UNIVERSITY OF EXETER

Computer Science Department

B.Sc. Computer Science
Computer Science Department

Continual Federated Learning for Uninterrupted IoT Operations

Candidate

William Droin

Student ID 690058620

Supervisor

Dr. Jia Hu

University of Exeter

Co-supervisor

Zeyu Fu

Academic Year
2022/2023

Abstract

The growing demand for Artificial Intelligence-enabled IoT devices has highlighted the importance of efficient and privacy-compliant learning techniques in the face of ever-changing real-world scenarios. IoT devices must continually adapt to rapidly evolving data streams, posing significant challenges. Continual Learning (Lifelong Learning) has recently emerged as a promising solution, allowing incremental learning on continuous data streams. When combined with Federated Learning, Continual Learning offers a previously unexplored advantage: the absence of costly training rounds, which could enable uninterrupted 24/7 operation for IoT devices. In this study, we present a comprehensive analysis of the potential of Federated Continual Learning for seamless IoT operations, shedding light on its feasibility and future applications.

	Yes	No
I certify that all material in this dissertation which is not my own work has been identified.	☑	☐
I give the permission to the Department of Computer Science of the University of Exeter to include this manuscript in the institutional repository, exclusively for academic purposes.	☑	☐

Contents

1 Introduction

Securing personal data has been a growing concern for the past 10 years, users are more and more resentful towards big tech and their tendency to utilise personal data in non-ethical ways, especially since the scandal with Facebook and Cambridge Analytica[1]. There was a big push for more respectful ways to interact with private data and new regulations from different states and government agencies made the training of models, on private data, more and more difficult. Securely keeping the data on centralised servers was not sufficient anymore, people wanted to be in control of their own information, and therefore, new solutions needed to be found.

From all of those concerns, a completely new way of doing machine learning emerged from Google researchers [2]: Federated Learning. Traditionally, machine learning on private information was done by collecting personal data from the user's devices, centralising it onto a server and training a model on the agglomerated data. However, centralising data created many problems regarding data privacy, security or compliance difficulties with regional data regulations. Federated Learning solves those problems because the data stays on the user's devices at all times. By flipping the traditional machine learning paradigm on its head, instead of collecting the data and training one model on this centralised data it would train small models on each device, based on local data, gather the locally trained models and agglomerate them to create one general model.

Federated Learning poses 3 new challenges: efficiency of the training, quality of the aggregation and securing private data. Firstly, since training is done on smartphones and other low-power devices a good balance between power, efficiency and quality of training is key for a successful FL model. Secondly, aggregating large amounts of models into one global model poses a problem regarding which technique to use to maximise performance for every device. Finally, since the data never leaves clients' devices, securing it is done in a completely different way compared to the more traditional artificial intelligence paradigm. Federated learning appeared in 2016 and since then has gained a lot of traction with many different applications, however, Edge Intelligence for IoT devices in particular has seen the most disruption from FL.

In order to have a quality Federated Learning strategy many IoT devices' shortcomings need to be considered. Devices are embedded into the real world, which is constantly changing and therefore the data gathered by such devices can be completely different from one device to another. The data distribution of a device can also drastically change over time, posing problems with some training strategies. IoT devices are also not all made equal, with different battery sizes, unequal computational power, diverging availability for FL training rounds, and limited or costly communication. Considering all of those challenges makes designing a one size fits all Federated Learning strategy usually not desirable, which is why in recent years we have seen many strategies specially tailored to certain "niche" scenarios.

One specific scenario where more research is needed is Federated Learning for time-critical IoT Operations. Traditional Federated Learning requires time-consuming and computationally heavy training rounds which automatically disqualifies it as a viable solution for continuous round-the-clock IoT operations since devices would have to stop their usual task to participate in the training round. There is however one way to train on a continuous data stream: Continual Learning. Combining Federated Learning with Continual Learning could mitigate the need for training rounds and therefore allow continuous IoT operation while benefiting from federated learning. Another Benefit of Continual learning specifically for IoT is the capacity to adapt on the fly, the performance is not static and this is especially important in the context of IoT devices embedded in the real world. A use case that could benefit from adaptative performance is cameras that count people in crowded areas, they need to run continuously and can't allocate time for AI training, they need to adapt to changes like from sunny to rainy people are not going to be dressed the same and umbrellas might interfere with the count and finally no crowd is like another and continual learning helps to personalise for the specific kind of data it receives.

Based on the above assessment of the current available research, this paper will be conducted through the prism of the following research question: "Will Federated Continual Learning allow IoT devices to run without interruption in situations where high availability is required ?"

2 Project Specification

2.1 Federated learning a decentralised approach

Federated Learning was first introduced by Brendan McMahan and al. [3], in this paper they laid out the foundations for all Federated Learning applications to come.

To help recognise if Federated could be applied to a specific problem, two key aspects were identified: It has to involve real-world data from mobile/IoT devices and the data is privacy sensitive and it is preferable not to centralise it on a datacenter. The typical stages of a federated learning process can be outlined as follows:

- Central server randomly selects a batch of clients and sends them a basic model with predefined parameters.
- The individual clients update the received model by training on their local data (training rounds).
- After the local training is done it sends back the parameters to the server.
- The server takes the average of the parameters and passes them back to the clients for another iteration.
- After the defined number of iterations is completed the server sends back the final and optimised model to all clients.
- Once a predetermined period has elapsed, the process cycles back to the second step.

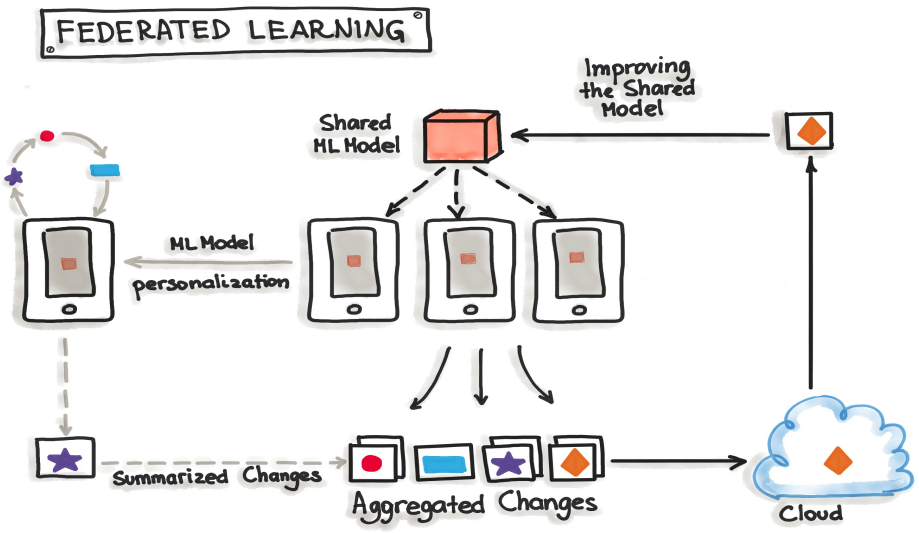

Figure 1: **Federated learning diagram [4].**

Federated learning is therefore capable of aggregating knowledge from various devices without having to ever centralise the data onto a server. From the previous lifecycle, we can outline two key aspects of federated learning that separate it from other machine learning methods: its decentralised nature and one-device training requirements. Training rounds are performed on IoT devices rather than on a centralised server. This ensures that the data stays private, however, it introduces a need for training artificial intelligence models on a low-power device. In the context of phones or smartwatches, their users will usually leave them on a table for the night plugged into the wall to charge up the battery for 5 to 8 hours. Therefore the device has practically no limitations on power consumption or time in order to perform the training. This is unfortunately not the case for every

IoT device, some have very important time constraints and cannot allow to pause their primary task to train artificial intelligence models. Such devices could be 24/7 surveillance systems or health monitoring devices. Those time constraints could cause implementation problems when dealing with Federated Learning for uninterrupted operations.

