

Data Partitioning Effects in Federated Learning

Mirwais Ahmadzai^{1,*}, Giang Nguyen¹

¹Faculty of Informatics and Information Technologies, STU in Bratislava, Ilkovičova 2, Bratislava 84216, Slovakia

Abstract

Federated learning is a potential ML approach that promotes cooperative learning among many distributed systems while ensuring data privacy. In this study, we present a wide review of the design and evaluation of FL, with a particular focus on data partitioning. We discuss the challenges and solutions associated with FL implementation and demonstrate the design and execution of our proposed FL architecture. The main contribution of this paper is an investigation of data partitioning in FL and its impact on system performance. Using real-world public opinion data, we evaluate our proposed FL architecture and investigate performance measures such as binary accuracy, F1 score, loss, communication overhead, and data transmission between the server and clients. The experimental results provide useful information on the effective use of FL in various contexts. We underline the distinct advantages of various data partitioning algorithms based on data distribution and privacy requirements. Our findings contribute to the creation of successful FL systems that protect privacy.

Keywords

Data Partitioning, Federated Learning, Architecture, Design, Implementation, Evaluation

1. Introduction

Machine learning (ML) algorithms have achieved remarkable success in various applications, ranging from image and speech recognition to natural language processing and recommendation systems. However, these algorithms typically require large amounts of data to be collected and analyzed, which can pose privacy and security concerns. Decentralized ML approaches, such as Federated Learning (FL), have emerged as a promising solution to address these challenges [1]. FL enables multiple entities to jointly train a model without sharing their raw data. Instead, each entity develops a local model using its own data and shares only model updates with a centralized server, which combines them to produce a global model. This strategy is especially helpful in situations where data is distributed across multiple devices or organizations, and data exchange is restricted by privacy considerations.


Despite its great potential, FL is still a developing topic with a considerable information gap on the impact of data partitioning on performance, particularly in the context of data from public opinion surveys. The purpose of this research is to examine the effects of different data partitioning strategies on FL systems. Our goal is to contribute to the existing literature by examining state-of-the-art approaches, identifying their limitations and challenges, and


SQAMIA 2023: Workshop on Software Quality Analysis, Monitoring, Improvement, and Applications, September 10–13, 2023, Bratislava, Slovakia

*Corresponding author.

✉ mirwais.ahmadzai@stuba.sk (M. Ahmadzai); giang.nguyen@stuba.sk (G. Nguyen)

ORCID 0000-0001-5201-2802 (M. Ahmadzai); 0000-0002-6769-0195 (G. Nguyen)

 © 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

suggesting best practices for selecting and implementing data partitioning techniques in FL for public opinion survey data.

Although a variety of model performance evaluation metrics are discussed, such as communication efficiency, model performance, privacy, system performance, system and statistics heterogeneity, and motivitability, our experimental evaluation only focuses on model performance (accuracy and F1 score) and communication overhead using public opinion data. Furthermore, this work does not address the ethical issues of FL, which would require further investigation. An additional study is required to investigate the influence of data partitioning strategies on other areas of FL system performance in order to provide an improved understanding of their effects and potential best practices.

In this context, the remainder of this paper is structured as follows: it starts with a brief review of related work and highlights the differences and contributions of our paper in Section 2. The contribution and motivation of the research in the context of data from the public opinion survey are described in Section 2.2. The proposed design of the FL architecture is described in Section 3. Data partitioning in FL is discussed in Section 4. The performance evaluation of the FL architecture using public opinion data and presenting the metrics and techniques used for this evaluation is done in Section 5, and Section 5.1. Finally, the article concludes the work and suggests potential research directions in Section 6.

2. Related Work

Recent years have seen a considerable increase in the level of research on FL, and many studies have been conducted on its application in various domains. In this section, a systematic literature review is applied to select and highlight relevant work on its design, application, and evaluation. The review is targeted searches in reputable databases using topic keywords to ensure completeness. Table 1 summarizes the review according to the focus area of the study, the method used to determine the main findings and limitations.

