IoT

Review

On the Performance of Federated Learning Algorithms for IoT

Mehreen Tahir [1,*] and Muhammad Intizar Ali [2]

[1] SFI Centre for Research Training in Machine Learning, Dublin City University, D09 E432 Dublin, Ireland

[2] School of Electronic and Computer Engineering, Dublin City University, D09 E432 Dublin, Ireland; ali.intizar@dcu.ie

* Correspondence: mehreen.tahir2@mail.dcu.ie

Abstract: Federated Learning (FL) is a state-of-the-art technique used to build machine learning (ML) models based on distributed data sets. It enables In-Edge AI, preserves data locality, protects user data, and allows ownership. These characteristics of FL make it a suitable choice for IoT networks due to its intrinsic distributed infrastructure. However, FL presents a few unique challenges; the most noteworthy is training over largely heterogeneous data samples on IoT devices. The heterogeneity of devices and models in the complex IoT networks greatly influences the FL training process and makes traditional FL unsuitable to be directly deployed, while many recent research works claim to mitigate the negative impact of heterogeneity in FL networks, unfortunately, the effectiveness of these proposed solutions has never been studied and quantified. In this study, we thoroughly analyze the impact of heterogeneity in FL and present an overview of the practical problems exerted by the system and statistical heterogeneity. We have extensively investigated state-of-the-art algorithms focusing on their practical use over IoT networks. We have also conducted a comparative analysis of the top available federated algorithms over a heterogeneous dynamic IoT network. Our analysis shows that the existing solutions fail to effectively mitigate the problem, thus highlighting the significance of incorporating both system and statistical heterogeneity in FL system design.

Keywords: federated learning; distributed machine learning; Internet of Things

check for updates

Citation: Tahir, M.; Ali, M.I. On the Performance of Federated Learning Algorithms for IoT. *IoT* **2022**, *3*, 273–284. https://doi.org/10.3390/iot3020016

Academic Editor: Francesco Bergadano

Received: 2 March 2022
Accepted: 19 April 2022
Published: 22 April 2022

1. Introduction

Currently, the Internet of Things (IoT) and related technologies are playing a pivotal role in designing, controlling, and managing large and complex systems, such as industrial process automation and optimization, surveillance, smart lighting, parking, waste management, leakage management, digital healthcare, smart grids, and many more. The typical architecture of these systems often consists of various heterogeneous and distributed interconnected devices. In traditional ML settings, the data generated by IoT devices is collected and analyzed in central nodes, such as physical servers or cloud clusters, which usually have powerful computational resources. However, IoT devices are usually deployed at dispersed geographical locations with limited control. IoT devices are very resource-constrained and suffer connectivity issues due to low bandwidth. Heterogeneity in hardware, software, and communication are also common challenges for any IoT infrastructure. Considering the above challenges, continuous data collection from a large number of IoT devices and their storage at a centralized location is very often not feasible. Additionally, IoT devices can acquire highly personalized data that needs to be protected, ensuring compliance with GDPR.

To mitigate the aforementioned issues, edge computing is becoming increasingly popular, which facilitates analytics at the device level without the need of transferring data to a cloud service [1]. Initially, the edge and fog computing was proposed to execute simple queries over low-powered, distributed devices [2,3]. A few recent studies in this field have focused on training ML models centrally and then deploying the trained models on local devices to provide personalization, and mobile user modeling [4]. However, with the enhanced storage and computational capabilities of edge devices, we can leverage the local resources on each device to train ML models using distributed data sets. One such ML technique is known as Federated Learning (FL). In FL, the central server sends a copy of an untrained ML model to all clients in the network. Each client computes an update to the globally-shared central model based on its local training data set and sends the updated local model parameters to the server. The server aggregates the updates from all the clients, updates global model parameters, and sends the updated model back to the clients. This process is repeated until the desired results are achieved [5]. A classic example of this is the Google keyboard. When the keyboard shows a possible suggestion, your phone locally stores the information about the current context and whether you clicked the suggestion. This on-device information is then used to improve the next iteration of Gboard suggestions.

The standard FL problem involves learning a globally shared statistical model from data residing on tens to potentially thousands of remote devices. This model is trained under the constraint that the data generated on the edge device is kept and processed locally, with only periodic updates communicated to the central server. Since the data is not shared at any point, the privacy is preserved. It also reduces communication overhead in IoT networks as it only shares periodic model updates instead of sending huge chunks of raw data. However, IoT systems consist of devices with different hardware capabilities (system heterogeneity) which generate massive amounts of data in a highly disparate manner (statistical heterogeneity), while the effects of heterogeneity are long studied and proven in IoT networks, and there have been many attempts to mitigate the problem, these methods cannot fully handle the scale of federated networks. Moreover, the impact of heterogeneity is far worse in federated networks because the model training relies entirely on distributed data sets.