We've observed that Federated Learning functions very well in distributed and privacy-compliant large-scale IoT device networks. However, federated learning training rounds can cause problems for some IoT devices.

2.2 Continual Learning a nature-inspired paradigm

Humans and animals behave towards new situations based on a life of experience, this is due to the fact that they can continually acquire new knowledge based on said experience[5]. This mechanism is regulated by a complex set of neurocognitive mechanisms that helps the development of long-term memory strength and availability. Preventing the loss of knowledge related to previously encountered tasks is a hard task for animals and synthetic neural networks, this phenomenon is known as catastrophic forgetting. Inspired by this mechanism of nature, Computer scientists came up with lifelong learning[6] (Continual learning) techniques for Artificial Intelligence to reproduce the benefits of the way humans and animals learn.

Continual Learning can be used to tackle an efficiency problem in traditional machine learning, in order to update an model with new knowledge acquired from new data you would have to re-learn the entire data set, meaning previously learned data and new data in order to get the full knowledge. To prevent that inefficiency you could create a new model only trained on the new data, however, in that case the model risks catastrophic forgetting, meaning that after many updates it is at risk of being less effective at certain tasks compared to previous versions of the model.

Apart from being more efficient, there is also another benefit of Continual Learning in scenarios where, during the lifetime of the model, there is a drastic change in terms of incoming data. An example of this could be deploying a sales predictions model for an E-commerce website, the products bought by users might completely shift and make the predictions from the model completely obsolete. This problem is called a "data distribution shift". It is especially a problem with models that interact on a daily basis with the ever-changing real world as demonstrated by the E-commerce example. Another field of computer science is very affected by data distribution shift due to its permanent interaction with the real world: Internet Of Things[7]. Even federated learning with its inherent design advantage for global data distribution shift due to a perpetual exchange of knowledge between devices and the central server, is still very prone to the problem on local data distribution shift.

2.3 Federated Continual Learning for Continuous IoT Operations

As mentioned previously, there is a massive problem with when implementing Federated Learning for constant IoT operation, training on the device makes it difficult to assume both the continuous task and the training rounds. We have also seen that continual learning can learn on a continuous data stream as the device evolves in its environment. Combining both techniques could help bring federated learning to uninterrupted IoT operations and allow for more adaptability from devices' models. By implementing a series of tests we will try to determine the practicality, advantages and disadvantages of federated continual learning (FCL) for 24/7 IoT strategy.

2.4 Challenges with IoT

2.4.1 Non-iid Data

As previously mentioned, IoT devices can be subject to very different environments and therefore their data distribution can vary a lot. Meaning that for a large Federated Learning pool of devices, many could have a vastly different distribution from one another. It is a very common challenge in the context of federated learning and can lead to the following problems:

- Model personalisation: Since the global model is an aggregation of several models that were trained on very different data sets, the global model may not generalise to all clients in the pool correctly and lead to poor performance at the individual and global level.
- Convergence difficulties: Effectively averaging a vast amount of models that were trained on very different data is very difficult and might lead to slower convergence and unstable performance which leads to a poor learning process and model performance.

Those problems can be mitigated by using techniques such as model clustering, and hierarchical training [8, 9].

We can see in Figure 2 a comparasion of iid data and non-iid data, on the left we observe distributions containing all 10 numbers contained int he MNIST data set, this represent the fact that data samples are independently and identically distributed, meaning each data point is independent of others and follows the same probability distribution. On the left we can observe that both phones do not have the full range of numbers, only 4s and 0s for one and 5s and 7s for the other. This means that data samples do not follow the same probability distribution, this could reflect the unique usage patterns or contexts of the individual devices.

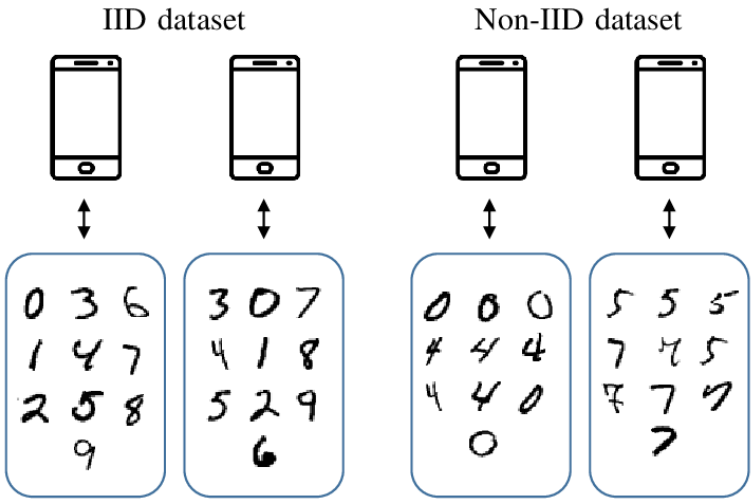

Figure 2: **Non-iid MNIST data distribution** [10]

2.4.2 Dynamic data distribution

As stated previously, IoT devices live in the ever-changing real world, therefore, their data distribution can drastically evolve over time and create challenges. Challenges of dynamic data distribution are exacerbated in Federated Learning and can cause problems like poor performance, drastically changing performance over time or longer convergence time[11]. The main feature of federated learning, on-device learning for privacy, can cause extra challenges when it comes to dynamic data distribution [12] since the people in charge of the Federated learning system do not have any control over the data set. In a centralised machine learning paradigm, if the data distribution changes over time, researchers can modify the training data set to account for the environmental changes, however, this is could not be done in a true decentralised and privacy-compliant Federated Learning strategy. One way to combat the challenges of Dynamic data distribution is to use Continual Learning [2] and contrary to Non-iid data this project will try to tackle the problem of dynamic data distribution by utilising Continual Learning.

2.4.3 Heterogeneity in IoT devices capabilities

When deploying Federated Learning across a large panel of devices, their inherent variety can become a real challenge, especially with some devices like smartphones where things the manufacturer, the

OS, the amount of memory, the computational power, the networking capabilities, and the storage can vary greatly between devices. This heterogeneity in capabilities can, in some cases, drastically reduce the number of devices that can participate in training rounds due to lack of storage, lack of computational power or low networking capabilities. However, on-device Continual Learning has been used to reduce the energy needed to do Machine learning on low-power IoT devices[13]. This project will cover the real-world energy consumption of an IoT device while doing traditional FL and under FCL. This could further demonstrate the capabilities of on-device continual learning for 24/7 operation with energy constraints.

3 Design and Implementation

The basis for any comparison between two techniques in a research environment is to change one parameter at a time and keep biases at their lowest. This is why in the implementation of both the synthetic benchmark and the real-world testing, the dataset, the model, the client pool size and the number of data points per round will all be kept the same and only the training technique will change. This is to ensure that all conclusions drawn from the comparison of FL and FCL are not due to uncontrolled biases coming from an uncontrolled testing environment.

The aim for both simulation and real-world implementation was to use industry-standard tools and hardware in order to have easily reproducible results.

3.1 Simulation Strategy

3.1.1 Tools and Libraries

As stated previously, the goal of this paper was not to develop a brand new framework from scratch but to utilise already existing techniques and code libraries to spark a discussion around uninterrupted IoT operation for Federated learning. Considering this, the choice was made to utilise the following Python libraries to conduct our research:

- Flower: A comprehensive Federated Learning framework compatible with Pytorch. It allows for large-scale simulation and real-world deployments[14].
- Avalanche: A research-focused Library that allows for fast prototyping of Continual Learning[15]
- Pytorch: The industry standard for Artificial Intelligence workloads in Python[16]
- Ray: A framework that aims to facilitate the scaling of Python and Deep Learning workloads[17]. It is already integrated into Flower as a tool to run large-scale Federated Learning simulations.