The paper [2] presents a scalable production system for FL on mobile devices, with an emphasis on the difficulties of privacy, security, and communication. Although the study provides useful insights into the practical implementation of FL, it does not evaluate or compare the performance of the system with other systems. The paper [3] looks at the latest developments and problems in FL, including ways to mitigate privacy risks, without focusing on their limits. The paper [4] proposes a practical FL method that reduces communication costs and is robust to non-IID data distributions, but its limitations include experiments conducted on a limited number of data sets and model architectures, as well as a lack of consideration for privacy preservation. The paper [5] proposes an algorithm that minimizes learning loss within a given resource budget, although it has constraints such as focusing on a certain class of ML models and conducting experiments in a simulated environment. The paper [6] presents a complete study of current research in the management of non-Independent and Identically Distributed (non-IID) data on ML models in FL, but no concrete conclusions are presented.

Table 1
Summary of Related Work. Source: author’s contribution.

Study	Focus Area	Methodology	Main Findings	Limitations
[2]	FL challenges and design	The paper describes a scalable FL system for mobile devices based on TensorFlow that addresses privacy, security, and communication issues.	The paper provides insights into the design and implementation of a practical FL system that handles privacy, security, and communication problems.	The paper lacks a detailed evaluation, comparisons with other systems, and discussion of FL limitations, open challenges, and future possibilities.
[3]	FL and its recent advances, open problems, and challenges.	The study investigates FL approaches, obstacles, and open problems, such as compressed communication, decentralized optimization, and differential privacy.	The paper highlights recent developments, challenges, and privacy-protection strategies in FL research, as well as unresolved issues and concerns.	The paper’s limitations are not stated, but it highlights shortcomings and unresolved issues in methods and approaches related to FL.
[4]	FL: A Decentralized Machine Learning Approach.	FL is suggested for decentralized ML along with empirical tests with various models and data sets.	FL decreases communication costs while being resistant to unbalanced and non-IID data. Experiments show that communication rounds are reduced by 10-100 times.	The limited data set and model architectures used in experiments, as well as the lack of privacy-preserving FL, are limitations of this paper.
[5]	Algorithm for optimal edge learning.	The proposed technique optimizes local updates and global aggregation to minimize the loss within a resource budget, evaluated using distributed gradient descent.	The proposed algorithm works well. Adjusts global aggregation frequency dynamically to minimize learning loss under a particular budget.	focuses on specific models, conducts experiments in a virtual environment, and assumes that all nodes have an equal amount of resources.
[6]	FL on non-IID data: challenges and solutions.	Evaluates the influence of Non-IID data on ML models in FL and reviews techniques to address it.	The study gives a complete analysis of existing research on FL with Non-IID data and makes recommendations for future research.	There are no limitations stated, the paper also does not give new experimental data, but rather summarizes previous studies.

2.1. Challenges and Solutions in Federated Learning Implementation

Implementing FL can be difficult because it requires balancing the privacy and utility of local data with the effectiveness of the ML process. Due to its distributed nature, FL encounters a variety of challenges during training, including problems with communication, heterogeneity of data and systems, and data privacy and security. In general, it requires careful consideration when designing an FL system [7], [8].

Table 2, describes the main issues in FL architecture due to privacy requirements and data volume, leading to limited communication in FL networks. Local updating, compression approaches, decentralized training, and importance-based updating are some of the solutions suggested by researchers. These strategies are designed to maintain the balance between effective communication, convergence, and accuracy of the model. Federated networks face the challenge of system heterogeneity, which causes participants with various communication, processing, and storage capacities. To address this issue, asynchronous communication, client participation, and fault tolerance are used. Client participation selects devices based on their resources and data quality, while fault tolerance adds algorithmic redundancy or coded

computation to handle device failures.

The existence of non-IID data throughout the network causes challenges in statistical heterogeneity in FL. To solve this, the researchers propose employing multitask learning, measuring heterogeneity with measures such as local dissimilarity, and representing user preferences with personalization layers. Recent research has revealed that FL may not always provide adequate privacy guarantees during model updates and may be vulnerable to two types of attack, including poisoning attacks and inference attacks [9], [10]. Poisoning attacks can be carried out during the model’s training phase or on the data. Inference attacks can occur during model updates and expose participants’ private information to the adversary [11]. There are various privacy-preserving mechanisms, such as Secure Multiparty Computing (SMPC), Differential Privacy (DP), and Homomorphic Encryption (HE), that can be used in FL. By integrating numerous parties, the SMPC maintains security. To preserve individual privacy, DP adds noise to the data, while HE modifies the encryption parameters to protect user data.

