Empirical Analysis of Federated Learning Algorithms: A Federated Research Infrastructure Use Case

Harshit Gupta[1], Abhishek Verma[1], O. P. Vyas[1], Marco Garofalo[2,3], Giuseppe Tricomi[2,3], Francesco Longo[2,3], Giovanni Merlino[2,3] and Antonio Puliafito[2,3]

[1]*IIIT Allahabad, India*

[2]*University of Messina, Italy*

[3]*CINI, Italy*

Keywords: Federated Learning, Federated Research Infrastructure, FedAvg, FedProx, Stragglers, Statistical Heterogeneity.

Abstract: Research Infrastructures provide resources and services for communities of researchers at large to conduct their experiments and foster innovation. Moreover, these can be used beyond research, e.g., for education or public service. The SLICES consortium is chartered to provide a fully programmable, distributed, virtualized, remotely accessible, European-wide, federated research infrastructure, providing advanced computing, storage, and networking capabilities, including interconnection by dedicated high-speed links. It will support large-scale, experimental research across various scientific domains. Data processing, in general, and especially Machine Learning, are of great interest to the potential audience of SLICES. According to these premises, this work aims to exploit such a peculiar Research Infrastructure and its Cloud-oriented development and deployment facilities to investigate Federated Learning (FL) approaches; in particular, here we evaluate the performance of two FL aggregation algorithms, i.e., FedAvg and FedProx, in settings, characterized by system heterogeneity, and statistical heterogeneity, that represent plausible, and possibly common, scenarios in forthcoming facilities, such as those mentioned above, community-oriented, shared Research Infrastructures. We have observed that the FedProx algorithm outperforms the FedAvg algorithm in such settings.

1 INTRODUCTION

Federated Learning represents a fundamental paradigm in Deep Learning models and a peculiar deployment model across Research Infrastructures. The Scientific Large Scale Infrastructure for Computing/Communication Experimental Studies consortium, commonly indicated as SLICES, intends to provide research communities with a rich environment to conduct experiments and foster innovation, a fully programmable and virtualized, distributed and remotely accessible one, providing advanced computing, storage and networking capabilities, including inter-site connectivity by means of dedicated high-speed links, i.e., granted by consortium-level agreements with GÉANT, which is a collaboration among European National Research and Education Networks (NRENs). Under GÉANT, NRENs together deliver an information ecosystem of infrastructure and services to advance research, education, and innovation on a global scale. SLICES aims for a common research infrastructure that combines diverse technologies and services with geographically distributed research ambitions (SLICES-RI, 2021b).

Machine Learning is growing and becoming essential as more and more researchers and industries adopt it, and Federated Learning is a groundbreaking approach to train models as near as possible to individual user data (Li et al., 2019). Federated Learning enhances traditional machine learning approaches with the additional (built-in) property of protecting (training) data in terms of privacy. This strategy has been implemented by several incumbent tech firms, including Google, the inventor of federated learning (Dhada et al., 2020), and Apple, a significant competitor in the field with its virtual assistant technology.

Figure 1 represents how FL may work as a typical collaboration pattern for similar research activities, which can be accomplished remotely, without sharing critical data, over the research infrastructure.

Gupta, H., Verma, A., Vyas, O., Garofalo, M., Tricomi, G., Longo, F., Merlino, G. and Puliafito, A.
Empirical Analysis of Federated Learning Algorithms: A Federated Research Infrastructure Use Case.
DOI: 10.5220/0012037600003488
In *Proceedings of the 13th International Conference on Cloud Computing and Services Science (CLOSER 2023)*, pages 324-331
ISBN: 978-989-758-650-7; ISSN: 2184-5042

In this diagram, the layer at the top depicts Federated Learning, comprising geographically distributed users at the lower layer via a network backbone as the middle layer.

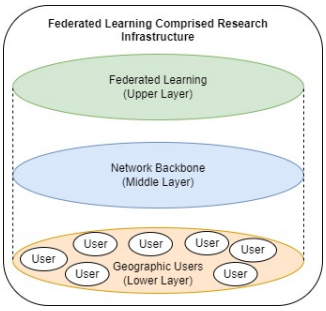

Figure 1: Federated Learning Comprised Research Infrastructure.

The research activities being performed at geographically disconnected areas for specific purposes may be brought together, and the (e.g., training) performance or (e.g., inference) accuracy achieved may be higher with the backing of Federated Learning methodologies within a research infrastructure.