Those three libraries have been chosen because they are all well-established in their domains and offer a good balance between ease of implementation, customisation and performance.

3.1.2 Simulation Settings

To make a quality assessment of a new approach regarding IoT research, it is important to use a sufficiently large amount of nodes and in the case of Federated learning a sufficient amount of communication rounds. Looking at existing papers on traditional federated learning, the consensus seems to be to use around 10 to 20 clients for 300 to 500 communication rounds[10, 18]. However, when it comes to Federated Continual Learning it seems like researchers tend to focus on a lower number of rounds and individual performance rather than focus on the global average performance[19, 20]. The most likely reason for that is the fact that papers that modify the federated learning algorithm want to stress test their technique with many clients and many rounds to see how it behaves. On the other hand, research modifying the on-device learning method focuses more on the individual clients' performance and how the model converges with different datasets. For this paper will centre our experiments mainly on individual performance since this is where most of the problems of FCL lie. Considering that, most of our simulations will be run with 10 clients and 20 rounds.

3.1.3 Datasets

The dataset chosen for this project were MNIST since it seems to be the industry standard for Federated Learning research [10, 18, 19, 20]. MNIST is a dataset comprised of 70 000 grayscale images of handwritten digits. It is a relatively simple dataset since it only contains 10 classes, it is also computationally efficient which is very important when dealing with IoT devices and MNIST can still represent relatively accurately how a model would perform during real-world computer vision workloads. Figure 2 shows a non-iid sample of the MNIST dataset.

3.1.4 Neural Networks

In this project we used both MLPs (Multilayer Perceptrons) and CNNs (Convolutional Neural Networks) models for the comparison of Federated Continual Learning (FCL) with the MNIST dataset. This choice was made for several reasons:

Widely used for image classification tasks: Both MLP and CNN models are commonly used and well-suited for image classification tasks, such as those involving the MNIST dataset.

Effectiveness of CNNs: CNNs, in particular, have proven to be highly effective in handling image data, as they can automatically learn local spatial features and hierarchies. Their convolutional layers help distinguish patterns and relationships between pixels, making them especially suitable for MNIST classification.

The simplicity of MLPs: While MLPs might not perform as well as CNNs in capturing spatial relationships, they offer simplicity and are easier to train. For simpler tasks like MNIST classification, they can still deliver good results (within 5% of CNNs). This makes MLPs a viable alternative to consider for Federated Learning especially when considering the relatively modest hardware that Fl is deployed on (small embedded computers, wearable devices, phones...).

Differences in architectures: CNNs and MLPs are extremely different in the way they approach machine learning, MLP is a feedforward neural network with multiple layers of neurons, suitable for simple tasks like classification. CNN is a specialized architecture for high-dimensional data, with convolutional and pooling layers for feature extraction, it represents data in a hierarchical way.

Scalability and adaptability: Both MLP and CNN models can be easily scaled up or down by adjusting the number of layers and neurons. This allows for very flexible and customisable model architectures, making it easier to adapt the models to the requirements of a specific task.

Comparison of performance: Employing both MLP and CNN models in FCL allows for a comparison of their performance in a federated learning setting. This can provide valuable insights into the strengths and weaknesses of each model type when used in Continual learning settings where new challenges, like catastrophic forgetting, arise.

In summary, using both MLP and CNN models for FCL with the MNIST dataset is a good idea because they are well-suited for image classification tasks, offer different levels of complexity, and allow for a comparison of their performance in a continual learning context.

3.2 Real world testing

Demonstrating how FCL could be used as an alternative for uninterrupted IoT operation is very challenging in a purely synthetic benchmark like the one described above. For this reason, we decided to include a real-world testing scenario. The real-world testing scenario would be constituted of an IoT device running an object recognition workflow and would be part of a Federated Learning pool. For this experiment, the libraries and datasets previously mentioned will be used in the same way as the simulation

3.2.1 Hardware

The list of components used for this experiment is as follows:

- IoT device: Raspberry Pi 4, 8GB of RAM

- Camera: Raspberry Pi Camera Module 2
- Storage/Hardrive: Samsung EVO 32gb SD card
- OS: Raspberry Pi OS 64 bit Desktop

To continue the trend of using industry-standard tools to conduct this research, a Raspberry PI with a webcam attached to it was chosen as the IoT device for the real-world test. The Raspberry Pi is the preferred choice amongst enthusiasts and companies for low-cost custom IoT projects, therefore, it is very representative in terms of the computational power of what most of the industry is currently using. It is also powerful enough for light machine learning and there is a lot of software support for it. The choice of the Raspberry Pi 4 was also made in the pursuit of having the most reproducible results. Other single-board computers were also considered, such as the jetson nano, which is widely used for image recognition since it has an artificial intelligence accelerator in the form of a low-power Nvidia GPU (graphics card) which can be used with CUDA (Nvidia's compute accelerator software) to improve performance in artificial intelligence workloads. However, this was not the final choice as it was considered too powerful to be representative of what is usually used in the industry. Nonetheless, the current trend is to develop and use more and more single-board computers with built-in AI accelerators like the jetson nano.

3.2.2 Comparison Strategy

The objective of this experiment is to evaluate the effectiveness of traditional FL and FCL when operating on IoT devices that must maintain a continuous workflow. A notable application that demands both IoT integration and uninterrupted operation is "Monitoring." This can take various forms, such as security surveillance systems, monitoring of power plants and factories through numerous sensors, overseeing infrastructure such as bridges or highways, Agricultural monitoring, data centre health assessment etc. In order to showcase the effectiveness of both techniques when it comes to such workflows we will host both a Federated learning server and a pool of simulated clients. On the same network, there will be a raspberry pi running both a Flower FL client connected to the pool and an object recognition workload in order to simulate a typical surveillance system task. The following illustration shows how the entire system is organised.

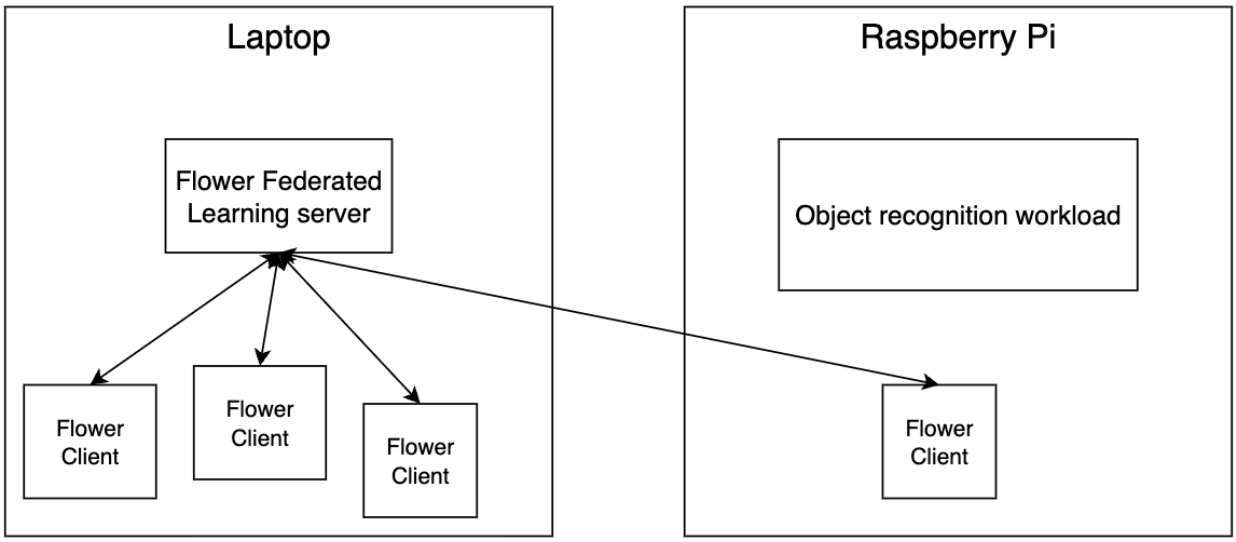

Figure 3: **Real world testing architecture.**

To be able to simulate a continuous data stream in the case of the real-world FCL test, modifications needed to be made to the code: The data stream coming from Avalanche (CL python library) was slowed down to resemble a more realistic continuous data stream. In Figure 2 we can see how the two different strategy were organised:

