

TCS Quantum Challenge: Solution Summary for Challenge 4 - Forecasting of FTSE100 Index

Gajendra Malviya and Aakash Swami

Data & Decision Sciences, TCS Research

Quantum presents a new paradigm for computing that promises a transformation in quantitative finance problem-solving. It has been observed to provide improvements over current approaches in terms of speed, and accuracy, etc., in AI/ML, optimization, and simulation. Predicting the prices of assets, influenced by various macro and micro factors and market dynamics, is a fundamental challenge in finance. While several approaches, which use classical methods, have been proposed for price prediction, leveraging quantum techniques holds the promise of further enhancement. This study aims to predict the price of the FTSE 100 index based on historical data, comparing the effectiveness of quantum and classical approaches.

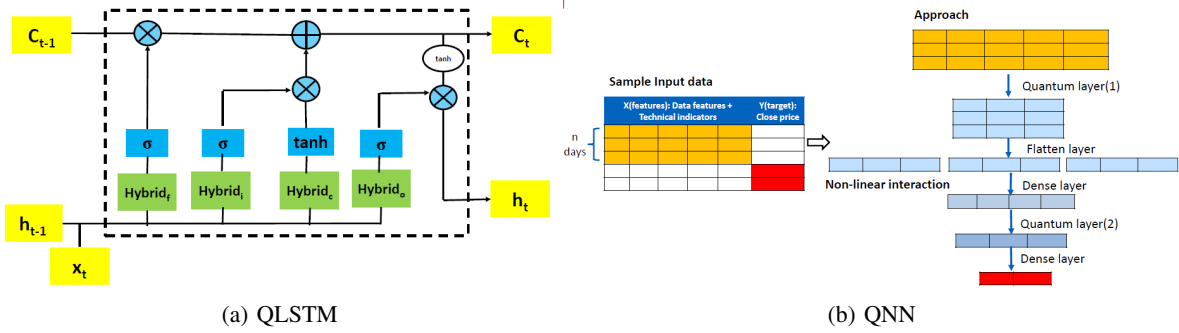


Fig. 1: Quantum approach

In this study, we first prepared the data, which involved several steps - such as generating side features, interpolation, and data normalization. We considered both exogenous features, including GBP USD, gold price, crude futures, and libor rate, and various technical indicators such as simple moving average, moving averages convergence divergences, Bollinger Bands, and exponential moving average. Then, we employed two quantum approaches for stock price prediction as shown in Figure 1. The first quantum approach, Quantum Long Short-Term Memory (QLSTM) [1], extends the classical LSTM by replacing the neural networks in its gates - namely input gates, forget gates, and output gates - with a 'hybrid' circuit. This 'hybrid' circuit consists of a stack of dense neural and quantum circuit layers. Secondly, we propose a novel quantum neural network approach (QNN). This approach involves passing 'n' days of feature data through the quantum circuit to generate latent factors for each day. Subsequently, a flattening layer is applied to these factors and arranged in a sequence. Finally, the nonlinear interaction between these latent factors for the last 'n' days is captured through a stack of dense neural and quantum layers to predict the future.

To model quantum circuits in both the QLSTM and QNN, we utilize multi-qubit data reuploading (DRC) [2] and demonstrate its significance. Implementing multi-qubit DRC involves partitioning input data features X into sets X_1, X_2 , etc., each with a dimension of three. These sets are then encoded into multiple qubits using single-qubit arbitrary rotation, followed by the

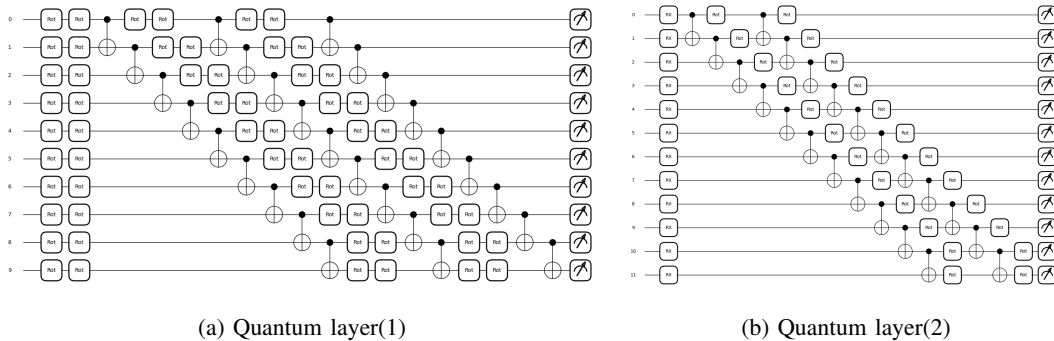


Fig. 2: Quantum circuit: QNN

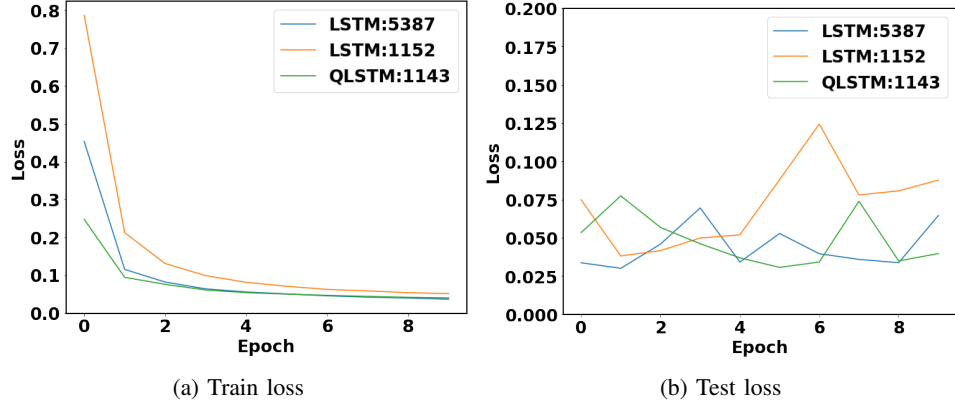


Fig. 3: LSTM and QLSTM: Effect of the number of learning parameters on train and test loss