To tackle the federated network challenges, Google formulated the very first vanilla FL algorithm known as FedAvg. However, FedAvg does not account for the intrinsic heterogeneous nature of federated networks. Since then, many algorithms have been developed to balance the global model performance including stochastic-based algorithms [6,7] and primal dual-optimization [8]. Although a significant effort has been put into developing robust, fair, fault-tolerant, and heterogeneity-aware solutions, to the best of our knowledge, the following questions remain unanswered:

1. *How sensitive is the FL training process to the system and statistical heterogeneity? Does this sensitivity change under different algorithms?*
2. *Are existing proposals effective in mitigating the statistical and system heterogeneity?*
3. *If so, are these solutions as effective as they claim? To what extent do they mitigate the network heterogeneity?*

To this end, we reviewed previous works in the field and conducted an extensive set of experiments to assess the impact of statistical and system heterogeneity on FL training. We simulated a heterogeneous federated network and collected a comprehensive set of measurements to analyze the problem empirically. Furthermore, we carefully selected the top six state-of-the-art FL algorithms and stress-tested them under varying levels of heterogeneity in the network. Our experimental setup comprehensively considers statistical and system heterogeneity to evaluate and compare these six algorithms in terms of model performance and training time. These metrics are then used to quantify the existing solutions' ability to mitigate the challenge of heterogeneity in the FL environment. The experimental environment is also carefully controlled and kept identical to ensure a fair comparison while testing these algorithms.

Our experimental results show that heterogeneity significantly affects the FL network and leads to model divergence. The results also show that its impact varies significantly depending on the type of heterogeneity in the network. For example, statistical heterogeneity causes non-trivial performance degradation in FL, including up to 33.76% accuracy drop in some algorithms. The impact of heterogeneity is more significant in the scenarios where both statistical and system heterogeneity is present. We not only observed a drastic degradation in average performance but also 3.42× lengthened training time and undermined fairness. Despite the claims, our experiments show that the existing solutions are less effective in mitigating the negative impact of heterogeneity in FL. Overall, we believe this study highlights the grey zones in existing solutions and provides essential and timely insights for system designers and FL researchers.

Outline: Section 2 discusses heterogeneity in federated IoT networks and covers both, statistical and system heterogeneity. Section 3 presents an overview of background and related previous works. Section 4 presents a brief overview of algorithms selected for this study. Section 5 presents our experimental setup and Section 6 presents evaluation results including the conclusions made from empirical analysis. We conclude our work and present a few future research directions in Section 7.

2. Heterogeneity in Federated IoT Networks

IoT networks are intrinsically heterogeneous. In real-life scenarios, FL is deployed over an IoT network with different data samples, device capabilities, device availability, network quality, and battery levels. As a result, heterogeneity is evidentiary and impacts the performance of a federated network. This section breaks down the heterogeneity and briefly discusses the two main categories, statistical and system heterogeneity.

2.1. Statistical Heterogeneity

Distributed optimization problems are often modeled under the assumption that data is Independent and Identically Distributed (IID). However, IoT devices generate and collect data in a highly dependent and inconsistent fashion. The number of data points also varies significantly across devices which adds complexity to problem modeling, solution formulation, analysis, and optimization. Moreover, the devices could be distributed in association with each other, and there might be an underlying statistical structure capturing their relationship. With the aim of learning a single, globally shared model, statistical heterogeneity makes it difficult to achieve an optimal performance.

2.2. System Heterogeneity

It is very likely for IoT devices in a network to have different underlying hardware (CPU, memory). These devices might also operate on different battery levels and use different communication protocols (WiFi, LTE, etc.) Conclusively, the computational storage, and communication capabilities differ for each device in the network. Moreover, IoT networks have to cope with stragglers as well. Low-level IoT devices operate on low battery power and bandwidth and can become unavailable at any given time.

The aforementioned system-level characteristics can introduce many challenges when training ML models over the edge. For example, federated networks consist of hundreds of thousands of low-level IoT devices, but only a handful of active devices might take part in the training. Such situations can make trained models biased towards the active devices. Moreover, low participation can result in a long convergence time when training. Due to the reasons mentioned above, heterogeneity is one of the main challenges for federated IoT networks, and federated algorithms must be robust, heterogeneity-aware, and fault-tolerant. Recently a few studies have claimed to address the challenge of heterogeneity; therefore, in this paper, we focus on empirical analysis for evaluating the existing approaches and quantifying their effectiveness in mitigating the problem of heterogeneity.

3. Background and Related Work

In the recent few years, there has been a paradigm shift in the way ML is applied in applications, and FL has emerged as a victor in systems driven by privacy concerns and deep learning [9–11]. FL is being widely adopted due to its compliance with GDPR, and it can be said that it is laying the foundation for next-generation ML applications. Despite FL is showcasing promising results, however, it also brings in unique challenges; such as communication efficiency, heterogeneity, and privacy, which are thoroughly discussed in [12–16]. To mitigate these challenges, various techniques have been presented over the last few years. For example, [17] presented an adaptive averaging strategy, and authors in [18] presented an In-Edge AI framework to tackle the communication bottleneck in federated networks. To deal with the resource optimization problem, [19] focused on the design aspects for enabling FL at the network edge. In contrast, [20] presented the Dispersed Federated Learning (DFL) framework to provide resource optimization for FL networks.

Heterogeneity is one of the major challenges faced by federated IoT networks. However, early FL approaches neither consider system and statistical heterogeneity in their design [5,21] and nor are straggler-aware. Instead, there is a major assumption of uniform participation from all clients and a sample fixed number of data parties in each learning epoch to ensure performance and fair contribution from all clients. Due to these unrealistic assumptions, FL approaches suffer significant performance loss and often lead to model divergence under heterogeneous network conditions.