As FL applications are expanding, getting into the area of Healthcare by protecting patient's medical data, thus keeping confidentially(Gope et al., 2021), as well as the Insurance sector to predict the risk associated with customers(Gupta et al., 2022), or, e.g., prediction of keyboard(Hard et al., 2018) input, it now represents a quickly growing research area. A research infrastructure with federated learning as a service offered to potential users gives a common thread to researchers who benefit from data privacy, performance enhancement, and reduced communication overhead.

Counterbalancing its intrinsic capabilities, FL suffers from the issue of heterogeneity in terms of the device (system heterogeneity) and data (statistical heterogeneity). In terms of the former, in general FL may engage nodes with uneven computation and/or storage capabilities to participate in the training process (De Vita and Bruneo, 2020). Also, the distribution of such private data among all the participating clients can be Independent and Identically Distributed (IID) or Non-Independent and Identically Distributed (NIID) (Long et al. 2021). Figure 2 provides an example of data distribution in IID and NIID environments. As can be observed in IID data, the data among the two clients are similar, while in NIID data, both clients have different data classes. The NIID data produces the issue of statistical heterogeneity(Li et al., 2018).

In an attempt to handle heterogeneity and tackle high communication costs, differential privacy (Wei et al., 2020) for security, optimization methods that allow for local updating and low participation are a popular approach for federated learning(Gad, 2020). In particular, FedAvg (McMahan et al., 2017) is an iterative method that has emerged as the de facto optimization method in federated settings.

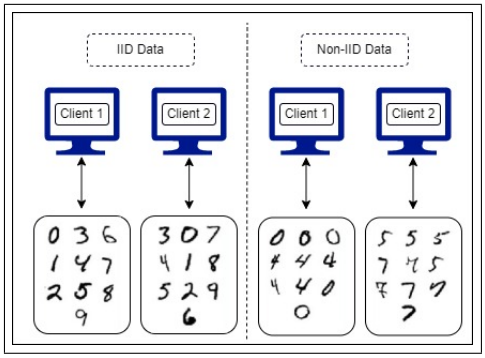

Figure 2: Distribution of Data in IID and NIID Environment.

The FedAvg algorithm performs fixed E epochs of Stochastic Gradient Descent (SGD) on K devices, where E is a small constant and K is a small fraction of the total devices in the network. In the presence of stragglers, i.e., devices which perform slowly, the FedAvg drops them if they are unable to perform E local epochs in due time. On the other hand, with dissimilar (heterogeneous) local objective F_k, a large number of local epochs may lead each device towards the optima of its local objective as opposed to the global objective, potentially causing the method to diverge and thus lacking convergence guarantees to characterize its behaviour.

It has been seen that due to the presence of such heterogeneity, the FL performance degrades while working with the FedAVG algorithm. To minimize the effect of system and statistical heterogeneity in the FL setup, authors have proposed a FedProx algorithm(Li et al., 2018) which is an optimized form of the FedAvg algorithm with the additional capacity to deal with stragglers and NIID data.

As FedAvg drops the devices which perform partial model updates, i.e., unable to perform fixed E epochs, it only considers the devices which perform complete E epochs. But FedProx does not drop out such slower devices, instead taking partial model updates from such devices as well.

Moreover, to limit the impact of varying local updates, a proximal term is added to the local subproblem. In particular, instead of just minimizing the local function $F_k(.)$, device k uses its local solver of choice to approximately minimize the following objective h_k :

$$\min_{w} h_k\left(w; w^t\right) = F_k(w) + \frac{\mu}{2}\left\|w - w^t\right\|^2 \quad (1)$$

where $\frac{\mu}{2}\|w - w^t\|^2$ is the proximal term while w and w^t are weight associated with local and global model respectively. μ is a hyperparameter that controls the strength of the proximal term.

Adding a proximal term to the objective helps to improve the stability of the method and improve the overall accuracy of federated learning in heterogeneous networks, significantly improving the absolute testing accuracy on average in highly heterogeneous settings.

Hence the work focuses on evaluating the performance of the FL algorithms, i.e. FedAvg and FedProx, in different heterogeneous environments with different datasets and models. Empirically we demonstrated that the performance of FedProx is better than FedAvg in the case of NIID data. This result is obtained through the exploitation of SLICES Research Infrastructure to provide some insight on how exploiting it to support the users in organizing and planning their experiments.