Table 2

Federated Learning Challenges and Proposed Solutions. Source: author’s contribution.

Possible Solutions	Challenges				Citations
	Communication	Systems heterogeneity	Statistic heterogeneity	Privacy leakage	
Local updating	✓				[12]
Quantization and Compression techniques	✓				[13]
Decentralized training	✓				[14]
Importance-based updating	✓				[15]
Asynchronous communication		✓			[16]
Client participation		✓			[17]
Fault tolerance		✓			[18]
Local dissimilarities			✓		[19]
Leveraging the idea of personalization layers			✓		[20]
DP, SMPC, HE				✓	[21]

2.2. Motivation and Contribution

Federated learning has limited use in the real world despite its benefits, such as improved model accuracy and privacy preservation. These issues can be resolved and its practical adoption improved by looking into the effects of data partitioning in FL. This article’s goal is to investigate how data partitioning techniques affect FL system performance with our following contributions:

1. An examination of data partitioning techniques in FL, focusing on how they affect system performance and communication effectiveness.
2. A novel approach to choosing and putting into practice the best data partitioning strategies for certain use cases.
3. Evaluating the effectiveness of the proposed methodology in increasing model accuracy and decreasing communication overhead using data from public opinion surveys.

3. Federated Learning Architecture Design and Implementation

The hardware and software requirements for implementing the FL architecture can vary depending on the individual use case and the scale of the devices involved. However, in general, clients, servers, models, and algorithms are components of the FL architecture [3]. Hardware requirements include a collection of distributed devices (such as smartphones, laptops, and the Internet of Things (IoT) devices) with enough processing power to locally train an ML model, a server that meets certain criteria, and a reliable network connection that can interact with the central server and other participating devices. Each client (local device) has its own data set, which is used to train the ML model. The FL process is coordinated by the server, which sends model updates to the clients and aggregates their updates. The server also keeps the global model safe and secure. The model is an ML model that has been trained using the FL method. It is usually a deep neural network that is trained and decentralized across clients. The optimization technique used to train the model is called an algorithm. In terms of software requirements, they are as follows [22, 23].

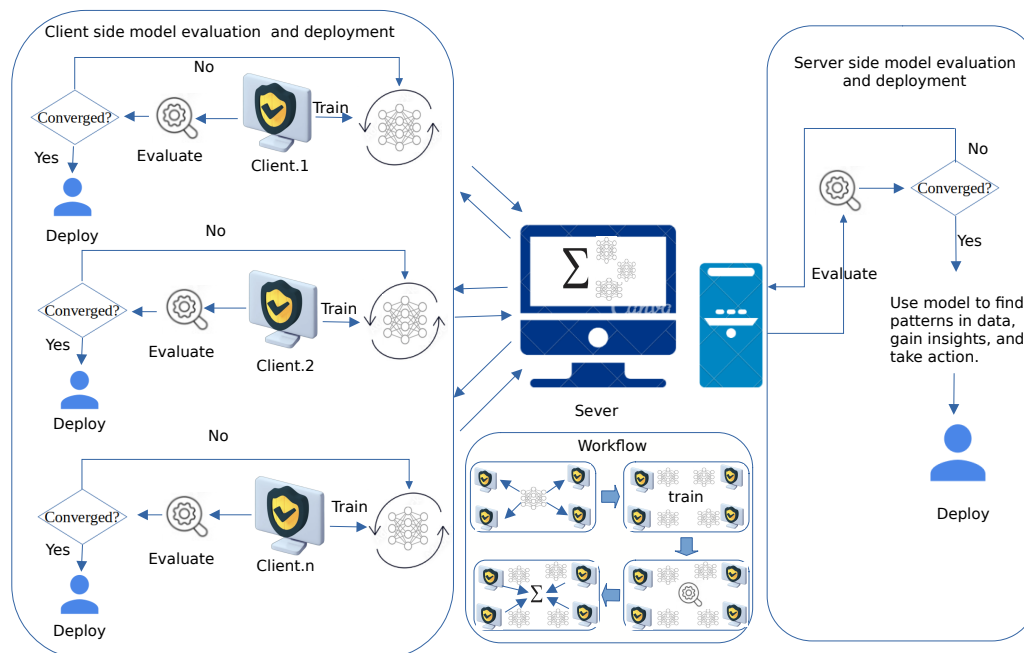


Figure 1: Federated Learning Architecture. Source: author's contribution.

