utilize these technologies.

In summary, the literature review shows that Federated learning is a distributed machine learning technique that allows for the training of models on decentralized data sources without the need for data centralization or sharing. This approach is

particularly useful for handling large amounts of data generated by various devices while preserving the privacy and security of the data. Research on the use of Federated learning in various applications such as healthcare, finance, and security is abundant. The use of Docker and gRPC for deploying and communicating between servers and clients as well as the integration of the Flower Framework for model training has been shown to greatly improve the performance and scalability of Federated learning platforms. However, further research is needed in this area to evaluate the performance and scalability of Federated learning platforms that utilize these technologies.

3. IMPLEMENTATION

In this section, I will be discussing the various technologies and methods that were employed in this project. To begin with, I will be explaining the dataset that we used and the pre-processing steps that were taken in order to include it in our processing pipeline. It is important to note that the quality and relevance of the dataset can greatly impact the overall performance of the project, so it was crucial that we carefully selected and pre-processed it.

Moving on, I will be discussing the machine learning method that we employed, along with the specific model and tables that we used. It's worth noting that different problems call for different machine learning approaches, and the method and model we chose were selected based on their suitability for this particular project.

Next, I will be talking about grpc, which is a technology that played a significant role in personalizing this project. It allows for efficient communication between different parts of the system, and it was crucial for achieving the desired level of customization.

Finally, I will be discussing docker-compose, which is a tool that we used to provide device simulation. This helped us to test and validate our system in a realistic environment without having to rely on physical devices. Overall, these technologies and methods were essential for the successful completion of this project and I will be providing more details on each one in the following sections.

3.1. Data Collection, Preprocessing

The dataset that I am using in my project is the mnist dataset, which is already present in the keras library. It can be easily imported into the python code and accessed directly from the website. Since it is an image processing dataset, I am using Convolutional Neural Networks (CNN) for the processing. The dataset is divided into four different parts, x_train, y_train, x_test, and y_test, which allows for easy manipulation and analysis. The total number of training data is 60000, and the total number of test data is 10000. In order to effectively utilize the training data, I am categorizing it before any processing is done. This categorization process results in a dataset that is in byte type, which increases the total number of training data to 200000.

One of the benefits of using byte type data is that it allows for processing the data piece by piece. This is something that I have been striving for from the beginning of my project, as it allows for a more thorough and detailed analysis. Furthermore, by taking the data piece by piece, it is possible to include it in the federated algorithm. This algorithm, which is known for its high efficiency, allows for dividing the data into several equal parts and distributing it among multiple nodes.

In my project, I am dividing the obtained data into three equal parts and distributing it to three different nodes that eliminate the need for three separate devices. This allows for a more efficient and streamlined process. Additionally, these data are assigned to clients for simulation purposes, which enables me to test and evaluate my results in a realistic environment.[13]

3.2. Model Training, Evaluation

3.2.1. Model Definition

In the process of defining our model, I have chosen to use a sequential model as the structure. This allows for a clear and organized way to stack layers on top of each other, as seen in Figure 3.1. After defining the structure of our model, we then proceed to train it by using the 'fit' function. I have set the number of epochs to be 10, which means that the model will iterate through the entire dataset 10 times. However, in practice, I have found that due to the stability of the mnist dataset, the model is able to reach high accuracy levels within just 4 iterations. This is likely due to the small size and simplicity of the dataset. Initially, I had started with 7 epochs, but as I continued to train the model, I found that this number could be reduced to 4 while still achieving similar results. The loss values, which measure the difference between the predicted output and the actual output, were consistently below 1 throughout the training process. You can refer to Figure 3.2 for a visual representation of these results.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 25, 25, 32)        544

 max_pooling2d (MaxPooling2D  (None, 12, 12, 32)       0
 )

 flatten (Flatten)           (None, 4608)              0

 dense (Dense)               (None, 128)               589952

 dense_1 (Dense)             (None, 10)                1290

=================================================================
Total params: 591,786
Trainable params: 591,786
Non-trainable params: 0
_________________________________________________________________
```

Figure 3.1: Model Summary.

