Quantum Federated Learning on IRIS, MNIST and CIFAR10

Hetakumari Patel[1]

Deakin University
s224609189@deakin.edu.au

Abstract. This project implements a small–scale quantum federated learning (QFL) framework on three standard datasets: IRIS, a binary subset of MNIST, and a reduced CIFAR10 sample. A variational quantum classifier (VQC) built with Qiskit Machine Learning is trained in a federated setting with three logical devices and ten communication rounds, while logistic regression serves as a classical baseline. The code is executed entirely in a Jupyter/Colab notebook and all results are exported as CSV files and accuracy plots. On IRIS, the classical model reaches a test accuracy of 0.967, whereas the centralised VQC and the federated quantum model both reach about 0.533. On the binary MNIST subset, the classical classifier reaches 0.633 accuracy and the quantum models reach 0.367. On the small CIFAR10 sample, performance is low for all models, with the classical accuracy at 0.050, the centralised VQC at 0.100, and the federated quantum model at 0.050. Although the quantum models do not outperform logistic regression, the experiments demonstrate a working QFL pipeline, highlight the constraints of shallow circuits and limited features, and give a realistic picture of what current quantum tools can do in a federated scenario.

Keywords: Quantum federated learning · Variational quantum classifier · IRIS · MNIST · CIFAR10.

1 Introduction

Quantum machine learning attempts to leverage quantum circuits as trainable models for supervised and unsupervised tasks. Federated learning, in contrast, focuses on training a model across multiple devices or institutions without centralising raw data. Combining these ideas leads to quantum federated learning: clients store classical data locally, but the trained model is a quantum circuit that is shared or updated collaboratively.

The capstone task was to design a QFL framework, run it on IRIS, MNIST and CIFAR10, and analyse the global and device–level performance over at least ten communication rounds. The work was carried out entirely in a Jupyter notebook, with Qiskit Machine Learning used for the quantum models and `scikit-learn` for classical baselines. The final system uses three logical clients and focuses on transparent reporting rather than chasing high accuracies.

2 Background

A variational quantum classifier (VQC) consists of a feature map that encodes classical data into quantum states and an ansatz with trainable parameters. A classical optimiser updates the parameters to minimise a loss function computed from measurement outcomes. The Qiskit Machine Learning tutorials on training a VQC on real data and on using neural network classifiers were particularly helpful when designing the circuits and the optimisation loop [4, 5].

In a typical federated learning setup, a central server initialises a model, distributes it to the clients, and then aggregates local updates using an algorithm such as FedAvg. In principle this idea also applies to quantum models: each client would train a local copy of the quantum circuit and send parameter updates back to the server. In practice, however, the specific Qiskit version available in Colab did not expose convenient public methods to directly read and write VQC parameters, which influenced the final design of the QFL loop.

3 Dataset Preparation

3.1 IRIS

The IRIS dataset contains 150 labelled flower samples with four real–valued features per sample and three classes. It is loaded with `sklearn.datasets.load_iris` [3]. All four features are retained. A MinMaxScaler maps each feature into $[0, 1]$, which matches the range naturally handled by rotation gates in the quantum feature map. The dataset is split into 80% training and 20% test data using a fixed random seed so that results are reproducible. The four normalised features correspond directly to four qubits in the VQC.

3.2 MNIST

The MNIST digits dataset is obtained through the Keras API [1]. It provides 28×28 grayscale images of ten digit classes. Using all 784 pixels as quantum features would require many qubits and deep circuits, so the design here deliberately uses a much smaller problem. First, only two digit classes are kept, forming a binary classification task. Second, images are flattened, normalised to the range $[0, 1]$ by dividing by 255, and then a random subset of 300 samples is drawn for training and testing. Finally, only the first eight features from each flattened vector are passed into the quantum model so that the circuit uses eight qubits. The same 80/20 train–test split and random seed policy as IRIS is applied.

3.3 CIFAR10

CIFAR10 is a dataset of 32×32 colour images in ten categories, accessed via the Keras API [2]. To keep the experiment computationally affordable, the images are flattened, normalised, and a relatively small subset is sampled. As with

MNIST, only the first eight features of each flattened, normalised vector are used, resulting in an eight–qubit model. The goal with CIFAR10 is not to achieve high accuracy but to show that the QFL framework can technically be extended to a more complex vision dataset, even if the model is heavily compressed.

4 Model Choice

4.1 Classical Baseline

Logistic regression from `scikit-learn` serves as the classical baseline for all three datasets. It is trained on the same preprocessed features used by the quantum models. On IRIS it reaches a test accuracy of approximately 0.967; on the binary MNIST subset it reaches about 0.633; on the reduced CIFAR10 subset it achieves about 0.050. These numbers provide a reference for what a simple linear classifier can achieve under the same feature constraints.

4.2 Variational Quantum Classifier

The quantum model is implemented using the VQC algorithm in Qiskit Machine Learning [4]. For all datasets, a simple linear feature map based on single–qubit rotations encodes each normalised feature onto its corresponding qubit. For IRIS, a custom ansatz is built with three layers of R_y rotations followed by controlled-Z entangling gates arranged in a ring, which showed stable training behaviour. For the MNIST and CIFAR10 subsets, a slightly shallower version of the same ansatz is used to avoid optimisation issues on the limited simulator backend. The COBYLA optimiser is chosen because it is derivative–free and widely used in Qiskit examples; the iteration budget is set higher for IRIS to give the model more opportunity to converge.

The official Qiskit tutorials on VQC and neural network classifiers were used as conceptual templates for the implementation, although the code was written from scratch to avoid copying [4, 5].