The paper is organized as follows: Section II gives us an overview of the current literature about FL, Section III describes the methodology and simulation, Section IV presents the results and their analysis, and, at last, Section V outlines our conclusions and planned future scope of the work.

2 LITERATURE SURVEY

Federated Learning (FL) is a machine learning setting in which the user's critical data is kept locally to the user where a local model is trained. In contrast to other ways to train deep learning models in which user input is pooled and processed at a centralized site, as described in Figure 3, the job of the central server in federated learning is to aggregate training done on the participating devices with their local data. In FL, the model updates are sent in rounds in which devices train the model using a local approach such as Stochastic Gradient Descent, each device sending the (current, locally-trained) model back to the server. Then the server aggregates the model together. The model can then be distributed again. Figure 3 represents the process flow in FL architecture.

In terms of privacy, federated learning has evident advantages. It solves the data privacy problems raised by previous machine learning systems by allowing data to stay decentralized, making it more difficult for attackers to access information from any user. This is because the data never leaves the devices, forcing attackers to circumvent device security rather than seeking to steal data, while it is more vulnerable in transit(Nishio and Yonetani, 2019). Despite this advantage, there are numerous difficulties and opportunities for advancement in federated learning, which we discuss further in later sections.

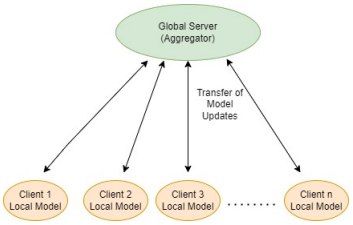

Figure 3: Federated Learning Architecture.

In (Li et al., 2018), the authors have proposed the FedProx algorithm by adding a proximal term to the FedAvg algorithm to deal with statistical and system heterogeneity. Statistical heterogeneity is the main problem and challenge in federated learning, as it has been observed that its behaviour shows a significant change when data distribution among the participating devices is NIID. If the data points of the individual devices diverge from their point of source, then the performance of the federated learning slightly decreases. It also increases model complexity.

Li et al. (Pan and Yang, 2010) mention some solutions for training in heterogeneous environments, such as local learning and aggregating the local updates from the local devices sent to the centralized server. This should continue until we hit some good performance metrics.

The SLICES research infrastructure focuses on the Internet of Things and Internet of Services, cloud/edge/fog computing, artificial intelligence, and many more. (SLICES-SC, 2022) along with distributed systems (SLICES-RI, 2021a) research, which mandates the need for privacy-preserving artificial intelligence methodologies in order to comply with the EU General Data Protection Regulation (GDPR, 2022) directives. This brings forth the role of Federated Learning in the context of SLICES, i.e., in a privacy-preserving, heterogeneous environment (e.g., in SLICES, data and/or code can be made inaccessible to other research infrastructure users, if so desired, e.g., due to some entity/company policy or other specific requirements.

The natural heterogeneity of sites participating, i.e., federated to assemble, the whole, EU-wide SLICES research infrastructure, and privacy-preserving facilities, lends itself naturally to experimentation and validation of Federated Learning approaches.

The studies performed on the heterogeneity in FL have captured various factors that negatively influence the global model's performance. The computation capability of participating devices makes the entire system slow and degrades global performance. At the same time, the NIID data among the participating clients gives statistical heterogeneity to the FL setup, which restricts the system from convergence by degrading the overall performance. In FedProx, a proximal term was found to be an essential step in boosting the performance of the existing model while alleviating the issue of stragglers compared to other state-of-the-art architectures. One of the most significant benefits of FedProx is that it guarantees convergence. Evidence exists that FedAvg is an excellent algorithm for training the data over local devices, but the convergence problem cannot decipher the features entirely. An additional proximal term would help in boosting the accuracy of models in many existing works.

3 METHODOLOGY & SIMULATION

This work conducted an empirical analysis of the FedAvg and the FedProx algorithms in different IID and NIID data environments by leveraging the SLICES research infrastructure, at the same time, publishing analyzed methods to the catalogue for other researchers, possibly to consume.

3.1 Datasets and Models

The simulation is performed on different datasets with their IID and NIID properties. The MNIST with IID and NIID data and CIFAR10 dataset with IID and NIID data are used in the proposed work.

In the case of IID data, the MNIST dataset contains 70000 samples of handwritten digits, with 60000 samples for training purposes and 10000 samples for testing purposes. With NIID data, the data are distributed differently, as shown in Figure 4. In this case, each client has digits that are not matched to other clients, making it independent non-identical data. On the other hand, the CIFAR10 dataset contains 60000 samples with 50000 images for training and 10000 for testing. The images(Dhada et al., 2020) are obtained and preprocessed with operations including segmentation, cropping and other processes before feeding it to the classification model.