Previously, many research works have tried to mitigate heterogeneity problem in distributed systems via system and algorithmic solutions [22–25]. In this context, heterogeneity results from different hardware capabilities of devices (system heterogeneity) and results in performance degradation due to stragglers. However, these conventional methods cannot handle the scale of federated networks. Moreover, heterogeneity in FL settings is not limited to hardware and device capabilities. Various other system artifacts such as data distribution [26], client sampling [27] and user behavior also introduce heterogeneity (known as statistical heterogeneity) in the network.

Recently, various techniques have been presented to tackle heterogeneity in a federated network. In [28], the authors proposed to tackle heterogeneity via client sampling. Their approach uses a deadline-based approach to filter out all the stragglers. However, it does not consider how this approach affects the straggler parties in model training. Similarly, [29] proposed to reduce the total training time via adaptive client sampling while ignoring the model bias. FedProx [6] allows client devices to perform a variable amount of work depending on their available system resources and also adds a proximal term to the objective to account for the associated heterogeneity. A few other works in this area proposed reinforcement learning-based techniques to mitigate the negative effects of heterogeneity [30,31]. Furthermore, algorithmic solutions have also been proposed that mainly focus on tackling statistical heterogeneity in the federated network. In [7], authors proposed a variance reduction technique to tackle the data heterogeneity. Similarly, [8] proposed a new design strategy from a primal-dual optimization perspective to achieve communication efficiency and adaptivity to the level of heterogeneity among the local data. However, these techniques do not consider the communication capabilities of the participating devices. Furthermore, they have not been tested in real-life scenarios which keeps us in the dark regarding their *actual* performance in comparison to the reported performance. Comparing the conventional and the new upcoming federated systems in terms of heterogeneity and distribution helps us understand the open challenges as well as track the progress of federated systems [32].

A few studies have also been presented to understand the impact of heterogeneity in FL training. In [33], the author demonstrated the potential impact of system heterogeneity by allocating varied CPU resources to the participants. However, the author only focused on training time and did not consider the impact of model performance. In [34], the authors characterized the impact of heterogeneity on FL training, but they majorly focused on system heterogeneity while ignoring the other types of heterogeneity in the systems. Similarly, in [35], the authors used large-scale smartphone data to understand the impact of heterogeneity but did not account for stragglers. However, all of the studies mentioned above failed to analyze the effectiveness of state-of-the-art FL algorithms under the heterogeneous network conditions. Table 1 summarizes these previous works along with their key contributions and drawbacks.

Table 1. Previous works along with their key contributions and drawbacks.

Ref.	Key Contribution	Drawbacks
[33]	Studied the impacts of hardware heterogeneity on training time by allocating varied CPU	• Only considers training time as the metric to quantify the impact of heterogeneity on FL • Does not consider the communication capabilities of participating devices • Does not quantify the effectiveness of existing solutions in mitigating heterogeneity in FL
[34]	Characterized the impact of device and behavioral heterogeneity on the trained model performance and fairness	• Does not consider the training time • Mostly focuses on hyper-parameters while ignoring other contributing factors, e.g., statistical heterogeneity • Does not quantify the effectiveness of existing solutions in tackling heterogeneity in FL
[35]	Characterized the impact of heterogeneity in FL using large-scale smartphone data	• Does not account for stragglers • Does not quantify the effectiveness of existing solutions in mitigating heterogeneity in FL

Our study differs from the existing works in three aspects (i) *we comprehensively consider statistical heterogeneity as well as system heterogeneity in the experimental environment*; (ii) *we focus on quantifying the effectiveness of proposed solutions in mitigating the heterogeneity in FL*; and (iii) *we carefully controlled the test-bed environment to keep all setting identical ensuring a fair comparison of the proposed solutions.*

4. State-of-the-Art FL Algorithms

For this paper, we chose the top six open-source FL algorithms that are easily reproducible. We mainly focus on algorithms that claim to incorporate heterogeneity in their design and comprehensively evaluate and compare them in terms of training time and model performance. Below we briefly summarize these selected six algorithms and their characteristics.

4.1. Fedavg

FedAvg was presented as a secure, privacy-preserving, and communication-efficient algorithm for FL over edge devices. In [21], the authors presented a practical method for the FL based on iterative model averaging. FedAvg assumes uniform participation from all the participating clients and drops out any slow-responding clients. Despite the early success and adoption of FedAvg, it does not fully solve the challenges and problems associated with the heterogeneity.

4.2. Fedprox

FedProx [6] tackles the heterogeneity problem in federated networks by allowing each participating device to perform a variable amount of work. It incorporates partial information from stragglers and adds a proximal term to account for associated heterogeneity, thereby promising a more stable and accurate convergence behavior.

4.3. Fedpd

In [8], the authors proposed a new algorithm design strategy from the primal-dual optimization perspective. It proposes a meta-algorithm called Federated Primal-Dual (FedPD), which attempts to deal with the general non-convex objective while achieving the best possible optimization and communication complexity when data is non-IID.

4.4. Scaffold

In [7], the authors proposed a Stochastic Controlled Averaging algorithm (SCAFFOLD) which uses control variates (variance reduction) to mitigate the adverse effects of data heterogeneity. SCAFFOLD estimates the update direction for the server model and each client and uses their difference to correct the local update. The algorithm is claimed to overcome heterogeneity and converge in significantly fewer rounds of communication.