- FCL: the training of the model is done at the same time as the object recognition workload.
- FL: the raspberry is performing object recognition while waiting for the server to start the training round, and then it starts training on local data while continuing to perform object recognition.

For both tests the object recognition task is left on at all times and both the performance of the object recognition task (measured in FPS) and the power consumption will be monitored.

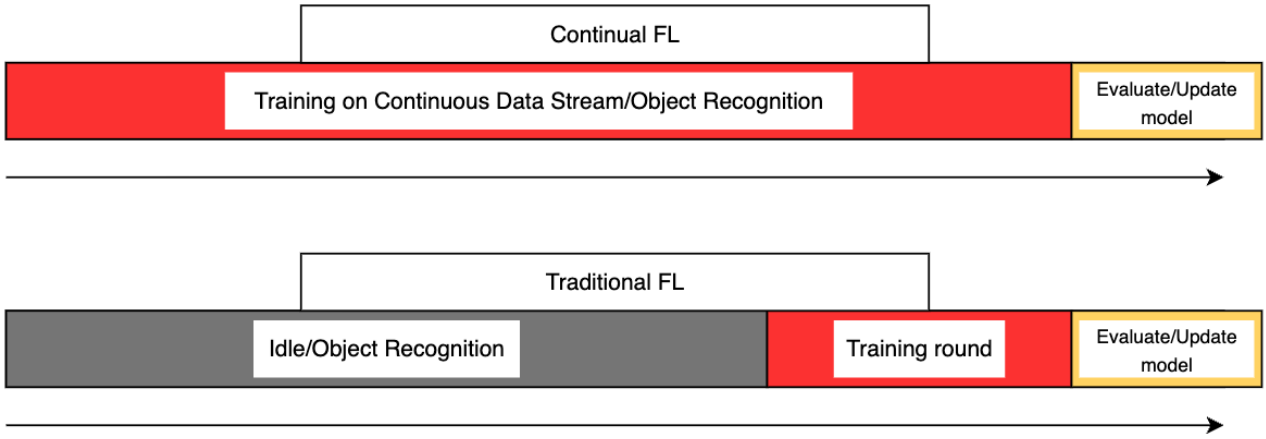

Figure 4: **Life-cycle of a Federated Learning epoch.**

4 Analysis of Results

4.1 Simulation

4.1.1 Performance analysis

As described earlier, the simulation was run multiple times with the following parameters:

- 10 clients
- 20 rounds
- 550 data points per round

Figure 6 and Table 1 show the performance graph of the FCL and FL strategy with two different models each, it demonstrates that both models achieve a very similar level of accuracy: the final accuracies for FL and FCL are within 0.5% for both MLP and CNN. However, we can clearly see that the FCl curves (blue and green) demonstrate a very inconsistent growth pattern when compared to the very steady FL curve (red). This may suggest that FCl is facing challenges in model stability during the training. In order to quantify this inconsistency we can compute the standard deviation of the ROC for both curves using the data in Table 1. Calculating the standard deviation, for the MLP curves, gives a standard deviation of 0.57% and 0.65% for FL and FCL respectively, which clearly shows that Federated Continual Learning has a slight disadvantage over Fl when it comes to its consistency. However, putting this inconsistency into perspective, it remains relatively small since on average FCL would be 0.08% less consistent than FL in our case of study. The inconsistency appears quite significant on the graph, however, it is probably due to the narrow scale of the graph (85%-100%) which exacerbates the problem.

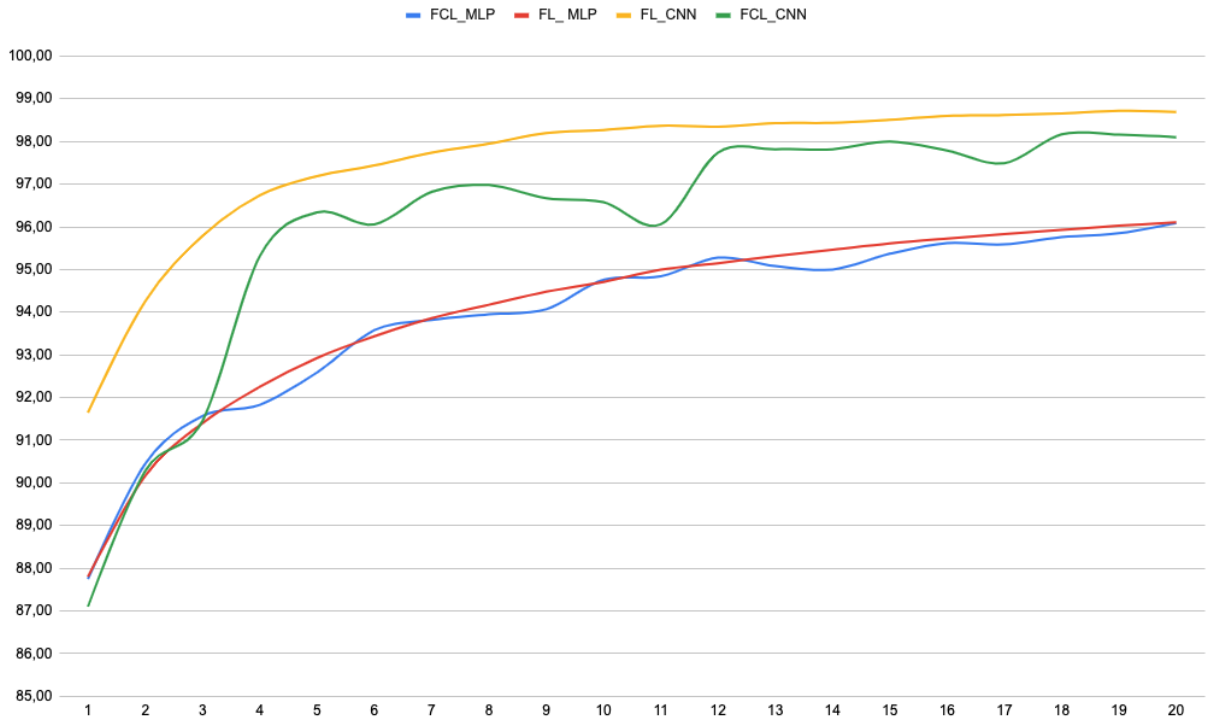

Figure 5: **Performance comparison FL/FCL.**

Table 1: Comparasion FCL FL

round	FL_MLP	FCL_MLP	FL CNN	FCL_CNN
1	87.80	87.75	91.65	87.10
2	90.17	90.45	94.26	90.28
3	91.40	91.57	95.79	91.46
4	92.25	91.83	96.74	95.32
5	92.93	92.59	97.19	96.34
6	93.44	93.58	97.44	96.06
7	93.86	93.82	97.74	96.82
8	94.17	93.95	97.95	96.98
9	94.48	94.07	98.20	96.67
10	94.71	94.76	98.27	96.58
11	95.00	94.84	98.37	96.06
12	95.15	95.28	98.35	97.74
13	95.32	95.08	98.43	97.82
14	95.47	95.00	98.44	97.82
15	95.61	95.37	98.51	98.00
16	95.73	95.62	98.60	97.79
17	95.8	95.59	98.62	97.49
18	95.93	95.76	98.66	98.17
19	96.03	95.85	98.72	98.16
20	96.10	96.09	98.69	98.10