In preprocessing the images, the 28x28 MNIST images and 32x32 CIFAR10 images are flattened into matrices. Before that, its dimension is changed to 3D, i.e., 28x28 and one is for the grayscale channel in the case of MNIST, and 32x32 with 3 RGB channels for CIFAR10.

The training of the models is done in a federated approach, where every client keeps its local model, which synchronizes with the global model available in the cloud using the model updates. In this work, two models, i.e. deep Convolutional Neural Network (CNN) (Albawi et al., 2017) and Multilayer Perceptron (MLP) (Taud and Mas, 2018) models, are used for training in different environments with different data distribution and presence of stragglers.

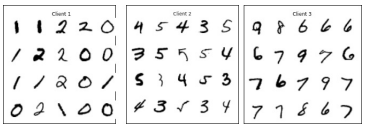

Figure 4: Clients with NIID.

3.2 Simulation Methodology

To analyze the FedAvg & FedProx algorithms, the simulation is performed with 0%, 50% and 90% stragglers and proximal term value $\mu = 0.3$. The simulation consists of 100 participating clients, a 0.1 learning rate and maximum 80 numbers of communication rounds between clients and servers. The following combinations of models and datasets are used in the simulation for 0%, 50% and 90% stragglers and proximal term value & $\mu = 0.3$.

1. FedAvg with CNN model on IID & NIID data properties of MNIST and CIFAR10 datasets.
2. FedAvg with MLP model on IID & NIID data properties of MNIST and CIFAR10 dataset.
3. FedProx with CNN model on IID & NIID data properties of MNIST and CIFAR10 dataset.
4. FedProx with MLP model on IID & NIID data properties of MNIST and CIFAR10 dataset.

3.3 Experiments Setup

The simulation is realized on the platform ***"SoBigData"***[1] (Giannotti et al. 2018)(Cresci et al., 2019), one of the gateway portals towards the federated SLICES research infrastructure, as well as the namesake of the Italian (country-wide) site of SLICES, by configuring the experiments via a method published and made available on the infrastructure. The published method performs the simulation requested according to the parameters used during the experiment execution and produces the aimed results.

[1]SoBigData e-infrastructure portal: https://sobigdata.d4science.org

The research infrastructure, to cope with this goal, provides two valuable tools for tests preparation:

1. The *Method Development* page (see Figure 5) is a dedicated environment based on JupyterHub, where through a notebook or console, the developer can write, develop, and interactively test his code in a sandbox provided with useful and ready-to-use libraries;
2. The pair of functions: *Method Importer* and *Method Engine*. Respectively, the former is used to upload and publish a method, and the latter is used to run the experiments according to the simulation parameters.

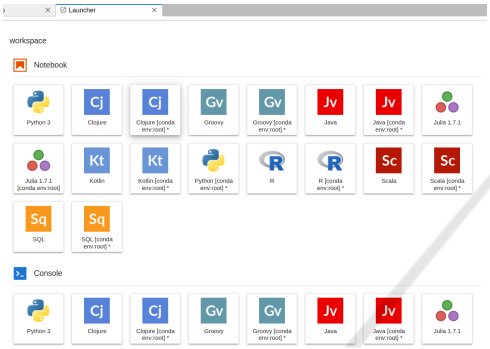

Figure 5: SoBigData platform: Method Development Launcher.

After the development phase, the method just implemented gets published (either as a black box, or the code shared), when deemed worthwhile, and its sharing is consistent with policies to foster code reuse across different experiments. The *Method Importer* generates a project for each method published and needs the definition of several elements such as input/output, code interpreter, general information, and so on. Figure 6 shows a view of the *Comparison_FedAVG_Fedprox* method. Finally, the experiments can be performed by launching the method via the opportune invocation panel (e.g., the function named Execute an experiment), and as it is shown in Figure 7, it provides a selection panel in which select the method to be executed according to the category and the description available.