4.5. Fedmed

In [36], the authors modeled the unpredictability of FL settings as a Byzantine failure and presented a distributed optimization algorithm that is robust against such failures, with a focus on achieving optimal statistical performance. The algorithm aggregates local solutions using a coordinate-wise median and uses only one communication round. The algorithm is claimed to be efficient while achieving statistical optimality and guarantees robustness against Byzantine failure.

4.6. Q-Fedavg

The naive minimization of aggregation loss in a large network may disproportionately advantage or disadvantage the model performance. Ref. [37] proposed a novel optimization objective inspired by fair resource allocation in wireless networks. The algorithm aims at a more uniform accuracy distribution across the federated network. It minimizes the aggregate re-weighted loss such that the devices with higher loss are given a higher relative weight. The algorithm claims to be fair, robust, and efficient.

Table 2 presents a comparison of the six selection algorithms in terms of their ability to incorporate statistical and system heterogeneity. The following section focuses on the experimental evaluation of the above-described algorithms and comparing their performance while treating vanilla FL (FedAvg) as a performance benchmark.

Table 2. FL algorithms and incorporated heterogeneity types.

Algorithms	Statistical Heterogeneity	System Heterogeneity
FedAvg	✗	✗
FedProx	✓	✓
FedPD	✓	✗
SCAFFOLD	✓	✗
FedMed	✗	✓
q-FedAvg	✗	✓

5. Experimental Design

We ran extensive experiments for image processing systems to evaluate the performance of different FL algorithms under heterogeneous network conditions. This section thoroughly discusses the experimental methodology.

5.1. Data Set

Our initial study includes a model for the MNIST digit recognition task. The MNIST data set has a training set of 60,000 28 × 28 grayscale images of the ten digits and a test set of 10,000 examples. This data set is used by almost all federated algorithms for experimentation, giving us a common ground for performance evaluation.

5.2. Experiment Details

For the classic MNIST digit recognition task, we carried out experiments using a convolutional neural network (CNN) with two 5 × 5 convolution layers (the first with 32 channels, the second with 64 each with a padding of 2 and followed by a 2 × 2 max pooling), a fully connected layer with 512 units and ReLu activation with a final softmax output layer. To understand the performance and optimization of different algorithms, we also needed to specify how to distribute data over the network. We distributed the data over 100 clients and studied two ways of data partitioning: the idealistic IID distribution where data is shuffled and distributed among the 100 clients so that each client receives 600 examples and the real-world non-IID distribution where data is sorted by the digit label divided into 200 shards of the size 300 images with each client receiving two shards. These settings let us explore the degree to which the algorithms can break under the data heterogeneity. We incorporate system heterogeneity in our experimental setup in the following two ways:

- Fraction of active clients: where only a specified number of clients are randomly selected to participate in a training round. For our experiments, we specified 100 clients for our simulated IoT network, out of which only 10% will be actively taking part in one training round. The threshold of active clients is chosen randomly but is kept identical for all the algorithms. Moreover, active clients are chosen randomly in each training round. This simulation is backed by the fact that even though IoT networks consist of a large number of devices, it is common for these devices to become unavailable for training due to numerous reasons, i.e., connectivity, low battery, and user interruption. Thus, only a tiny fraction of devices are usually available in any training round.
- Stragglers: which refers to the number of devices that fails to complete the local training rounds. It is common for IoT devices to drop out during the training round due to interruptions in network connectivity. Thus, initially, active devices can also fail to complete the local training and significantly delay the execution of the federated task. To simulate the problem, we set a threshold for devices that act as stragglers.

To thoroughly study the effects of system heterogeneity, we experimented with 10%, 30%, 50%, 70%, and 90% of stragglers. These settings serve as a testing bed, helping us understand how varying levels of system heterogeneity might affect an algorithm's performance.

6. Evaluation & Results

This section is divided into two subsections, Statistical Heterogeneity and System Heterogeneity, to explain the corresponding results.

6.1. Impact of Statistical Heterogeneity

To mimic the ideal conditions of IID data distribution, we randomly shuffle the data and distribute it among the 100 clients. Only 10% of the clients were randomly chosen to participate in a training round. Under these settings, all the federated algorithms performed well, reaching an accuracy as high as 97%. Figure 1a shows the average accuracy of FL algorithms over IID data. These results were expected since all the distributed optimization problems converge faster under IID data distribution.

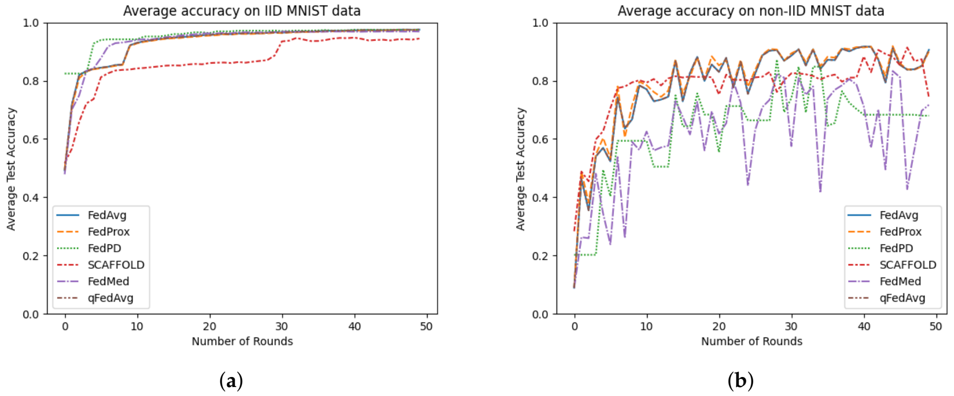


Figure 1. Average accuracy of FL algorithms over (**a**) IID and (**b**) non-IID data distribution for MNIST data set.