According to various researchers, the main reason for inconsistency in Continual Learning is Catastrophic Forgetting [21, 22], which describes the phenomenon of neural networks losing previously acquired knowledge while learning new tasks. It occurs when the model's weights are updated when confronted with new information, inadvertently overwriting previous knowledge. This is usually not a problem in traditional machine learning since the models are retrained on the totality of the collected data. A few strategies have been proposed to combat Catastrophic forgetting:

- Elastic Weight Consolidation: it is a regularization-based method which means that it imposes constraints and penalties during the learning process to protect previously learned data.

- Replay: it stores previously seen data and interleaves the old stored data within the current data stream.
- Dynamic network expansion: techniques such as Progressive Neural Networks [**?**] Dynamic Expandable Networks [**?**] both have the common principle of expanding the neural network as it learns rather than modifying existing features.

Moving away from the FL/FCL comparison, we can compare the performance between MLP (Multi Layer Perceptron) and CNN (Convolutional Neural Network). The CNN achieves a higher accuracy throughout the entirety of the experiment and seems to converge faster than its MLP counterpart. However, the Convolutional Neural Network seems to be a lot more inconsistent in Federated Continual Learning compared to the MLP in the same scenario. This could mean that a CNN is more subject to catastrophic forgetting than an MLP would. There does not seem to be literature directly comparing the catastrophic forgetting between these two models, however, based on existing research we can make an educated guess on why CNNs could be more affected by catastrophic forgetting than MLPs.

The main architectural difference between these two models is the hierarchical aspect of CNNs[23]. They are designed to learn hierarchical representations of data, especially when it comes to image recognition. These representations are formed during training by combining local features into higher-level patterns. While the CNN is exposed to new data, the previously acquired data hierarchy could be disrupted. This could be the cause of the very inconsistent performance that we are observing. Catastrophic forgetting in convolutional networks is a well-known problem and a few papers present techniques mentioned before and they focus especially on CNN for their analysis for both replay and elastic weights consolidation techniques[6, 8].

We can identify another problem with Figure 5, is that it focuses on the averaged performance of all the clients, in the case of FCL, this could hide the true impact of catastrophic forgetting on an individual scale. In order to cover better the topic of catastrophic forgetting and how to mitigate it, a Replay based method will be used and the focus will be on individual client performance rather than global performance since it is on an individual scale that catastrophic forgetting is the most visible because two IoT devices experiencing catastrophic forgetting could cancel out each other when averaged globally. For this investigation, the replay method will be compared against the "naive" method, which describes the neural network being trained on a new task using standard backpropagation and gradient descent, without any consideration of the impact the new task could have on previously learned tasks.

In order to investigate, a modified version of the previous test in Figure 5 was run with the following parameters: 5 clients, 10 rounds, 500 data points for training and 100 for testing and instead of recording the global accuracy (average accuracy from all clients) the accuracy was recorded individually. A total of four experiments were conducted, with two utilizing the Naive method and the other two employing the Replay technique. Both the Naive and Replay approaches were tested using Multi-Layer Perceptron (MLP) and Convolutional Neural Network (CNN) architectures. The result of this experiment is Figure 4, which shows the resulting graphs for a comparison between a "naive" continual learning approach and a Replay method. The green curves are the CNNs, and the blue ones represent the MLPs. We can clearly see that the Replay method's curve is a lot smoother for both CNNs and MLPs.

Focusing on the blue curves (MLP), they reach 95% accuracy by round 7 with replay. On the other hand, the clients using the Naive method have curves that are a lot less smooth and do not reach 95% accuracy within the 10 rounds of this experiment. When comparing the performance of Convolutional Neural Networks (CNNs) (represented in green) to Multi-Layer Perceptrons (MLPs), CNNs demonstrate an improvement of around 4-5% in both scenarios. Clients using CNNs also seem to benefit a lot from the replay method, they converge faster: reaching 98.5% at the end of the 10 rounds compared to 97.2% for the naive method. They also seem to grow more steadily and computing the average variance from every point there seems to be a lot less divergence between the clients:

- CNN: 0.30% for replay method against 0.54%

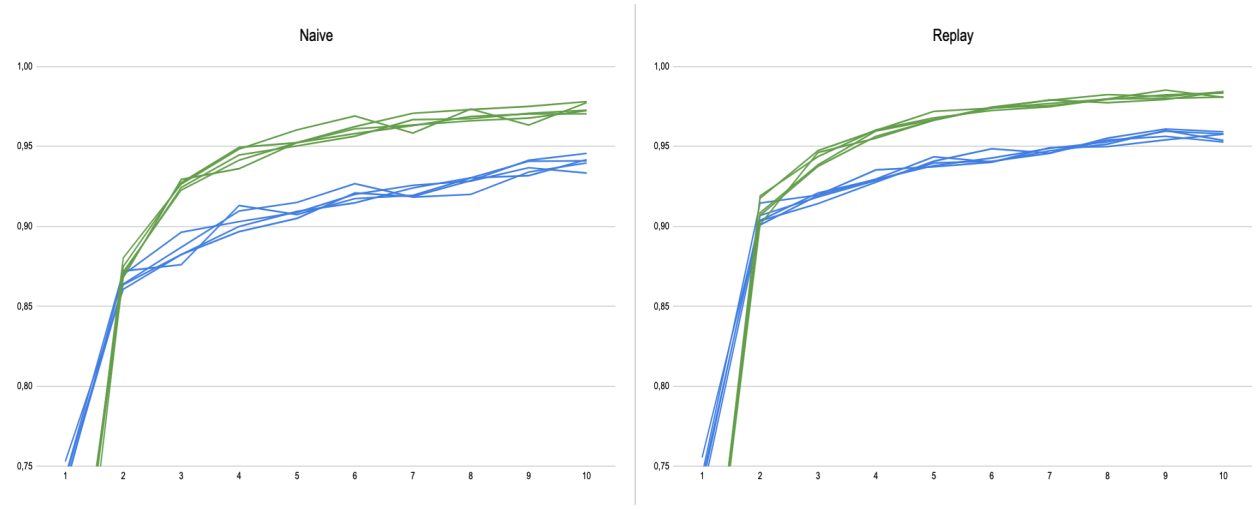

Figure 6: **Comparasion Naive/Replay.** MLP(blue), CNN(green)

- MLP: 0.35% for the replay and 0.50% for the naive method

From the previously computed data, we can see that CNN clients seem to benefit more from replay since they have a decrease of 0.24% between the two methods and MLP have a decrease of only 0.15%. The more significant difference for CNN might confirm the troubles of Convolutional Networks and their hierarchical data representation when it comes to catastrophic forgetting. Although the difference between the values obtained is quite small and falls within the margin of error, the prior comparison of the spread of clients between CNN and MLP may not be relevant. However, the analysis of the reduction in variance between naive and replay methods is substantial enough and supported by numerous research papers [6, 8] to be considered.