4 RESULT & ANALYSIS

To analyze the FL aggregation algorithms, FedAvg and FedProx, simulations of different scenarios have been performed by executing the published methods with suitable inputs to dictate the expected method's behaviour and corresponding results stored for further analysis. A careful value assignment to μ is important

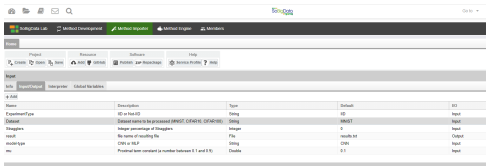

Figure 6: SoBigData platform: Method Importer view.

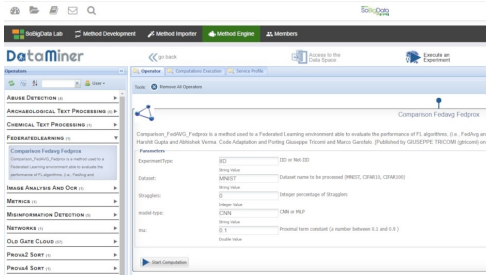

Figure 7: SoBigData platform: Method Invoker view.

in the FedProx algorithm. So for the entire simulation, the value of μ is taken as 0.3.

Table 1 shows the accuracy obtained when the probability of stragglers is 0%, meaning there is no straggler in the entire FL setup. When the dataset is MNIST, and the model is a CNN, the accuracy of FedProx is 90.91%, but it is 81.90% in the case of the FedAvg algorithm. So It can be deduced that in the case of NIID data, the FedProx algorithm performed better than the FedAvg algorithm. Also, the performance of the FedProx algorithm is better in the case of IID data as compared to NIID data, which is represented in figures 8 and 9.

Figures 10 and 11 represent the comparative performance of FedAvg and FedProx for all the possible combinations of simulations performed in this work. Here in the case of any dataset or any model, the performance of the FedProx algorithm is better for NIID data than the FedAvg algorithm. While in the case of only IID data, the performance difference between FedAvg and FedProx algorithms is minimal. So in any setup where the distribution is IID, the FedAvg can be preferred, but when the distribution becomes NIID, adopting the FedProx algorithm would show its benefits. So it can be confirmed that the FedProx can deal with the issue of statistical heterogeneity in FL.

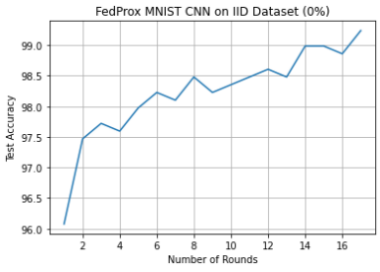

Figure 8: Accuracy Graph for FedProx in IID data.

Table 1: Performance comparison for FedAvg and FedProx for 100 Clients , 0% stragglers, μ=0.3 and learning rate=0.01.

Dataset	Distribution	Algorithm	Model	Accuracy(%)
MNIST	IID	FedAvg	CNN	93
MNIST	IID	FedProx	CNN	92.66
MNIST	Non-IID	FedAvg	CNN	81.90
MNIST	Non-IID	FedProx	CNN	90.91
MNIST	IID	FedAvg	MLP	90.28
MNIST	IID	FedProx	MLP	89.87
MNIST	Non-IID	FedAvg	MLP	76.09
MNIST	Non-IID	FedProx	MLP	82.33
CIFAR	IID	FedAvg	CNN	83.03
CIFAR	IID	FedProx	CNN	81.75
CIFAR	Non-IID	FedAvg	CNN	77.28
CIFAR	Non-IID	FedProx	CNN	79.17
CIFAR	IID	FedAvg	MLP	76.23
CIFAR	IID	FedProx	MLP	75.97
CIFAR	Non-IID	FedAvg	MLP	69.14
CIFAR	Non-IID	FedProx	MLP	73.56

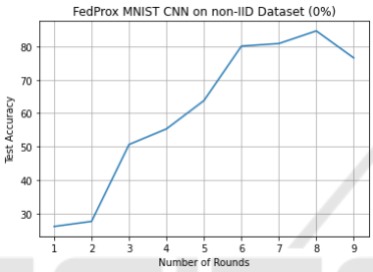

Figure 9: Accuracy Graph for FedProx in NIID data.

The results obtained with 50% probability for the presence of stragglers, 100 participating clients, and μ=0.3 is shown in Table 2. The accuracy of the FedProx algorithm in the NIID data case is always greater than the FedAvg algorithm. So it can be concluded that the performance of the FedProx algorithm is better in the case of NIID data, as compared to the FedAvg algorithm. It is also observed that with the MNIST dataset, the FedProx works better than FedAvg in the case of IID data but, on the other hand, in the case of the CIFAR10 dataset, FedAvg works better than FedProx in IID data.