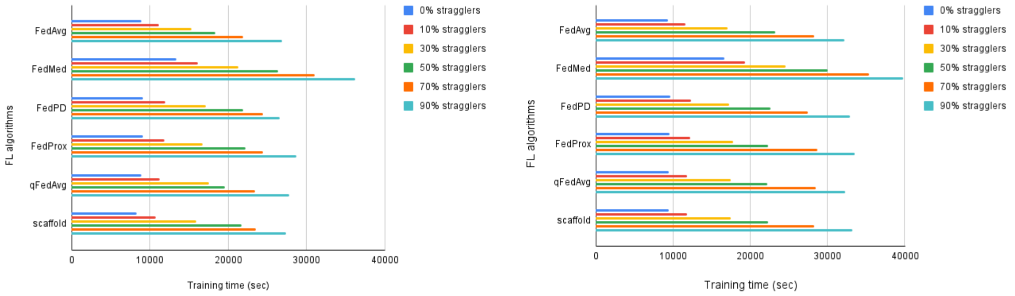
However, federated networks have non-IID data distribution across the network, which reflects in performance degradation in real-world scenarios. The results in Figure 1b show that the statistical heterogeneity significantly affects FL training. We observed an accuracy drop of 7.09% in the case of FedAvg. This accuracy drop does not come as a surprise since FedAvg does not account for any underlying heterogeneity. Our experiments also show a drastic accuracy drop of 30.24%, 26.02%, and 19.22% in the performance of FedPD, FedMed, and SCAFFOLD, respectively. These results were highly unexpected considering the fact that FedPD and SCAFFOLD incorporate data heterogeneity in their design, also claiming to outperform FedAvg.

The impact of statistical heterogeneity becomes more significant if the client population is small or if the participating clients have small numbers of data samples. Although recently developed solutions such as FedPD and SCAFFOLD claim to nullify the effects of statistical heterogeneity and guarantee convergence, not all of them are as *good* as they claim. We observed that FedMed and SCAFFOLD are very sensitive to local data batch size, and even a small change can cause a drastic shift in model performance. SCAFFOLD performs better over small data batches, but that comes at the cost of high training time and is not feasible. On the other hand, FedPD does not show any significant change.

The results show that statistical heterogeneity slightly lengthens the training time, but its impact is more significant on model performance. Despite the claims, all the existing proposals fall short in mitigating the impact of statistical heterogeneity, and none of the FL algorithms could outperform FedAvg.

6.2. Impact of System Heterogeneity

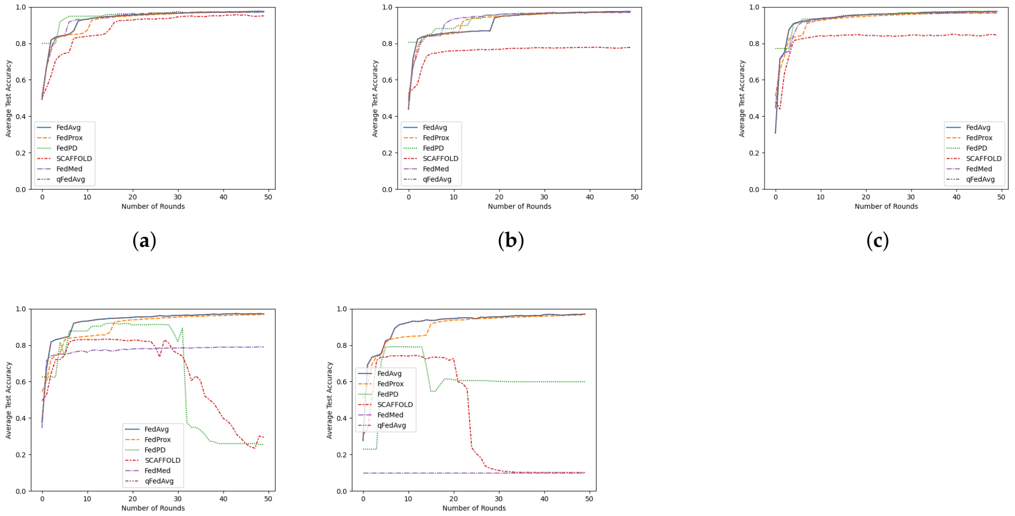
System heterogeneity is deep-rooted in IoT networks and often leads to divergence of the model, bias, etc. In order to estimate the effects of system heterogeneity in federated IoT networks, we set a threshold for the number of stragglers to have a controlled environment and keep the testing set up the same for all algorithms. In our experiments, every client has ten local epochs to complete one training iteration successfully. If a client fails to complete a training round in the given time frame, it is considered a straggler. We carried out experiments over IID and non-IID data distribution with varying levels of stragglers to explore the degree to which algorithms can break. We considered model performance and total training time the key performance metrics for our evaluation. The results, as shown in Figure 2, demonstrate that system heterogeneity lengthens the training time irrespective of whether data distribution is IID or non-IID.