Method	**Variance**	**Final Average accuracy**
Naive and MLP	0,5009%	94,56%
Replay and MLP	0,3512%	95,37%
Naive and CNN	0,5413%	97,23%
Replay and CNN	0,3017%	98,33%

Table 2: **Result Table for Naive Replay Comparison.**

In summary, catastrophic forgetting poses a significant challenge in continual learning, where neural networks may lose previously acquired knowledge while learning new tasks. Various strategies, such as elastic weight consolidation[8], replay [6], and dynamic network expansion, have been proposed to address this issue. A comparison between MLP and CNN architectures indicates that CNNs generally outperform MLPs in terms of accuracy, but are more prone to catastrophic forgetting due to their hierarchical structure. Our investigation on an individual client scale, comparing naive and replay methods, demonstrates that replay methods lead to smoother learning curves and reduced divergence between clients, particularly for CNNs. Although the differences in client spread between CNN and MLP might not be significant, the substantial reduction in variance between naive and replay methods is supported by existing research. This highlights the importance of adopting suitable strategies to mitigate catastrophic forgetting in continual learning scenarios, ensuring more robust and reliable performance across various tasks.

4.2 Real world testing

As described in the section Design and Implementation, the real-world testing was run with the following structure:

- 5 total clients: 1 running on the Raspberry Pi and 4 running on a separate computer
- 5 training rounds
- Data stream of MNIST data point per second
- TensorFlow's example for image classification running on the raspberry

The main focus of this experiment was the performance of the image recognition task when ran in parallel with a federated learning client. To measure said performance, FPS (Frames Per Second) will be the metric of choice since it is widely used to assess the performance of real-time image processing techniques[24, 25]. On figure 7, two graphs representing the FPS over time, one for FL and one for FCL, both graphs were made from data captured at the end of epoch 2 and the beginning of epoch 3, meaning that the system experienced a training round, an evaluation round and an update of its model.

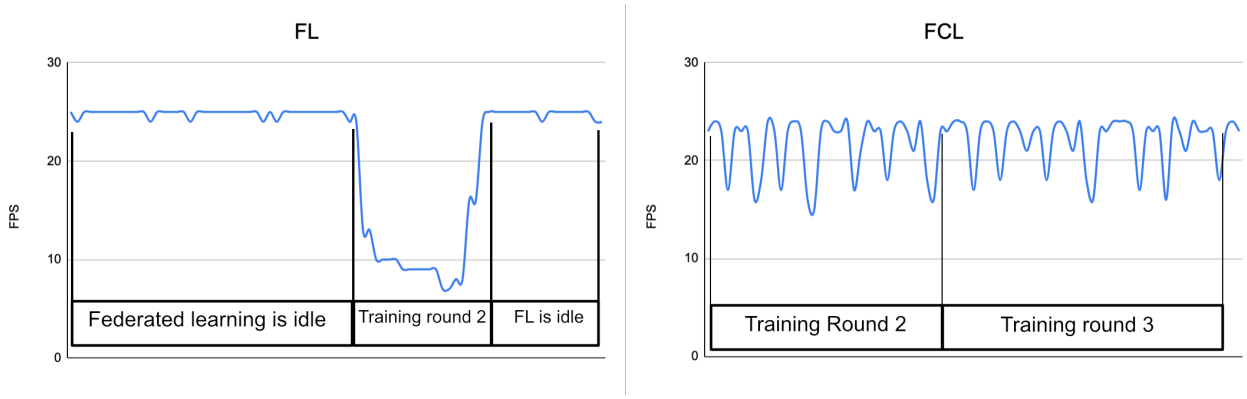

Figure 7: **Frames Per Second for Image recognition during training round.**

After running the experiment, the average FPS was computed and for the entirety of the 10 rounds it was 22.1 FPS for the traditional FL and 19.3 FPS for the Continual Learning method, so overall it looks like FCL performed worse. However, the average FPS does not tell the full story, when looking at the graph for traditional FL, we can observe the different stages of Federated Learning, three parts are clearly observable, the first 2 thirds of the graph is a flat curve oscillating between 24 and 25 FPS and it corresponds to the period where the Raspberry is waiting for a signal form the server to start a training round. As soon as the training starts, there is a dip below 10 FPS and it is sustained for the entirety of the training round. After the end of the training round, the performance recovers to a normal 25 FPS. On the other hand, The FCL is heretically oscillating between 23 FPS and 17 FPS most of the time and the lowest recorded FPS for the entirety of the 5 rounds recorded is 15. There is no discernment possible for the start or the end of the training round.

The Traditional Federated Learning strategy renders the image-processing performance very hard to exploit in an environment where 24/7 uninterrupted operation is mandatory. Furthermore, 8 FPS on a relatively light image processing task could translate to a complete crash of the system with more demanding models. On the other hand, even if the FCL performance seems very unstable in generally stays in an exploitable range (18 to 23 FPS) and there is no significant and unexpected drops in performance.

In conclusion, Traditional FL exhibited a significant dip in FPS during the training rounds, which is unfavourable for applications requiring continuous and uninterrupted operation. In contrast, FCL maintained a more consistent FPS range despite appearing unstable. This suggests that FCL might be more suitable for real-time image processing tasks in scenarios where stability and uninterrupted performance are crucial.

5 Going further than the analysis/Other untested benefits of FCL

5.1 Untested benefits of FCL

There are quite a few benefits of federated continual learning that were not measured in this report and some are especially useful when it comes to IoT. One of those advantages is the ability to adapt on the fly to changing data distributions: In many real-world scenarios, data distributions change over time. FCL can handle these changes more effectively by updating the model as new data becomes available and does not have to wait for a new training round to be updated like traditional FL. This ensures that the model remains relevant and accurate in the face of evolving data patterns and environments. Efficient resource utilisation is another side effect of continual learning since it allows models to learn incrementally, which can be more computationally efficient compared to retraining the entire model on the entire dataset every time new data becomes available. This can be especially useful in resource-constrained environments like IoT. However, this adaptability comes with a trade-off as it can lead to inconsistent performance due to catastrophic forgetting. More frequent global model updates: the models are continuously updated and training rounds are not problematic computationally heavy and time-consuming anymore, global updates can be done more frequently without disturbing the lifecycle of the device, and this can help the global model adapt more quickly to changes in the data distribution. Personalisation and adaptation to local data: In distributed ML scenarios, each client device may have its unique data distribution. FCL can help adapt the global model to these local distributions, resulting in better performance and more personalized models for each client. However, these personalisations would be "overwritten" by the next global model update, which is why in situations where devices have very distinct data distribution (non-iid data), it would be a good idea to cluster devices based on their model's similarities[9].

5.2 Potential use cases

Our preliminary findings open up a large panel of possible use cases for FCL in uninterrupted IoT that seemed out of reach or unpractical with traditional federated learning. Some examples of such scenarios could be:

- Healthcare monitoring systems: Wearable devices that track patients' health could benefit from privacy-compliant decentralised Machine Learning like federated learning, and the uninterrupted nature of its task makes it a very good candidate for federated continual learning.
- Smart city infrastructure: FCL can be applied to traffic management systems, where traffic patterns and traffic jam levels change over time. Continual learning ensures that traffic control performs at all times while keeping the number plates, cars and whereabouts of drivers private.
- Predictive maintenance in manufacturing: Manufacturing plants with a large number of IoT sensors can use FCL to continually update their predictive maintenance models as new failure patterns emerge or as machines age. This ensures that the system remains effective at all times while identifying potential new issues. Federated learning also ensures data privacy when the company producing the IoT devices for predictive maintenance is different from the manufacturing plant owner. This also allows different production facilities to learn from each other while maintaining data privacy.