ML frameworks that support the FL process such as TensorFlow or PyTorch. A central server software that manages the process, including model aggregation and device synchronization, which can be built with technologies like Apache Kafka, RabbitMQ, and Redis. A client-side software library that allows devices to participate in FL and communicate securely with the central server. Protocols for secure and encrypted communication to protect the privacy of

data on participating devices. FL architecture design workflow for the public opinion survey example is depicted in Fig. 1, the central server distributes the initial model parameters to all clients. Clients train their local models with initial parameters and exchange the results with the central server. The central server aggregates the local models and distributes the global model to the clients.

Depending on the particular use case and privacy restrictions, many methodologies can be utilized to evaluate and deploy the model. Clients do a local evaluation of the model and send the results to the central server for aggregation. The performance of the global model is then evaluated in general by the server. As an alternative, the server validation data set can be used for evaluation. When privacy is an issue, clients can also receive the global model that has been aggregated for local predictions. The global model, on the other hand, can be hosted by the server and made available as a service to clients, who can then send their data for predictions. Data privacy, resource limitations, and the complexity of the model management process are a few examples of considerations that impact the decision to choose client-side or server-side evaluation and deployment [24].

4. Data Partitioning in Federated Learning

FL partitioning distributes data across multiple parties who collaborate to increase the usefulness of their combined data. This method overcomes the limitations of domain-specific data and makes it easier for clients with various interests to work together. Based on data flow between parties, FL data partitioning can include transfer learning, vertical partitioning, and horizontal partitioning (Fig. 2). It takes careful preparation to bring together the interested parties and partition the data in a way that produces an FL environment, as proper data partitioning is crucial for the FL process [25].

Horizontal FL (HFL) combines data from entities with similar features but different samples. In the HFL example, two research organizations (regions A and B) collect data from a public opinion survey but are only able to share limited information because of privacy concerns. The purpose is to develop an ML model that uses parameters such as age, gender, and service type to predict how satisfied clients are with government services. Each organization first trains a local model using its own data, then shares model updates with a central server to create a global model, and then deploys the aggregated model to clients.

Vertical FL (VFL) combines data from entities with the same sample IDs but distinct features. VFL allows different respondents to share demographic data while maintaining the privacy of survey responses. Each client trains a local model using local data and survey results and then shares model updates with the server, which constructs a global model capable of generating predictions across all attributes.

Transfer FL (TFL) involves the use of a previously trained model on a similar task to improve the performance of a new model on a new task. TFL can be applied for both VFL and HFL. TFL involves training a model on one set of data and then fine-tuning it on another. When one region has more data than the other, the model is trained on the bigger data set first and then fine-tuned on the smaller data set.

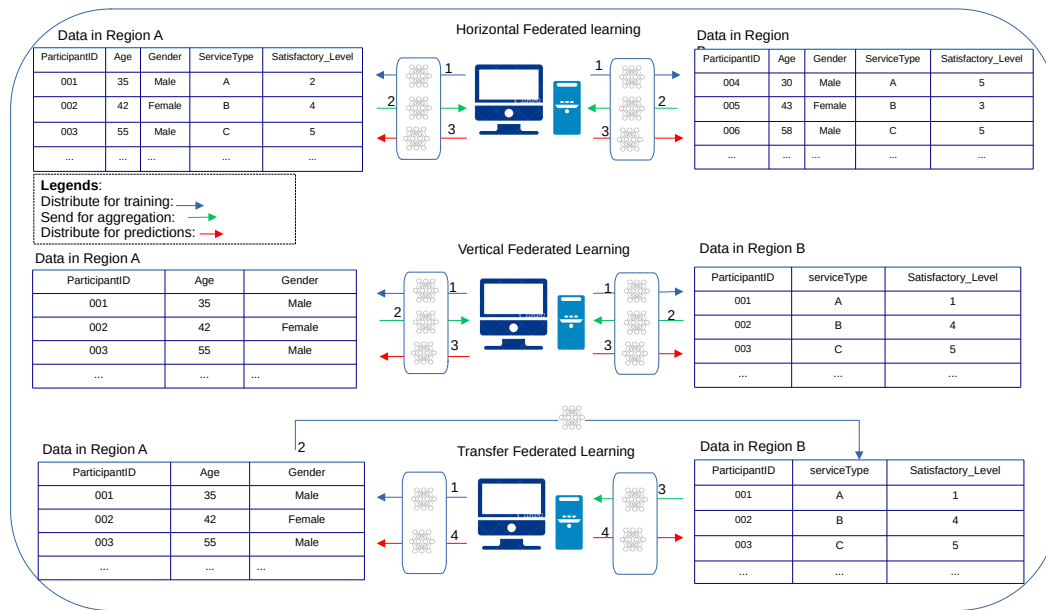


Figure 2: Data Partitioning in Federated Learning. Source: author's contribution.