The performance of FedProx with MNIST IID distribution is better than CIFAR10 IID distribution which can be seen in Figures 12 and 13. Hence, it can be inferred that in NIID data with 50% stragglers, the performance of FedProx is better than FedAvg. However, in the case of IID data with 50% stragglers, the performance of the FedProx algorithm mostly depends on the type and properties of the dataset being used. Thus, by these results, it can be deduced that FedProx can deal both with system heterogeneity (i.e., characterized by the occurrence of stragglers) and statistical heterogeneity (i.e., dataset distributions and their properties).

Table 3 shows the accuracy obtained when the percentage of occurrence of stragglers is 90%. In this scenario, it has been observed that in both cases, IID and NIID data, the performance of the FedProx algorithm is better than FedAvg. Figures 14 and 15 show the accuracy at different rounds for the FedAvg and FedProx algorithms with IID data. Similarly, Figures 16 and 17 represent the accuracy of the FedAvg and FedProx algorithms at different rounds with NIID distribution. Looking at these simulation results, it can be deduced that FedProx performs better in intense heterogeneous environments as compared to the FedAvg algorithm.

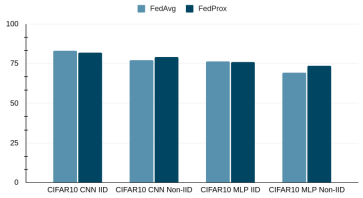

Figure 10: Performance Comparison of FedAvg and FedProx with 0% stragglers and μ=0.3 using CIFAR10 dataset.

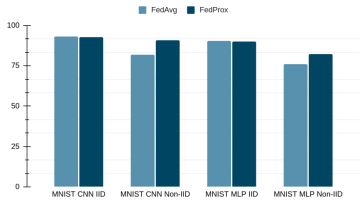

Figure 11: Performance Comparison of FedAvg and FedProx with 0% stragglers and μ=0.3 using MNIST dataset.

Putting together all the previous results, it can be deduced that the FedProx algorithm is always preferred to the FedAvg algorithm, especially when dealing with both system and statistical heterogeneity. In the SLICES research infrastructure where heterogeneous sites (e.g., featuring the availability of re-

Table 2: Performance comparison for FedAvg and FedProx for 100 Clients , 50% stragglers, μ=0.3 and learning rate=0.01.

Dataset	Distribution	Algorithm	Model	Accuracy(%)
MNIST	IID	FedAvg	CNN	97.34
MNIST	IID	FedProx	CNN	99.08
MNIST	Non-IID	FedAvg	CNN	89.24
MNIST	Non-IID	FedProx	CNN	96.57
MNIST	IID	FedAvg	MLP	91.55
MNIST	IID	FedProx	MLP	92.46
MNIST	Non-IID	FedAvg	MLP	85.72
MNIST	Non-IID	FedProx	MLP	94.07
CIFAR	IID	FedAvg	CNN	64.09
CIFAR	IID	FedProx	CNN	52.89
CIFAR	Non-IID	FedAvg	CNN	84.47
CIFAR	Non-IID	FedProx	CNN	94.59
CIFAR	IID	FedAvg	MLP	55.18
CIFAR	IID	FedProx	MLP	51.48
CIFAR	Non-IID	FedAvg	MLP	77.28
CIFAR	Non-IID	FedProx	MLP	93.30

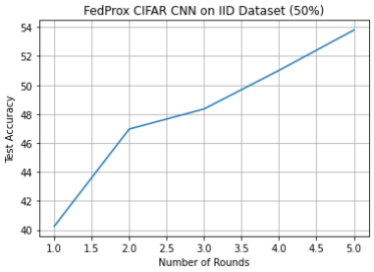

Figure 12: Performance of FedProx on CIFAR10 IID data with 50% Stragglers.

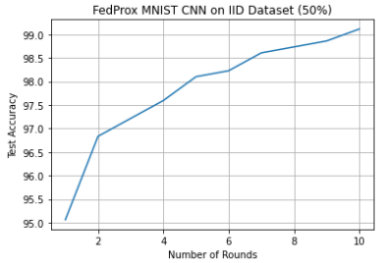

Figure 13: Performance of FedProx on MNIST IID data with 50% Stragglers.