```
Epoch 1/10
1875/1875 [==============================] - 40s 21ms/step - loss: 0.1389 - accuracy: 0.9581 - val_loss: 0.0622 - val_accuracy: 0.9792
Epoch 2/10
1875/1875 [==============================] - 37s 20ms/step - loss: 0.0485 - accuracy: 0.9853 - val_loss: 0.0533 - val_accuracy: 0.9824
Epoch 3/10
1875/1875 [==============================] - 37s 20ms/step - loss: 0.0317 - accuracy: 0.9898 - val_loss: 0.0391 - val_accuracy: 0.9864
Epoch 4/10
1875/1875 [==============================] - 36s 19ms/step - loss: 0.0214 - accuracy: 0.9933 - val_loss: 0.0424 - val_accuracy: 0.9865
<keras.callbacks.History at 0x7f6ede16c880>
```

Figure 3.2: Model Epochs.

3.2.2. Model Results

When we look at the metric values of our model, we can see that the loss value is approaching zero and the accuracy value is approaching one. This is a clear indication that our model is performing well and is able to accurately classify the data. It's important to note that the method of reading values used in the model is through tabulation. We have also included tables that show the training speed of these values in the figures below. Please refer to Figure 3.4, Figure 3.5, and Figure 3.6 for more information. Overall, this suggests that our model is effectively learning and generalizing the underlying patterns in the data, which is the ultimate goal of any machine learning model.

	loss	accuracy	val_loss	val_accuracy