5.3 Alternative solutions

Throughout the entirety of this project, we laid out the advantages and disadvantages of federated continual learning for uninterrupted IoT, however, there might be other solutions to this problem. In order to speculate on the effectiveness of other potential approaches, we have to evaluate their ability to prevent training rounds from interfering with a task that needs to remain uninterrupted.

On option could be to separate completely the task and the training rounds on two different parts of the IoT device, for example, more and more devices come with embedded AI accelerator,

there could be an opportunity to optimize federated learning processes, enabling faster training and potentially mitigating the need for continual learning by offloading the training to an AI accelerator. There could be quite a few advantages to this approach:

- Enhanced performance: Offloading training rounds to AI accelerators can lead to faster training and inference, enabling devices to handle more complex models and larger datasets more efficiently. This could potentially result in improved overall system performance.
- Avoid catastrophic forgetting: Mitigating the negative effects of Continual learning, such as catastrophic forgetting.
- Reduced resource consumption: AI accelerators are usually a lot more efficient at tasks like that and therefore can help reduce the power consumption on the main processors of IoT devices and better performance stability compared to our findings for real-world FCL.
- More Scalable Approach: With AI accelerators handling the training rounds, it may be possible to scale federated learning to more complicated models without significantly impacting individual performance.

However, it is also crucial to consider the drawbacks:

- Inefficient computer and cost allocation: having an AI accelerator dedicated to training rounds that do not happen very frequently is a waste of money, computational power and space. Those three aspects are usually the most important things to efficiently manage when dealing with IoT.
- Increased complexity: Incorporating AI accelerators into IoT devices adds another layer of complexity to the overall system, which is also something that is usually minimised as much as possible in order to prevent unnecessary points of failures

In conclusion, the idea of using embedded AI accelerators only for federated learning rounds holds promise at first, however, when looking at the drawbacks they do not outweigh the potential benefits, especially considering how important some of the issues are when dealing with IoT.

6 Project discussion and conclusion

6.1 Project outcomes

Throughout this investigation on federated continual learning, we have demonstrated a strong case for various uninterrupted IoT applications. The comparison of FCL and traditional Federated Learning (FL) showed that FCL allows tasks that must remain uninterrupted to have more stable and consistent performance. We have postulated that quite a few scenarios, like healthcare or industrial monitoring systems, could greatly benefit from federated continual learning for its privacy-compliant and non-disruptive nature. Other methods that could eliminate the need for continual learning, like AI accelerators, to offload training tasks. While this approach has potential benefits, such as enhanced performance and faster training, it also raises concerns about inefficient resource allocation, increased costs, and added complexity to IoT devices. Overall, the drawbacks of this alternative seem to outweigh its potential advantages, especially when considering the importance of resource management and simplicity in IoT systems

Furthermore, we have listed several untested benefits of FCL that warrant further exploration, such as its ability to adapt on the fly to changing data distributions, efficient resource utilization, and frequent global model updates. These features can improve the adaptability, efficiency, and personalization of FCL in IoT environments.

6.2 Limitations

During our research, we also highlighted the very important drawbacks of continual learning when it comes to its robustness, catastrophic forgetting is the most important problem with this technique. however, we have also identified ways to mitigate this problem like Elastic Weights Consolidation or Replay method.

A glaring problem of our project is the focus on a single dataset (MNIST) in order to conduct our experiments, this is usually not a good practice as different datasets help to highlight different problems with new techniques. therefore, by focusing on a single dataset we could have missed interesting behaviours from CNNs or MLPs in a Continual learning setting.

We have not explored the possibilities of modifying the hyperparameters of models like the number of neuron layers, the number of neurons or the dropout rate. This is especially a problem in our case since modifying the number of layers or the number of neurons per layer could have had an interesting effect on catastrophic forgetting and might have prevented the unstable performance of continual learning.

Limitations from our research in particular come mainly from the fact that the real-world test was not as representative as it should have been, the object recognition task and the federated learning strategy were two separate workloads instead of being combined like they would in industrial applications. An actual implementation could have highlighted other latency problems from the model being updated by continual learning.

6.3 Future Work

Federated learning, although a relatively new approach, has gained significant attention as more companies prioritize privacy-compliant machine learning techniques. Despite some drawbacks, continual learning has also received considerable interest, particularly as real-world machine learning applications, such as autonomous vehicles and robots, demand the ability to adapt and evolve in constantly changing environments.

When it comes to combining both federated and continual learning, there are very few research papers on the matter, possibly due to their experimental nature and concerns about resource utilization. Nevertheless, we have demonstrated their potential usefulness in certain situations. To establish this combined approach as a viable solution for uninterrupted, privacy-compliant IoT, further research and validation are needed on the following points:

- More experiments with real-world IoT scenarios: To validate the effectiveness of Federated Continual Learning, it is crucial to conduct extensive tests and evaluations in real-world IoT environments to account for factors that may not be present in simulated or controlled settings.
- Improvements on continual learning's robustness: Enhancing the robustness of continual learning methods in the face of changing data distributions and evolving tasks is essential to combat catastrophic forgetting and ensure the reliability of Continual Learning systems.
- Handling device heterogeneity: Federated Continual Learning should be able to accommodate different devices with varying capabilities, resources, and data distributions. Addressing these disparities may pose difficulties in the implementation and optimization of the learning process.
- Hyperparameters tuning: As mentioned in the limitations section, modifying the hyperparameters of the models used in this context could have a drastic impact on the performance and reliability of FCL. It needs to be explored as a way to render FCL more suitable for industrial applications.
- Personalisation: Continual learning adapts the model to the users' data in real-time, which means models from on device to another could diverge quite a lot. Incorporating personalized FL, like model clusterisation, that caters to the specific needs and preferences of individual devices is especially useful in the context of federated continual learning.

7 Conclusion

In this investigation of federated continual learning, we demonstrated its potential for various uninterrupted IoT applications. Our comparison of FCL and traditional Federated Learning (FL) revealed that FCL provides more stable and consistent performance for tasks that must remain uninterrupted. We postulated that numerous scenarios, such as healthcare or industrial monitoring systems, could greatly benefit from federated continual learning due to its privacy-compliant and non-disruptive nature. However, we also identified several limitations in our research, including the robustness of continual learning, the focus on a single dataset (MNIST), the unexplored possibilities of modifying hyperparameters, and the real-world test not being as representative as it should have been.

Future work should focus on conducting more experiments with real-world IoT scenarios, improving continual learning's robustness, handling device heterogeneity, and exploring the impact of hyperparameter tuning. Additionally, incorporating personalized FL, such as model clusterization, can cater to the specific needs and preferences of individual devices in the context of federated continual learning. While our analysis suggests that Federated Continual Learning is a promising approach for uninterrupted IoT applications, further research and experimentation are needed to fully understand its capabilities and limitations and to explore alternative strategies that can enhance the efficiency and reliability of continuous IoT systems.

References

[1] N. Confessore, Cambridge Analytica and Facebook: The Scandal and the Fallout So Far *New York Times*, 04 2018.

[2] A. Ashfahani and M. Pratama, *Autonomous Deep Learning: Continual Learning Approach for Dynamic Environments*, pp. 666–674.

[3] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, Federated Learning of Deep Networks using Model Averaging *CoRR*, vol. abs/1602.05629, 2016.

[4] "MS Windows NT federated learning: Collaborative machine learning without centralized training data." https://ai.googleblog.com/2017/04/federated-learning-collaborative.html. Accessed: 2022-11-25.

[5] S. H. . T. van de Ven, G.M., Brain-inspired replay for continual learning with artificial neural networks *CoRR*, 2020.