(**a**) (**b**)

Figure 2. Training time of FL algorithms over (**a**) IID and (**b**) non-IID MNIST data for varying levels of stragglers.

The results also show that the impact of heterogeneity varies depending on the scenario and sensitivity of the algorithm. A small threshold of stragglers is less likely to affect the model performance if the network has IID data distribution. However, this impact becomes more significant in the scenarios where both statistical and system heterogeneity come into the picture. For instance, we did not observe any accuracy drop for a threshold of 10% stragglers over IID data distribution, as can be seen in Figure 3. On the other hand, Figure 4 shows the same 10% threshold reports an accuracy drop of 16.04% over non-IID data for FedMed.

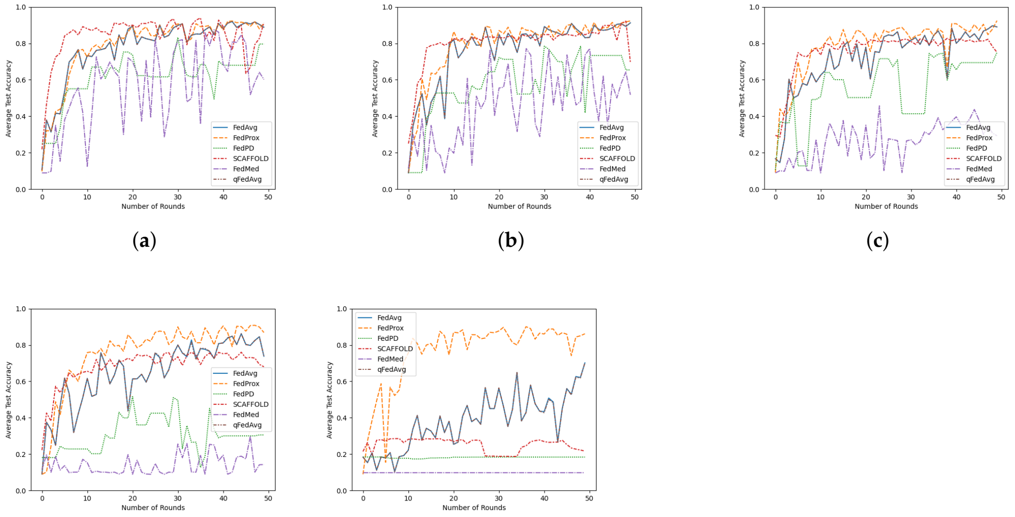


(**a**) (**b**) (**c**)

(**d**) (**e**)

Figure 3. Average accuracy for (**a**) 10%, (**b**) 30%, (**c**) 50%, (**d**) 70%, and (**e**) 90% stragglers over IID data for MNIST data set.

We extended our experiments to include the highly heterogeneous environment where data is distributed in a non-IID manner and up to 90% of the participating devices are stragglers. Under these settings, FedAvg reported an accuracy drop of 22.54% and a lengthened training time up to 3.42×. This behavior can be expected because it does not account for system or statistical heterogeneity in its design. We also observed an accuracy drop of 20.03% and 4.84% in the performance of qFedAvg and FedProx, respectively, while FedMed, FedPD, and SCAFFOLD led to model divergence.



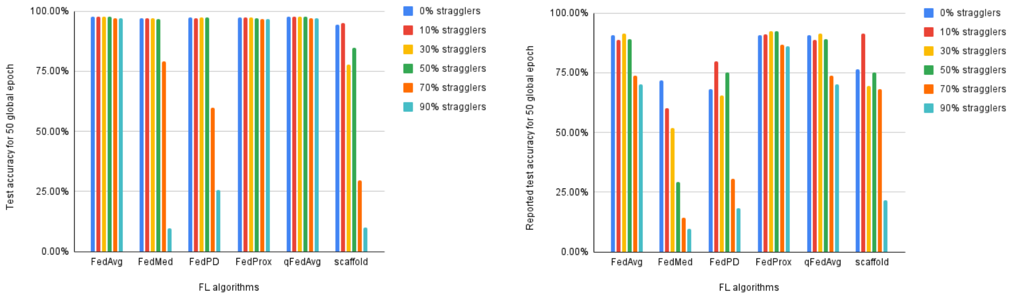
(**a**) (**b**) (**c**)

(**d**) (**e**)

Figure 4. Average accuracy for (**a**) 10%, (**b**) 30%, (**c**) 50%, (**d**) 70%, and (**e**) 90% stragglers over non-IID data for MNIST data set.

Our experiments show that the existing solutions are less effective in improving the training process when heterogeneity is considered. Instead, they are proved to be more sensitive to heterogeneity and thus lead to model divergence.

Figure 5 shows that all the FL algorithms report performance degradation under heterogeneous network conditions. The impact is more significant on model performance in scenarios where both system and statistical heterogeneity are present. Here, the devices are chosen at random for each training iteration which skews the distribution more and results in model divergence. Hence, we conclude that both statistical and system heterogeneity affects a federated network's performance and we cannot simply ignore the repercussions of either.



(**a**) (**b**)

Figure 5. Average accuracy of FL algorithms over (**a**) IID and (**b**) non-IID MNIST data for varying level of stragglers.