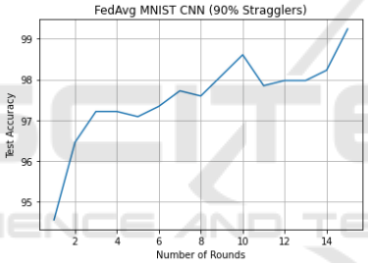

Figure 14: Performance of FedAvg on MNIST IID data with 90% Stragglers.

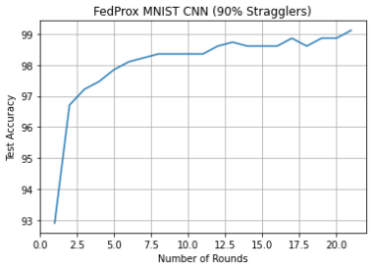

Figure 15: Performance of FedProx on MNIST IID data with 90% Stragglers.

sources with wide-ranging computational and/or storage capabilities) establish together a federation, the benefits of the FedProx algorithm to perform distributed privacy-preserving training without significant performance degradation (e.g., due to the inevitable presence of stragglers) can be highlighted distinctly.

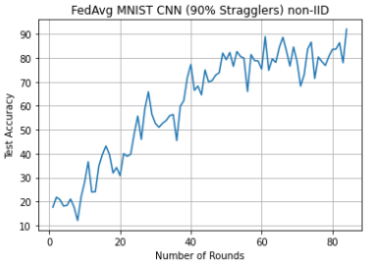

Figure 16: Performance of FedAvg on MNIST NIID data with 90% Stragglers.

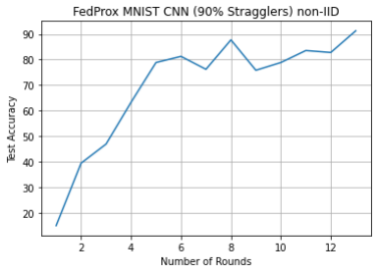

Figure 17: Performance of FedProx on MNIST NIID data with 90% Stragglers.

Table 3: Performance comparison for FedAvg and FedProx for 100 Clients , 90% stragglers, μ=0.3 and learning rate=0.01.

Dataset	Distribution	Algorithm	Model	Accuracy(%)
MNIST	IID	FedAvg	CNN	98.37
MNIST	IID	FedProx	CNN	99.08
MNIST	Non-IID	FedAvg	CNN	92.08
MNIST	Non-IID	FedProx	CNN	91.05
MNIST	IID	FedAvg	MLP	82.72
MNIST	IID	FedProx	MLP	93.68
MNIST	Non-IID	FedAvg	MLP	89.09
MNIST	Non-IID	FedProx	MLP	94.19
CIFAR	IID	FedAvg	CNN	79.52
CIFAR	IID	FedProx	CNN	87.43
CIFAR	Non-IID	FedAvg	CNN	83.12
CIFAR	Non-IID	FedProx	CNN	90.32
CIFAR	IID	FedAvg	MLP	73.39
CIFAR	IID	FedProx	MLP	75.27
CIFAR	Non-IID	FedAvg	MLP	71.42
CIFAR	Non-IID	FedProx	MLP	76.68

5 CONCLUSIONS AND FUTURE SCOPE

The proposed work performed an empirical analysis of the FedAvg and FedProx algorithms to evaluate their behaviour in the presence of system and statistical data heterogeneity. The simulation is performed by varying the number of stragglers to check the performance in the presence of IID and NIID data distributions. The simulation of the FedAvg and FedProx algorithms performed on MNIST and CIFAR10 in IID and NIID scenarios shows that FedProx han-

dles NIID data much better than FedAvg. Also, the CNN model performs better than MLP in this simulation arrangement. It is found that FedProx with straggling clients outperformed FedAvg, likely due to the inherent randomness of the client's selection.

The work opens the area for future research where such algorithms could be analysed in more complex environments with multiple probabilities of stragglers and different data distributions. The results obtained in this work show promise about the suitability of Federated Learning approaches in the SLICES research infrastructure, where heterogeneous devices with their private data distribution may participate in a (common) experiment to achieve the benefits of programmable environments with many participating clients without relinquishing privacy concerns and unique site requirements.

ACKNOWLEDGMENT

This work was partially supported by the Horizon Europe "Scientific Large-Scale Infrastructure for Computing/Communication Experimental Studies-preparation project" (SLICES-PP), under grant 101079774.