[6] H. Shin, J. K. Lee, J. Kim, and J. Kim, Continual Learning with Deep Generative Replay 2017.

[7] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, Continual Lifelong Learning with Neural Networks: A Review *CoRR*, vol. abs/1802.07569, 2018.

[8] J. Kirkpatrick, R. Pascanu, N. C. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, Overcoming catastrophic forgetting in neural networks *CoRR*, vol. abs/1612.00796, 2016.

[9] F. Sattler, K.-R. Müller, and W. Samek, Clustered Federated Learning: Model-Agnostic Distributed Multi-Task Optimization under Privacy Constraints 2019.

[10] Y. Zhou, Q. Ye, and J. Lv, Communication-Efficient Federated Learning With Compensated Overlap-FedAvg *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 1, pp. 192–205, 2022.

[11] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, Federated Learning with Non-IID Data 2018.

[12] S. Han, S. Park, F. Wu, S. Kim, C. Wu, X. Xie, and M. Cha, FedX: Unsupervised Federated Learning with Cross Knowledge Distillation in *Computer Vision – ECCV 2022* (S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, eds.), (Cham), pp. 691–707, Springer Nature Switzerland, 2022.

[13] A. Falzberger, *Energy-efficient IoT by Continual Learning for Data Reduction in Wireless Sensor Networks*. PhD thesis, Wien, 2023.

[14] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, T. Parcollet, and N. D. Lane, Flower: A Friendly Federated Learning Research Framework *CoRR*, vol. abs/2007.14390, 2020.

[15] V. Lomonaco, L. Pellegrini, A. Cossu, A. Carta, G. Graffieti, T. L. Hayes, M. D. Lange, M. Masana, J. Pomponi, G. van de Ven, M. Mundt, Q. She, K. Cooper, J. Forest, E. Belouadah, S. Calderara, G. I. Parisi, F. Cuzzolin, A. Tolias, S. Scardapane, L. Antiga, S. Amhad, A. Popescu, C. Kanan, J. van de Weijer, T. Tuytelaars, D. Bacciu, and D. Maltoni, Avalanche: an End-to-End Library for Continual Learning 2021.

[16] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, PyTorch: An Imperative Style, High-Performance Deep Learning Library 2019.

[17] P. Moritz, R. Nishihara, S. Wang, A. Tumanov, R. Liaw, E. Liang, M. Elibol, Z. Yang, W. Paul, M. I. Jordan, and I. Stoica, Ray: A Distributed Framework for Emerging AI Applications 2018.

[18] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, Federated Learning with Non-IID Data 2018.

[19] A. Usmanova, F. Portet, P. Lalanda, and G. Vega, A distillation-based approach integrating continual learning and federated learning for pervasive services 2021.

[20] J. Yoon, W. Jeong, G. Lee, E. Yang, and S. J. Hwang, Federated Continual Learning with Weighted Inter-client Transfer 2021.

[21] G. M. van de Ven and A. S. Tolias, Three scenarios for continual learning 2019.

[22] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, Overcoming catastrophic forgetting in neural networks *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[23] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[24] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection 2016.

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, SSD: Single Shot MultiBox Detector in *Computer Vision – ECCV 2016*, pp. 21–37, Springer International Publishing, 2016.

8 Appendices

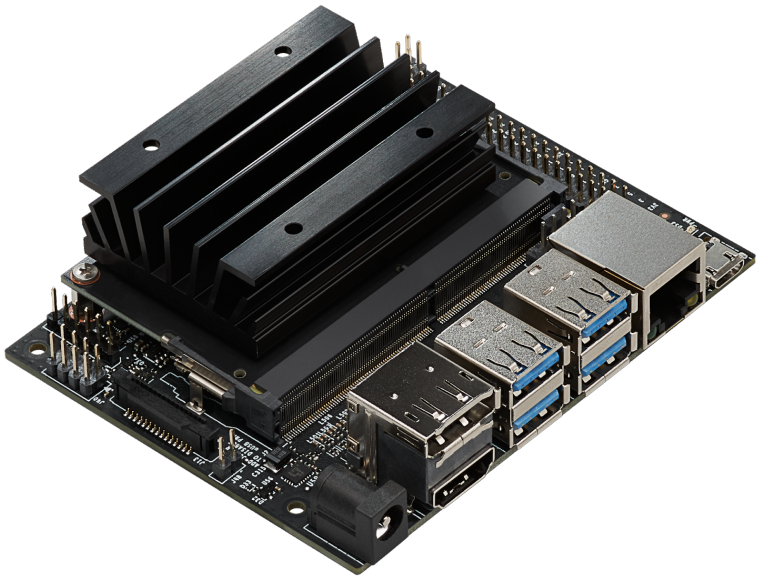

Figure 8: **Nvidia Jetson Nano.**

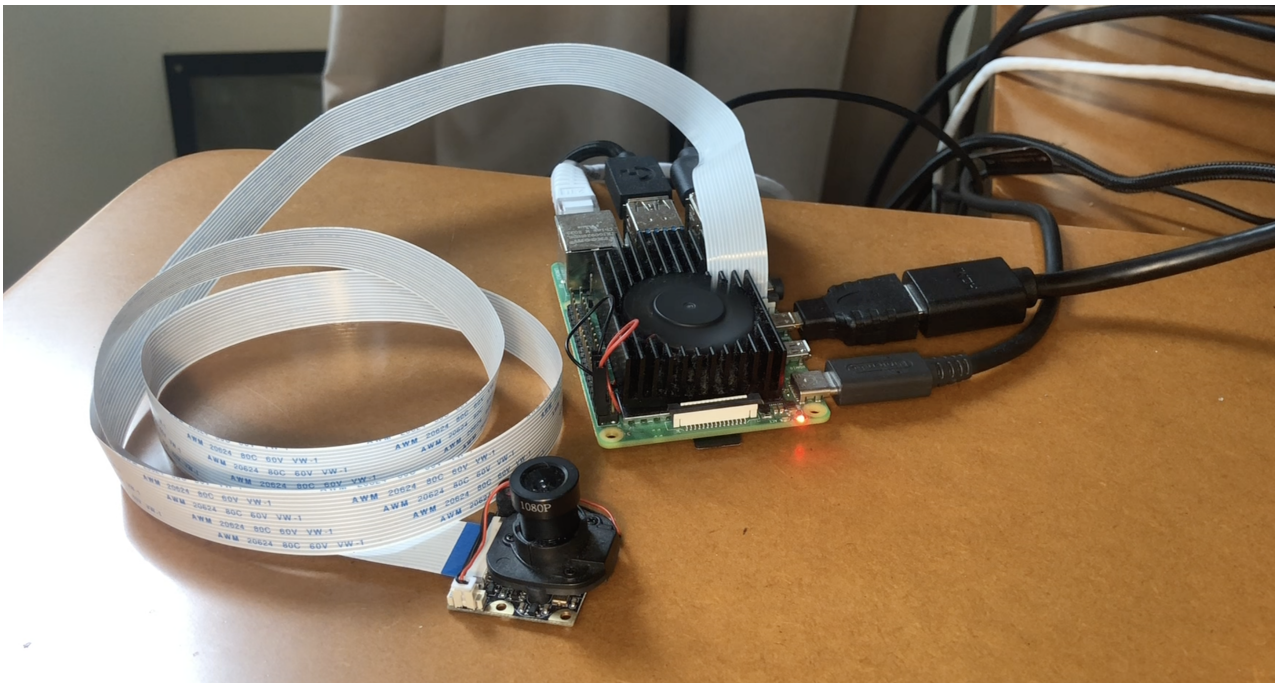

Figure 9: **Raspberry Pi real-world testing**