5. Performance Evaluation of Data Partitioning in Federated Learning Architectures

Evaluation of FL architecture is a crucial component because it allows us to measure the efficiency of the model and make additional improvements. Evaluation metrics, methods, and best practices for FL architecture will be discussed in this part. The metrics shown in Table 3 are frequently used to assess the effectiveness and efficiency of the FL approach. These metrics include communication costs, model performance, system scalability and performance, attack rates, computation and energy costs, convergence rates, statistical and system heterogeneity, client motivation, and data and device security [26, 27].

5.1. Experimental Results and Discussion

The performance of HFL, VFL, and TFL was compared using data collected by the Asia Foundation. It was a public opinion survey to obtain civilian thoughts and impressions on a variety of Afghanistan-related issues. The data set includes survey questions and responses related to security, governance, and country development.

Due to the FL nature, clients send their local model updates to the server, which aggregate these updates to improve the global model. Communication is a substantial bottleneck in the FL process, especially if the network bandwidth is limited or a large number of clients are participating. In our experiments, the quantization technique was investigated as a substantial solution to this problem for three FL architectures (HFL, VFL, and TFL). The findings showed

Table 3
Summary of Evaluation Metrics. Source: author’s contribution.

Evaluate	Metric	Interpretation
communication efficiency	Communication cost, dropout ratio, system running time	FL communication efficiency can be increased by lowering expenses, dropout rates, and system running time.
Model Performance	AUC-ROC, F1-score, precision, recall, and perplexity	These metrics are important to evaluate the performance of the FL model.
System Performance	attack rate, costs, dropout, convergence rate, running time.	These metrics are used in FL to evaluate system performance and suitability for a specific task.
Statistical heterogeneity	Model and Equation Proof	In FL, statistical heterogeneity measures the variation in model and equation proof and the distribution of data between different clients.
System heterogeneity	device dropout ratio due to limited resources	Measures device variances that affect performance. Hardware resources, dropout ratio, communication and processing costs, and system running time are a few examples.
Client moti-vatability	incentive rate	In FL, client motivation is the incentive for clients to participate in providing data.
System scalability	Communication Cost and Running Time	determining the system’s ability to handle an increasing number of clients and data while maintaining high performance.
Data privacy	SMPC, HE, DP	FL requires data privacy mechanisms that must be evaluated for efficiency, performance, scalability, and sensitive data protection.

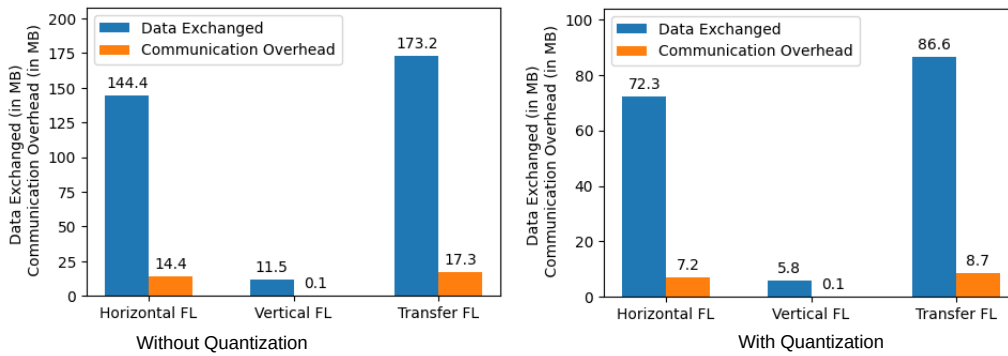


Figure 3: Data Exchanged and Communication Overhead for Different Data Partitioning in FL. Source: author’s contribution.

in Fig. 3 that the transmission overhead without quantization was $HFL:14.44 MB$, $VFL:0.12 MB$, $TFL:17.32 MB$, while, the communication overhead for the three techniques decreased significantly after applying quantization to model updates as $HFL:7.22$, $VFL:0.06$, $TFL:8.70$.