7. Conclusions

We present a comprehensive experimental study to anatomize the potential impact of statistical and system heterogeneity on the performance and training time of the collaboratively learned models in FL settings. We evaluate state-of-the-art FL algorithms in their ability to mollify the negative implications of heterogeneity. Although, recent FL algorithms claim to be resilient and guarantee convergence under heterogeneous settings, this study shows that the existing solutions are less effective when system and statistical heterogeneity are considered. This situation is especially distressful for smart system manufacturers and

IoT solution providers who demand highly optimum and adaptable mechanisms, while FL aims at laying the foundation for technology that extends security, the lack of resilient, heterogeneity-aware solutions is a major roadblock to achieving this goal.

We believe that our study not only highlights the limitations of existing algorithms but also provides timely insight for FL system designers. Although an optimizer's choice depends on the nature of the learning problem, FL researchers can use this study to design more efficient and heterogeneity-aware FL systems. The choice of local epochs and batch size has to be moderate since there is a trade-off in training time and model performance. The performance benchmarks should be introduced for all learning domains to keep track of progress in federated IoT networks. Finally, we emphasize the importance of testing and fine-tuning the proposed solutions in realistic, heterogeneous environments to achieve better performance.

Funding: This work is funded by Science Foundation Ireland through the SFI Centre for Research Training in Machine Learning under Grant Number 18/CRT/6183 and also under Grant Number SFI/12/RC/2289_P2 (Insight).

Data Availability Statement: The data used in this study is publicly available. The conducted experiments and results are also accessible on GitHub.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Garcia Lopez, P.; Montresor, A.; Epema, D.; Datta, A.; Higashino, T.; Iamnitchi, A.; Barcellos, M.; Felber, P.; Riviere, E. Edge-centric Computing. *ACM Sigcomm Comput. Commun. Rev.* **2015**, *45*, 37–42. [CrossRef]
2. Bonomi, F.; Milito, R.; Zhu, J.; Addepalli, S. Fog computing and its role in the internet of things. In Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing—MCC '12, Helsinki, Finland, 17 August 2012. [CrossRef]
3. Madden, S.R.; Franklin, M.J.; Hellerstein, J.M.; Hong, W. TinyDB: An acquisitional query processing system for sensor networks. *ACM Trans. Database Syst.* **2005**, *30*, 122–173. [CrossRef]
4. Kuflik, T.; Kay, J.; Kummerfeld, B. Challenges and Solutions of Ubiquitous User Modeling. *Ubiquitous Disp. Environ.* **2012**, 7–30. [CrossRef]
5. Federated Learning: Collaborative Machine Learning without Centralized Training Data, 2017. Available online: https://bigmedium.com/ideas/links/federated-learning.html (accessed on 18 April 2022).
6. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated Optimization in Heterogeneous Networks. *arXiv* **2020**, arXiv:1812.06127.
7. Karimireddy, S.P.; Kale, S.; Mohri, M.; Reddi, S.J.; Stich, S.U.; Suresh, A.T. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. *arXiv* **2021**, arXiv:1910.06378.
8. Zhang, X.; Hong, M.; Dhople, S.; Yin, W.; Liu, Y. FedPD: A Federated Learning Framework with Optimal Rates and Adaptivity to Non-IID Data. *arXiv* **2020**, arXiv:2005.11418.
9. Abdulrahman, S.; Tout, H.; Ould-Slimane, H.; Mourad, A.; Talhi, C.; Guizani, M. A Survey on Federated Learning: The Journey from Centralized to Distributed On-Site Learning and Beyond. *IEEE Internet Things J.* **2021**, *8*, 5476–5497. [CrossRef]
10. Zhang, P.; Sun, H.; Situ, J.; Jiang, C.; Xie, D. Federated Transfer Learning for IIoT Devices with Low Computing Power Based on Blockchain and Edge Computing. *IEEE Access* **2021**, *9*, 98630–98638. [CrossRef]
11. Zhang, P.; Wang, C.; Jiang, C.; Han, Z. Deep Reinforcement Learning Assisted Federated Learning Algorithm for Data Management of IIoT. *IEEE Trans. Ind. Inform.* **2021**, *17*, 8475–8484. [CrossRef]
12. Bonawitz, K.; Eichner, H.; Grieskamp, W.; Huba, D.; Ingerman, A.; Ivanov, V.; Kiddon, C.; Konečný, J.; Mazzocchi, S.; Brendan, M.H.; et al. Towards Federated Learning at Scale: System Design. *arXiv*, **2019**, arXiv:1902.01046.
13. Kairouz, P.; Mcmahan, H.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. Advances and Open Problems in Federated Learning. *arXiv* **2019**, arXiv:1912.04977
14. Khan, L.U.; Saad, W.; Han, Z.; Hossain, E.; Hong, C.S. Federated Learning for Internet of Things: Recent Advances, Taxonomy, and Open Challenges. *arXiv* **2021**, arXiv:2009.13012.
15. Li, T.; Sahu, A.K.; Talwalkar, A.; Smith, V. Federated Learning: Challenges, Methods, and Future Directions. *arXiv* **2019**, arXiv:1908.07873.
16. Aledhari, M.; Razzak, R.; Parizi, R.M.; Saeed, F. Federated Learning: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Access* **2020**, *8*, 140699–140725. [CrossRef] [PubMed]
17. Leroy, D.; Coucke, A.; Lavril, T.; Gisselbrecht, T.; Dureau, J. Federated Learning for Keyword Spotting. *arXiv* **2019**, arXiv:1810.05512.
18. Wang, X.; Han, Y.; Wang, C.; Zhao, Q.; Chen, X.; Chen, M. In-Edge AI: Intelligentizing Mobile Edge Computing, Caching and Communication by Federated Learning. *IEEE Netw.* **2019**, *33*, 156–165. [CrossRef]