0	0.138908	0.958133	0.062219	0.9792
1	0.048531	0.985283	0.053261	0.9824
2	0.031661	0.989767	0.039065	0.9864
3	0.021411	0.993283	0.042383	0.9865

Figure 3.3: Model metric results.

In the precision section, the accuracy values of the trained model were measured and tabulated. The accuracy values are similar to Figure 3.7, which shows that the model is performing well in terms of precision. The mnist dataset is known to be highly predictable and the model structure is compatible with it, which results in almost perfect prediction values. The high accuracy of correct prediction is also highlighted in another way of reading the results, which is the confusion matrix. The confusion matrix is given in Figure 3.8. The values at the coordinates where the x and y values

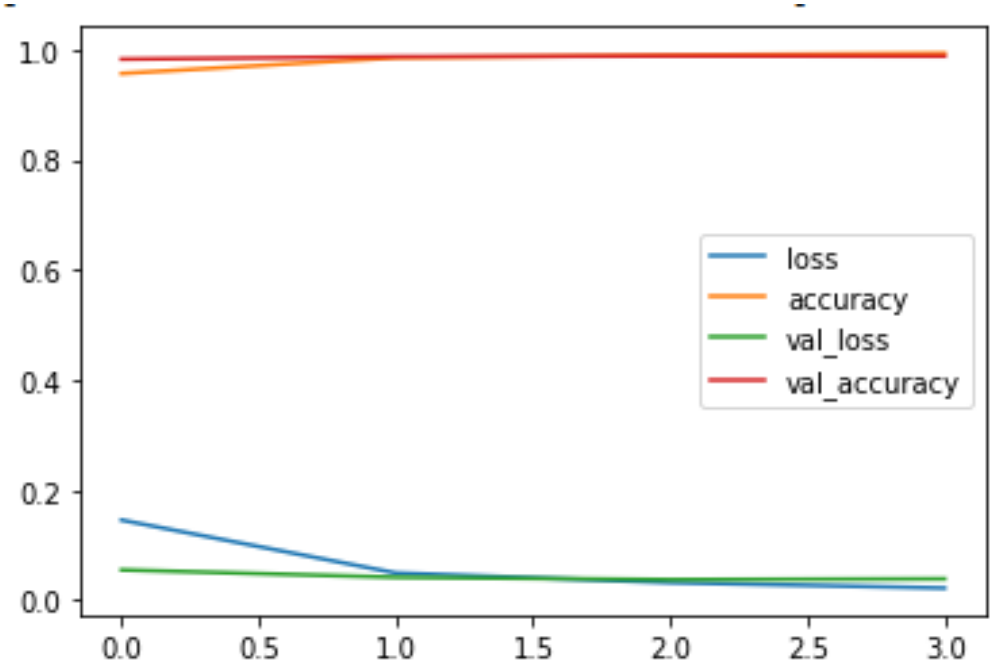

Figure 3.4: Model Loss, Accuracy, Val_Loss, Val_Accuracy.

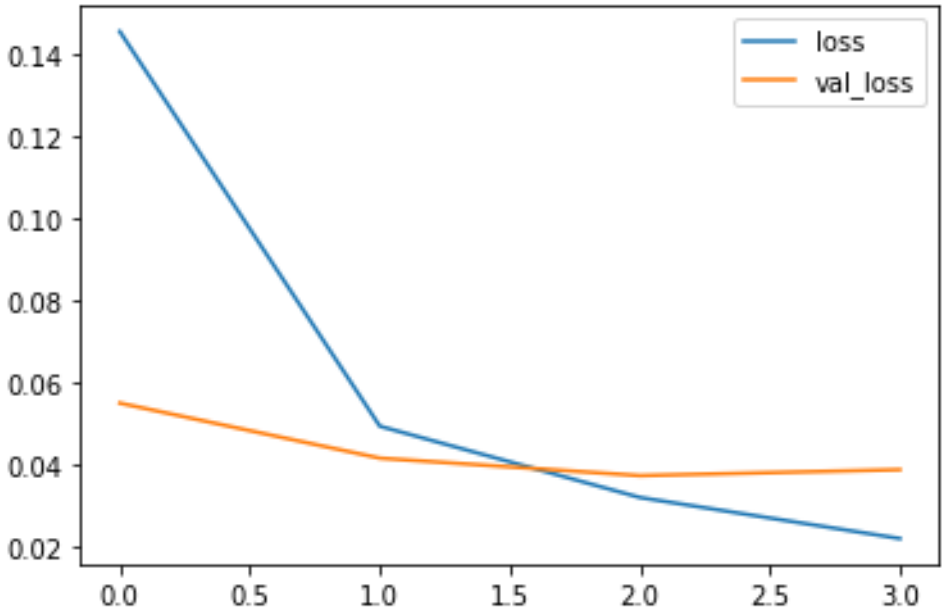

Figure 3.5: Model Loss, Val_Loss.

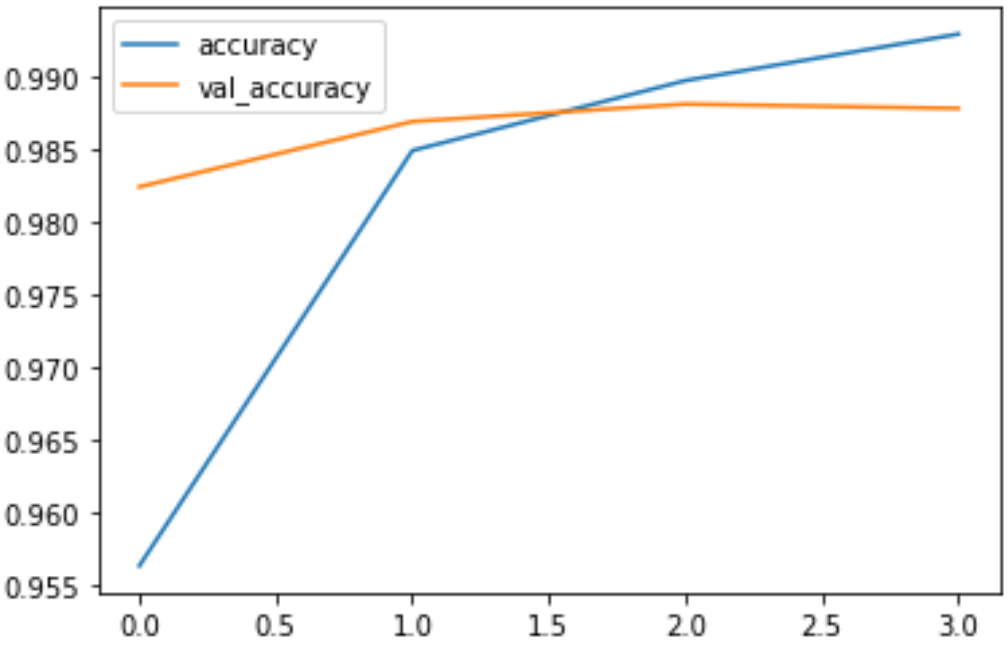

Figure 3.6: Model Accuracy, Val_Accuracy.

are the same, show the high accuracy of correct prediction numerically. This is an important metric for evaluating machine learning models as it helps to understand how well the model is able to correctly identify the classes in the dataset. Overall, it's clear that the model has been effectively trained on the mnist dataset and is able to make accurate predictions on unseen data.