These findings indicate the efficiency of quantification in reducing communication overhead in FL systems. Reduced overhead can result in faster convergence, improved scalability, and lower communication costs. However, it is critical to assess the impact of quantization on the accuracy and loss metrics of the model. In this research, we also conducted tests for binary classification models in horizontal, vertical, and transfer FL setups. The results showed that

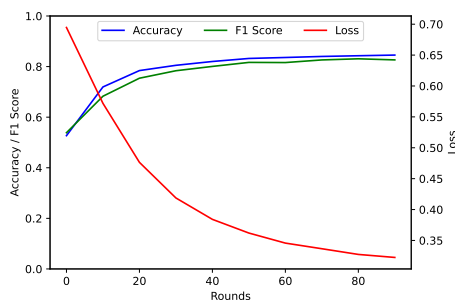
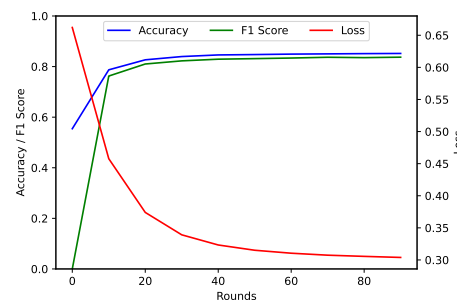
Table 4

Summary of the Experimental Methodology and Analysis. Source: author's contribution.

Item	Description
Data set	A public opinion survey data set
Framework	TensorFlow Federated (TFF)
Models Used	2 regional client models, 1 server model
Optimization Algorithm	Stochastic Gradient Descent (SGD)
Server Model	Feed-forward neural network with sigmoid activation function
Loss Function	Binary cross-entropy
Data Partitioning Methods	HFL, VFL, and TFL
Learning Model	Federated Averaging (FedAvg)
Hyperparameters	Learning rate: 0.01, Local epochs: 5, Batch size: 1
Evaluation Metrics	Binary Accuracy, F1 score, Loss, Data Exchange, Communication overhead
Experimental Procedure	<ol style="list-style-type: none"> 1. Preprocess the data set 2. Apply data partitioning methods 3. Train the model using FedAvg 4. Evaluate the model
Analysis	Compared the performance of partitioning methods
Figures 3, 4, and 5 Results	Obtained from experiments

the use of quantization maintained acceptable levels of accuracy and loss, making it a feasible solution to reduce communication overhead in FL systems.

Table 4, summarizes the experimental methodology and analysis performed in our study. It includes key elements such as the description of the data set, the comparison of different data partitioning methods, the specific federated learning model used, the hyperparameters chosen for training, the evaluation metrics used, and an outline of the experimental procedure, details about the analysis process, and the source of the results shown in Fig. 3, Fig. 4, Fig. 5, and Fig. 6. Mentioned and presented figures represent the author's contribution in this research.

**Figure 4:** HFL Model Performance**Figure 5:** VFL Model Performance.

The performance of three different FL approaches for HFL, VFL, and TFL is studied. The performance of each approach is evaluated using three metrics: test loss, accuracy, and F1 score. The findings of the HFL experiment are indicated in Fig. 4.

The initial loss of the HFL test is 0.69, the test accuracy is 0.53, and the F1 score is 0.54. Model performance improved consistently throughout 90 rounds, with the test loss dropped to 0.32, the accuracy increased to 0.85, and the F1 score increased to 0.83. As shown in Fig. 5, the first test loss for vertical FL is 0.66, the accuracy is 0.55, and the F1 score is 0.0. Over 90 rounds, the model improved in all metrics, with the test loss decreased to 0.30, the accuracy increased to 0.85, and the F1 score increased to 0.84.

Finally, for TFL in Fig. 6, the initial test loss in the initial data set is 0.32, the accuracy is 0.8461, and the F1 score is 0.8292. The model improved during 90 rounds, with the test loss dropped to 0.30, and the accuracy and F1 score slightly increased to 0.8480, 0.8329 respectively.

In summary, during 90 rounds, the three FL approaches showed a continuous improvement in performance in all evaluation metrics. The findings show that FL can be efficiently applied to a variety of situations, each strategy providing unique benefits based on specific data distribution and privacy needs.