REFERENCES

Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. "Understanding of a convolutional neural network". In: 2017 International Conference on Engineering and Technology (ICET). 2017,

Cresci, S., Minutoli, S., Nizzoli, L., Tardelli, S., Tesconi, M. (2019). Enriching Digital Libraries with Crowdsensed Data. In: Manghi, P., Candela, L., Silvello, G. (eds) Digital Libraries: Supporting Open Science. IRCDL 2019. Communications in Computer and Information Science, vol 988. Springer, Cham. https://doi.org/10.1007/978-3-030-11226-4_12

De Vita, F., & Bruneo, D. (2020). Leveraging Stack4Things for federated learning in intelligent cyber physical systems. Journal of Sensor and Actuator Networks, 9(4), 59.

Maharshi Dhada, Amit Kumar Jain, and Ajith Kumar Parlikad. "Empirical Convergence Analysis of Federated Averaging for Failure Prognosis". In: IFAC-PapersOnLine 53.3 (2020). 4th IFAC Workshop on Advanced Maintenance Engineering, Services and Technologies - AMEST 2020

Ahmed Gad. Introduction to Federated Learning. Apr. 2020.

Fosca Giannotti, Roberto Trasarti, Kalina Bontcheva, and Valerio Grossi. 2018. SoBigData: Social Mining & Big Data Ecosystem. In Companion Proceedings of The Web Conference 2018 (WWW '18). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 437–438. https://doi.org/10.1145/3184558.3186205

General Data Protection Regulation. https://gdpr-info.eu/

Gupta, H., Patel, D., Makade, A., Gupta, K., Vyas, O. & Puliafito, A. Risk Prediction in the Life Insurance Industry Using Federated Learning Approach. *2022 IEEE 21st Mediterranean Electrotechnical Conference (MELECON)*. pp. 948-953 (2022)

Birjit Gope et al. "Handwritten Digits Identification Using Mnist Database Via Machine Learning Models". In: IOP Conference Series: Materials Science and Engineering 1022.1 (Jan. 2021)

Andrew Hard et al. Federated Learning for Mobile Keyboard Prediction. 2018. doi: 10.48550/ARXIV.1811.03604. url: https://arxiv.org/abs/1811.03604.

Li, T., Sahu, A., Zaheer, M., Sanjabi, M., Talwalkar, A. & Smith, V. Federated Optimization in Heterogeneous Networks. (arXiv,2018). https://arxiv.org/abs/1812.06127

Tian Li et al. Federated Optimization in Heterogeneous Networks. 2018. arXiv

Tian Li et al. Federated Learning: Challenges, Methods, and Future Directions. Aug. 2019

Long, G., Shen, T., Tan, Y., Gerrard, L., Clarke, A. & Jiang, J. Federated Learning for Privacy-Preserving Open Innovation Future on Digital Health. (2021,8)

McMahan, B., Moore, E., Ramage, D., Hampson, S. & Arcas, B. Communication-Efficient Learning of Deep Networks from Decentralized Data. *Proceedings Of The 20th International Conference On Artificial Intelligence And Statistics*. 54 pp. 1273-1282 (2017,4,20), https://proceedings.mlr.press/v54/mcmahan17a.html

Akayuki Nishio and Ryo Yonetani. "Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge". In: ICC 2019 - 2019 IEEE International Conference on Communications (ICC). IEEE, May 2019. doi:10.1109/icc.2019.8761315. url:https://doi.org/10.1109%2Ficc.2019.8761315.

Sinno Jialin Pan and Qiang Yang. "A Survey on Transfer Learning". In: IEEE Transactions on Knowledge and Data Engineering 22.10 (2010), pp. 1345–1359. doi: 10.1109/TKDE.2009.191.

Scientific LargeScale Infrastructure for Computing/Communication Experimental Studies – Starting Community. https://slices-sc.eu/.

Scientific LargeScale Infrastructure for Computing/Communication Experimental Studies. https://slices-ri.eu/slices-ri-objectives/

SLICE-RI. Scientific LargeScale Infrastructure for Computing/Communication Experimental Studies. Url:https://www.slices-ri.eu/slices-ri-objectives/.

Taud, H. & Mas, J. Multilayer Perceptron (MLP). *Geomatic Approaches For Modeling Land Change Scenarios*. pp. 451-455 (2018)

Kang Wei et al. "Federated Learning With Differential Privacy: Algorithms and Performance Analysis". In: IEEE Transactions on Information Forensics and Security 15 (2020), pp. 3454–3469. doi: 10.1109/TIFS.2020.2988575.