```
313/313 [==============================] - 2s 6ms/step
              precision    recall  f1-score   support

           0       0.99      1.00      0.99       980
           1       0.99      1.00      0.99      1135
           2       0.98      0.99      0.98      1032
           3       0.99      0.98      0.98      1010
           4       0.98      1.00      0.99       982
           5       0.98      0.99      0.99       892
           6       1.00      0.98      0.99       958
           7       0.98      0.99      0.98      1028
           8       0.99      0.98      0.98       974
           9       0.99      0.97      0.98      1009

    accuracy                           0.99     10000
   macro avg       0.99      0.99      0.99     10000
weighted avg       0.99      0.99      0.99     10000
```

Figure 3.7: Model precision.

```
array([[ 976,    1,    0,    1,    0,    0,    2,    0,    0,    0],
       [   0, 1130,    1,    2,    0,    0,    1,    1,    0,    0],
       [   1,    0, 1021,    0,    1,    0,    0,    9,    0,    0],
       [   0,    0,    9,  986,    0,   11,    0,    2,    2,    0],
       [   0,    0,    0,    0,  978,    0,    0,    0,    0,    4],
       [   1,    0,    0,    2,    0,  886,    1,    0,    0,    2],
       [   5,    2,    2,    0,    3,    5,  939,    0,    2,    0],
       [   0,    2,    4,    2,    0,    0,    0, 1016,    1,    3],
       [   6,    0,    7,    0,    2,    0,    0,    5,  950,    4],
       [   0,    2,    1,    2,   11,    4,    0,    5,    1,  983]])
```

Figure 3.8: Model Confusion Matrix Result.

3.3. gRPC Technology

Federated learning is a distributed machine learning method that allows multiple devices or clients to train a model together without sharing their data. This is particularly useful in situations where data is sensitive or distributed across multiple locations. gRPC is an ideal technology for building the communication infrastructure for a federated learning platform. It is a high-performance, open-source framework for building remote procedure call (RPC) APIs.

gRPC allows for efficient transfer of data between servers and clients in a federated learning platform. By using a binary encoding, gRPC can transfer data faster and with less overhead than traditional text-based protocols like HTTP. This is important in a federated learning platform, as the amount of data being transferred can be quite large.

To make the process of implementing gRPC in a federated learning platform easier, the flower framework can be used. This is a set of tools and libraries that provide a simplified interface for building gRPC-based systems.

The first step in using gRPC in a federated learning platform is to write a protoc file. This file defines the API and data structures that the server and clients will use to communicate. The protoc file is written in Protocol Buffers, a language- and platform-neutral data serialization format. Once the protoc file is written, it can be compiled to generate the necessary code for the server and client.

After the code is generated, data can be easily transferred between the server and clients using gRPC's efficient binary encoding. The server can send model updates and other information to the clients, and the clients can send back their updated models and other data. This process can be repeated iteratively until the model is sufficiently trained.

In conclusion, gRPC is an ideal technology for building the communication infrastructure for a federated learning platform. It offers high-performance and efficient data transfer, which is crucial in distributed machine learning. The flower framework can be used to simplify the process of implementing gRPC in a federated learning platform, and the protoc file can be used to define the API and data structures. This allows for easy transfer of data between servers and clients, making the training of a model in a distributed setting a more efficient process.

3.4. Docker Technology

Docker is a popular technology that can be used to containerize the various components of a federated learning platform that uses gRPC. By containerizing the components, they can be easily deployed and run on any system that supports Docker. This allows for a more efficient and consistent deployment process, as the same container can be used across multiple environments.

Using Docker in conjunction with gRPC and the flower framework allows for an even more streamlined process for building and deploying a federated learning platform. The gRPC server and client code, as well as the flower framework, can be packaged into separate containers. These containers can then be easily deployed and run on any system that supports Docker. This allows for a more efficient and consistent deployment process, as the same container can be used across multiple environments.

Additionally, by containerizing the gRPC server and client code, it can be isolated from the host system, which increases security and stability. It also allows for easy scaling of the federated learning platform by spinning up multiple instances of the server and client containers as needed.

To sum up, using Docker in conjunction with gRPC and the flower framework, allows for an efficient and consistent deployment process and can increase the security and stability of the federated learning platform. It also allows for easy scaling of the system by spinning up multiple instances of the server and client containers as needed.