6. Conclusion

The paper provides a thorough investigation of the architecture of FL, with a particular emphasis on data partitioning. The importance of FL has been emphasized and the problems and limitations of existing FL techniques have been studied. The design concepts and factors necessary to establish an FL architecture have also been investigated. FL architectures have been evaluated using metrics and approaches related to data partitioning strategies. The implementation and evaluation of the FL architecture was carried out using various data partitioning architectures and the results were thoroughly explained. By evaluating the FL system using new measures, future development of more efficient, effective, and privacy-preserving FL systems can be helped. These measures should address statistical and system heterogeneity, system performance, client motivation, system scalability, and data privacy, in particular. Taking these factors into account, we can improve our understanding of FL, leading to the development of more efficient, effective, and secure FL learning systems.

Acknowledgments

This publication has been written thanks to the support of the Operational Programme Integrated Infrastructure for the project: International Center of Excellence for Research on Intelligent and Secure Information and Communication Technologies and Systems – Phase II (ITMS code: 313021W404), co-funded by the European Regional Development Fund. It is also supported by

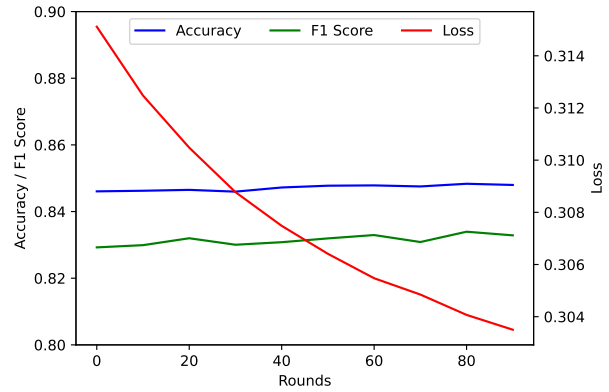


Figure 6: TFL Model Performance. Source: author's contribution.

the Operational Program Integrated Infrastructure for the project: National infrastructure for supporting technology transfer in Slovakia II – NITT SK II, co-funded by the European Regional Development Fund, and the AI4EOSC project under grant number 101058593.

References

- [1] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, D. Ramage, Federated learning for mobile keyboard prediction, 2019. [arXiv:1811.03604](https://arxiv.org/abs/1811.03604).
- [2] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, H. B. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, J. Rosenthaler, Towards federated learning at scale: System design, 2019. [arXiv:1902.01046](https://arxiv.org/abs/1902.01046).
- [3] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al., Advances and open problems in federated learning, *Foundations and Trends® in Machine Learning* 14 (2021) 1–210. doi:10.1561/22000000083.
- [4] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: *Artificial intelligence and statistics*, PMLR, 2017, pp. 1273–1282. URL: <http://proceedings.mlr.press/v54/mcmahan17a/mcmahan17a.pdf>.
- [5] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, K. Chan, Adaptive federated learning in resource constrained edge computing systems, *IEEE journal on selected areas in communications* 37 (2019) 1205–1221. doi:10.1109/JSAC.2019.2904348.
- [6] H. Zhu, J. Xu, S. Liu, Y. Jin, Federated learning on non-iid data: A survey, *Neurocomputing* 465 (2021) 371–390. doi:10.1016/j.neucom.2021.07.098.
- [7] Z. Zheng, Y. Zhou, Y. Sun, Z. Wang, B. Liu, K. Li, Applications of federated learning in smart cities: recent advances, taxonomy, and open challenges, *Connection Science* 34 (2022) 1–28. doi:10.1080/09540091.2021.1936455.
- [8] P. M. Mammen, Federated learning: Opportunities and challenges, 2021. [arXiv:2101.05428](https://arxiv.org/abs/2101.05428).
- [9] A. Shafahi, W. R. Huang, M. Najibi, O. Suciú, C. Studer, T. Dumitras, T. Goldstein, Poison frogs! targeted clean-label poisoning attacks on neural networks, *Advances in neural information processing systems* 31 (2018). URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/22722a343513ed45f14905eb07621686-Paper.pdf.
- [10] A. N. Bhagoji, S. Chakraborty, P. Mittal, S. Calo, Analyzing federated learning through an adversarial lens, in: *International Conference on Machine Learning*, PMLR, 2019, pp. 634–643. URL: <http://proceedings.mlr.press/v97/bhagoji19a/bhagoji19a.pdf>.
- [11] L. Melis, C. Song, E. De Cristofaro, V. Shmatikov, Exploiting unintended feature leakage in collaborative learning, in: *2019 IEEE symposium on security and privacy (SP)*, IEEE, 2019, pp. 691–706. doi:10.1109/SP.2019.00029.
- [12] S. Zhang, A. E. Choromanska, Y. LeCun, Deep learning with elastic averaging sgd, *Advances in neural information processing systems* 28 (2015). URL: https://proceedings.neurips.cc/paper_files/paper/2015/file/d18f655c3fce66ca401d5f38b48c89af-Paper.pdf.