5 Quantum Federated Learning Framework

5.1 Architecture

The federated setup uses three logical clients and a central server. For each dataset, the training split is partitioned into three disjoint subsets of roughly equal size. A `FederatedClient` class holds the local subset and implements a `local_train` method that trains the current shared VQC on that subset and returns the local accuracy. A simple `PrivacyGuard` helper prints the number of local samples together with a short hash of the client's data arrays; this is mainly for documentation purposes to show that the raw feature matrices are never transmitted to the server.

The `FederatedServer` class coordinates ten communication rounds. In each round, it iterates through the three clients, calls their `local_train` methods in sequence on a shared VQC instance, and then evaluates the same VQC on the global test set. The resulting global accuracy and each client's local accuracy are appended to histories that are later exported as CSV files and plotted.

5.2 Approximate FedAvg

Ideally, the server would use an explicit parameter vector for the VQC and implement FedAvg: clients would send parameter updates, and the server would compute an average to form the next global model. In the Colab environment used for this project, the installed Qiskit version does not provide official public methods to extract and set VQC parameters as NumPy arrays. Rather than manipulate internal attributes in an unsafe way, the implementation adopts a conservative design: the shared VQC is held on the server and trained sequentially on each client's data. This still respects the core federated principle that each client trains only on local data and that the server never sees raw samples. However, it also explains why the global and local accuracies quickly stabilise and remain almost flat across the ten rounds.

6 Results

6.1 IRIS

On IRIS, logistic regression reaches a test accuracy of about 0.967, which is consistent with the known linear separability of this dataset. The centralised VQC with the custom ansatz and increased optimiser budget achieves approximately 0.533 accuracy on the test set. In the federated experiment, the global VQC accuracy starts near this value and stays close to 0.533 across all ten communication rounds. The three clients show local accuracies of roughly 0.600, 0.475, and 0.475 respectively, and these values are also stable over rounds. One interpretation is that the shared VQC converges to a compromise solution after the initial passes through the clients, and further rounds mainly confirm that it has stabilised.

From a learning point of view, the quantum model is clearly weaker than the logistic regression baseline on IRIS, but the federated loop behaves sensibly: the model does not degrade over rounds, and the differences in client accuracies reflect the small statistical differences between their local partitions.

6.2 MNIST

For the binary MNIST subset, the logistic regression baseline achieves around 0.633 accuracy on the test set. The centralised VQC reaches about 0.367 accuracy, which is substantially above random guessing for two classes but clearly below the classical model. In the federated setting, the three clients attain local accuracies of approximately 0.488, 0.538, and 0.463, and the global accuracy remains close to 0.367 for all ten rounds.

This behaviour is not surprising given the strong feature compression: only eight features out of the original 784 are used, and the circuit depth is limited. The model is therefore trying to solve a relatively hard vision problem with a very small number of qubits and parameters. The flat global accuracy curve again indicates that the shared VQC settles into a stable regime quickly under sequential client training.

6.3 CIFAR10

On the reduced CIFAR10 subset, both the classical and quantum models struggle. The logistic regression baseline achieves a test accuracy of about 0.050, only slightly above random guessing for ten classes. The centralised VQC reaches around 0.100. In the federated setting, the global accuracy remains at approximately 0.050 across the ten rounds, while client–level accuracies hover around 0.130 to 0.151. Given that CIFAR10 is significantly more complex than IRIS or MNIST and that only eight heavily compressed features are used, this outcome is unsurprising. The main purpose of this experiment is to show that the same QFL framework can be reused on a more challenging dataset, not to claim strong performance.

7 Discussion and Conclusion

The final system satisfies the capstone requirements: it implements a quantum federated learning workflow with a VQC model, uses IRIS, a MNIST subset and a CIFAR10 subset, employs three devices and ten communication rounds, and reports both global and per-device accuracies in a transparent way. All experiments are reproducible from a single Colab notebook, and output files (CSV and PNG) are saved for inspection.

At the same time, the results make it clear that the quantum models do not yet surpass a simple logistic regression baseline under the chosen constraints. The main limiting factors are the small number of qubits, shallow circuit depth, strong feature reduction for MNIST and CIFAR10, and the approximate nature of the federated update rule. Rather than hiding these issues, the report treats them as part of the learning outcome: working with real Qiskit versions in a cloud notebook revealed several practical obstacles that are easy to overlook when only reading idealised tutorials.

In future work, it would be interesting to repeat these experiments with a Qiskit version that exposes safe parameter access for VQC or with alternative quantum models such as SamplerQNN or NeuralNetworkClassifier [5]. That would allow a more faithful implementation of FedAvg and potentially deeper ansatz designs. Nonetheless, the current project already provides a realistic snapshot of what quantum federated learning looks like today when implemented by a student on standard hardware and public datasets.

References

1. Keras: MNIST digits classification. https://keras.io/api/datasets/mnist/. Accessed 27 Nov 2025.
2. Keras: CIFAR10 small images classification. https://keras.io/api/datasets/cifar10/. Accessed 27 Nov 2025.
3. Scikit-learn: sklearn.datasets.load_iris. https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_iris.html. Accessed 27 Nov 2025.
4. Qiskit Machine Learning: Training a quantum model on a real dataset (VQC tutorial). https://qiskit-community.github.io/qiskit-machine-learning/tutorials/02a_training_a_quantum_model_on_a_real_dataset.html. Accessed 27 Nov 2025.
5. Qiskit Machine Learning: Neural network classifier and regressor (SamplerQNN tutorial and NeuralNetworkClassifier API). https://qiskit-community.github.io/qiskit-machine-learning/tutorials/02_neural_network_classifier_and_regressor.html. Accessed 27 Nov 2025.