BIBLIOGRAPHY

[1] Kim Martineau. “What is federated learning?” (2022), [Online]. Available: https://research.ibm.com/blog/what-is-federated-learning.

[2] Ferdi DOĞAN, İbrahim TÜRKOĞLU. “Derin öğrenme modelleri ve uygulama alanlarına İlişkin bir derleme.” (2018), [Online]. Available: https://dergipark.org.tr/tr/download/article-file/738321.

[3] Matt Mazur. “A step by step backpropagation example.” (2015), [Online]. Available: https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/.

[4] Alexander Brecko, Erik Kajati, Jiri Koziorek, Iveta Zolotova. “Federated learning for edge computing: A survey.” (2022), [Online]. Available: https://www.mdpi.com/journal/applsci.

[5] Nasron Cheong. “Design a federated learning system in seven steps.” (2021), [Online]. Available: https://towardsdatascience.com/design-a-federated-learning-system-in-seven-steps-d0be641949c6.

[6] G Anthony Reina, Alexey Gruzdev, Patrick Foley, Olga Perepelkina, Mansi Sharma, Igor Davidyuk, Ilya Trushkin, Maksim Radionov, Aleksandr Mokrov, Dmitry Agapov, Jason Martin, Brandon Edwards, Micah J. Sheller, Sarthak Pati, Prakash Narayana Moorthy, Shih-han Wang, Prashant Shah, Spyridon Bakas. “Openfl: An open-source framework for federated learning.” (2021), [Online]. Available: https://arxiv.org/abs/2105.06413.

[7] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, Virginia Smith. “Federated optimization in heterogeneous networks.” (2018), [Online]. Available: https://arxiv.org/abs/1812.06127.

[8] Ivan Kholod, Evgeny Yanaki, Dmitry Fomichev, Evgeniy Shalugin, Evgenia Novikova, Evgeny Filippov, Mats Nordlund. “Open-source federated learning frameworks for iot: A comparative review and analysis.” (2021), [Online]. Available: https://www.mdpi.com/1424-8220/21/1/167.

[9] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletarì, Holger R. Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N. Galtier, Bennett A. Landman, Klaus Maier-Hein, Sébastien Ourselin, Micah Sheller, Ronald M. Summers, Andrew Trask, Daguang Xu, Maximilian Baust, M. Jorge Cardoso. “The future of digital health with federated learning.” (2020), [Online]. Available: https://www.nature.com/articles/s41746-020-00323-1.

[10] Saheed Tijani. “Federated learning: A step by step implementation in tensorflow.” (2020), [Online]. Available: https://towardsdatascience.com/federated-learning-a-step-by-step-implementation-in-tensorflow-aac568283399.

[11] Hui Jiang, Min Liu, Bo Yang, Qingxiang Liu, Jizhong Li, Xiaobing Guo. “Customized federated learning for accelerated edge computing with heterogeneous task targets.” (2020), [Online]. Available: https://doi.org/10.1016/j.comnet.2020.107569.

[12] Zonghao Lu. “A case study about different network architectures in federated machine learning.” (2020), [Online]. Available: https://uu.diva-portal.org/smash/get/diva2:1500720/FULLTEXT01.pdf.

[13] (), [Online]. Available: https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz.

CV

ELIFNUR KABALCI

elifnurkabalci387@gmail.com
https://www.linkedin.com/in/elifnurkabalci/
https://github.com/elifnurkabalci

PROFESSIONAL SUMMARY
Soft Skills :
Work in a team, coachable, strong communication skills, take responsibility and initiative, open to learning new things, ready to learning different languages, knowledge of software and hardware, ability to teach.
Tech Skills :
C, C++, Java, Arduino, Assembly, Python, SQL, C Sharp, Lisp, Unity.

CERTIFICATIONS
BUILD 15 AUGMENTED REALITY (AR) APPS WITH UNITY VUFORIA - Udemy
ARTIRILMIŞ GERÇEKLIK - 1 - Udemy
DEVOPS ÇÖZÜMLERI (JENKINS) - BTK Akademi
HYPERCASUAL TEMELLERI - BTK Akademi
GÜNCEL JAVA TEKNOLOJILERI ILE KURUMSAL UYGULAMA GELIŞTIRME ATÖLYESI - i2i-BTK Akademi
BORUSAN TEKNOLOJI OKULU - BORUSAN
TURKCELL AKADEMI GELECEĞI YAZANLAR - Turkcell
3D ZIRVE ’21 - ACC Bilkent
TURKCELL AKADEMI / YAZILI İLETIŞIM - Turkcell

APPENDICES

Technical Documentation: Detailed technical documentation, including system diagrams, code snippets, and test results, will be provided in the appendix. This will include information on the platform architecture, data transformation, model training, and performance evaluation.

Source Code: The source code for the platform will be provided in the appendix. This will include the code for the server and clients, as well as the code for the data transformation, model training, and performance evaluation.

Test Results: The test results for the platform will be provided in the appendix. This will include the results of the performance evaluation, including accuracy, data privacy and security, and the federated learning loop, fedavg.

User Manual: A user manual will be provided in the appendix. This will include instructions on how to install and configure the platform, as well as how to use the platform to collect and transform data, train models, and evaluate performance.

Additional Resources: Any additional resources, such as data sets and libraries, used in the project will be provided in the appendix.

List of references: A list of all the references used in the project will be provided in the appendix. This will include academic papers, books, and online resources that were used in the research and development of the platform.

Installation guide: A detailed guide for the installation of the platform, including the necessary dependencies and configurations will be provided in the appendix.