- [13] S. Han, H. Mao, W. J. Dally, Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding, 2016. [arXiv:1510.00149](https://arxiv.org/abs/1510.00149).
- [14] T. Lin, S. U. Stich, K. K. Patel, M. Jaggi, Don't use large mini-batches, use local sgd, 2020. [arXiv:1808.07217](https://arxiv.org/abs/1808.07217).
- [15] Z. Tao, Q. Li, esgd: Commutation efficient distributed deep learning on the edge, HotEdge (2018) 6. URL: https://proceedings.neurips.cc/paper_files/paper/2010/file/abea47ba24142ed16b7d8fbf2c740e0d-Paper.pdf.
- [16] M. Zinkevich, M. Weimer, L. Li, A. Smola, Parallelized stochastic gradient descent, Advances in neural information processing systems 23 (2010). URL: https://proceedings.neurips.cc/paper_files/paper/2010/file/abea47ba24142ed16b7d8fbf2c740e0d-Paper.pdf.
- [17] T. Nishio, R. Yonetani, Client selection for federated learning with heterogeneous resources in mobile edge, in: ICC 2019-2019 IEEE international conference on communications (ICC), IEEE, 2019, pp. 1–7. doi:10.1109/ICC.2019.8761315.
- [18] V. Smith, C.-K. Chiang, M. Sanjabi, A. S. Talwalkar, Federated multi-task learning, Advances in neural information processing systems 30 (2017). URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/6211080fa89981f66b1a0c9d55c61d0f-Paper.pdf.
- [19] I. I. Eliazar, I. M. Sokolov, Measuring statistical heterogeneity: The pietra index, Physica A: Statistical Mechanics and its Applications 389 (2010) 117–125. doi:10.1016/j.physa.2009.08.006.
- [20] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, S. Choudhary, Federated learning with personalization layers, 2019. [arXiv:1912.00818](https://arxiv.org/abs/1912.00818).
- [21] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, K. Seth, Practical secure aggregation for privacy-preserving machine learning, in: proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 1175–1191. doi:10.1145/3133956.3133982.
- [22] M. R. Behera, R. Otter, S. Shetty, et al., Federated learning using distributed messaging with entitlements for anonymous computation and secure delivery of model (2020). URL: <https://www.academia.edu/download/70178855/25661363.pdf>.
- [23] J. Á. Morell, E. Alba, Dynamic and adaptive fault-tolerant asynchronous federated learning using volunteer edge devices, Future Generation Computer Systems 133 (2022) 53–67. doi:10.1016/j.future.2022.02.024.
- [24] T. Yang, G. Andrew, H. Eichner, H. Sun, W. Li, N. Kong, D. Ramage, F. Beau-fays, Applied federated learning: Improving google keyboard query suggestions, 2018. [arXiv:1812.02903](https://arxiv.org/abs/1812.02903).
- [25] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, G. Srivastava, A survey on security and privacy of federated learning, Future Generation Computer Systems 115 (2021) 619–640. doi:10.1016/j.future.2020.10.007.
- [26] G. Xu, H. Li, S. Liu, K. Yang, X. Lin, Verifynet: Secure and verifiable federated learning, IEEE Transactions on Information Forensics and Security 15 (2019) 911–926. doi:10.1109/TIFS.2019.2929409.
- [27] S. K. Lo, Q. Lu, C. Wang, H.-Y. Paik, L. Zhu, A systematic literature review on federated machine learning: From a software engineering perspective, ACM Computing Surveys (CSUR) 54 (2021) 1–39. doi:10.1145/3450288.