19. Khan, L.U.; Pandey, S.R.; Tran, N.H.; Saad, W.; Han, Z.; Nguyen, M.N.H.; Hong, C.S. Federated Learning for Edge Networks: Resource Optimization and Incentive Mechanism. *IEEE Commun. Mag.* **2020**, *58*, 88–93. [CrossRef]
20. Khan, L.U.; Alsenwi, M.; Yaqoob, I.; Imran, M.; Han, Z.; Hong, C.S. Resource Optimized Federated Learning-Enabled Cognitive Internet of Things for Smart Industries. *IEEE Access* **2020**, *8*, 168854–168864. [CrossRef]
21. Brendan, M.H.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A.y. Communication-Efficient Learning of Deep Networks from Decentralized Data. *arXiv* **2016**, arXiv:1602.05629.
22. Chen, C.Y.; Choi, J.; Brand, D.; Agrawal, A.; Zhang, W.; Gopalakrishnan, K. AdaComp: Adaptive Residual Gradient Compression for Data-Parallel Distributed Training. *arXiv* **2017**, arXiv:1712.02679.
23. Jiang, J.; Cui, B.; Zhang, C.; Yu, L. Heterogeneity-aware Distributed Parameter Servers. In Proceedings of the 2017 ACM International Conference on Management of Data, Chicago, IL, USA, 14–19 May 2017. [CrossRef]
24. Schäfer, D.; Edinger, J.; VanSyckel, S.; Paluska, J.M.; Becker, C. Tasklets: Overcoming Heterogeneity in Distributed Computing Systems. In Proceedings of the 2016 IEEE 36th International Conference on Distributed Computing Systems Workshops (ICDCSW), Nara, Japan, 27–30 June 2016. [CrossRef]
25. Thomas, J.; Sycara, K. Heterogeneity, stability, and efficiency in distributed systems. In Proceedings of the International Conference on Multi Agent Systems (Cat. No.98EX160), Paris, France, 3–7 July 1998. [CrossRef]
26. Zawad, S.; Ali, A.; Chen, P.Y.; Anwar, A.; Zhou, Y.; Baracaldo, N.; Tian, Y.; Yan, F. Curse or Redemption? How Data Heterogeneity Affects the Robustness of Federated Learning. *arXiv* **2021**, arXiv:2102.00655.
27. Cho, Y.J.; Wang, J.; Joshi, G. Client Selection in Federated Learning: Convergence Analysis and Power-of-Choice Selection Strategies. *arXiv* **2020**, arXiv:2010.01243.
28. Nishio, T.; Yonetani, R. Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge. In Proceedings of the ICC 2019—2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019. [CrossRef]
29. Luo, B.; Xiao, W.; Wang, S.; Huang, J.; Tassiulas, L. Tackling System and Statistical Heterogeneity for Federated Learning with Adaptive Client Sampling. *arXiv* **2021**, arXiv:2112.11256.
30. Pang, J.; Huang, Y.; Xie, Z.; Han, Q.; Cai, Z. Realizing the Heterogeneity: A Self-Organized Federated Learning Framework for IoT. *IEEE Internet Things J.* **2021**, *8*, 3088–3098. [CrossRef]
31. Wu, Q.; He, K.; Chen, X. Personalized Federated Learning for Intelligent IoT Applications: A Cloud-Edge based Framework. *IEEE Comput. Graph. Appl.* **2020**, *1*, 35–44. [CrossRef]
32. Li, Q.; Wen, Z.; Wu, Z.; Hu, S.; Wang, N.; Li, Y.; Liu, X.; He, B. A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection. *arXiv* **2021**, arXiv:1907.09693.
33. Chai, Z.; Anwar, A.; Zhou, Y.; Baracaldo, N.; Ludwig, H.; Fayyaz, H.; Fayyaz, Z.; Cheng, Y. Towards Taming the Resource and Data Heterogeneity in Federated Learning, 2019. Available online: https://mason-leap-lab.github.io/docs/opml19-fl.pdf (accessed on 18 April 2022).
34. Abdelmoniem, A.M.; Ho, C.Y.; Papageorgiou, P.; Bilal, M.; Canini, M. On the Impact of Device and Behavioral Heterogeneity in Federated Learning. *arXiv* **2021**, arXiv:2102.07500.
35. Yang, C.; Wang, Q.; Xu, M.; Chen, Z.; Bian, K.; Liu, Y.; Liu, X. Characterizing Impacts of Heterogeneity in Federated Learning upon Large-Scale Smartphone Data. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021. [CrossRef]
36. Yin, D.; Chen, Y.; Kannan, R.; Bartlett, P. Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates. *arXiv* **2018**, arXiv:1803.01498.
37. Li, T.; Sanjabi, M.; Beirami, A.; Smith, V. Fair Resource Allocation in Federated Learning. *arXiv* **2020**, arXiv:1905.10497.