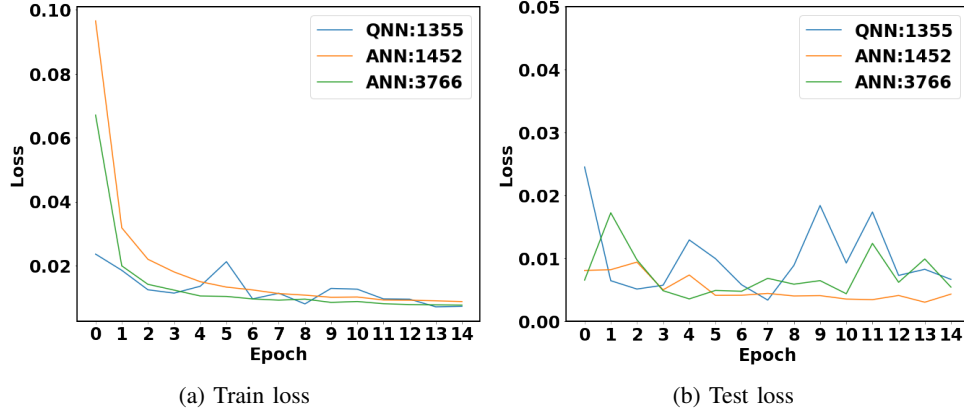


Fig. 4: ANN and QNN: Effect of the number of learning parameters on train and test loss

addition of a parametrized quantum layer using single-qubit arbitrary rotation, and subsequently, an entanglement layer is added. By iteratively stacking data uploading, parametrized gates, and entangling layers, a highly complex feature can be generated, enhancing the learning capacity of the algorithm. Finally, the expectation value of the Pauli Z operator is measured for all the qubits. The quantum circuit employed in QNN is illustrated in Figure 2. The quantum circuit utilized in QLTM is the same as Figure 2(a) but with 6 qubits and 1 layer.

In the current experimentation, we utilized the ‘default.qubit’ simulator available in PennyLane. We used 96% of the data for training (approx. 2647 days) and 4% for testing (approx. 111 days). We implemented the QNN using Keras and PennyLane, and coded the QLSTM using PyTorch and PennyLane by referring to the work in [3]. The effect of data reuploading was analyzed for both 4-qubit and 6-qubit quantum circuits when using QLSTM. Details on the comparison of QLSTM and QNN with their classical counterpart are provided in detail in the extended report available at the GitHub link [4], [5].

An analysis of the results indicates that data reuploading aids in better learning, as the training loss decreases more for a quantum circuit with DRC than for a quantum circuit without DRC after a certain epoch. Consequently, experimental results with QLSTM demonstrate that the 6-qubit QLSTM can provide a comparable solution to the classical LSTM while requiring fewer learnable parameters (1143 compared to 5387), as shown in Table I. Figure 3 shows the training and test loss change for both LSTM and QLTM for different numbers of learning parameters indicated in the legend. We used a learning rate of 0.001 to train the LSTM and a learning rate of 0.05 to train the QLSTM. Additionally, the loss with QLSTM is lower for the same number of learning parameters. Experimentation with QNN reveals that QNN provides comparable results to its classical counterpart and requires fewer computational parameters (1355 compared to 3766), as shown in Table II. Figure 4 shows the training and test loss change for both ANN and QNN for different numbers of learning parameters indicated in the legend. We used a learning rate of 0.001 to train the ANN and a learning rate of 0.005 to train the QNN. The lower test loss with ANN, with 1452 parameters, is attributed to a larger test dataset, but this aspect warrants further exploration. Furthermore, QNN learns more information in the initial few epochs compared to the classical approach. It is noteworthy that we attempted to use a higher learning rate in the classical model, but the results were not encouraging. Hence, the learning rate, as a hyperparameter, needs to be tuned for both classical and quantum models separately.

As part of the competition, we were asked to predict the FTSE closing prices for the period from March 11th to March 15th to evaluate our approach [6]. We used Mean absolute error (MAE) and Root mean square error (RMSE) as evaluation

TABLE I: Comparison between classical LSTM and QLSTM

Models	Train MAE	Train RMSE	Test MAE	Test RMSE	Parameters
LSTM	74.93	97.74	109.78	130.50	5387
QLSTM	73.81	97.08	76.87	98.09	1143

TABLE II: Comparison between classical ANN and QNN

Models	Train MAE	Train RMSE	Test MAE	Test RMSE	Parameters
ANN	88.03	104.84	92.37	100.28	3766
QNN	97.63	107.87	104.39	110.72	1355

metrics. Table III presents the final results of the predictions for this period. We only present quantum results for QLSTM using a 6-qubit DRC circuit. We got an MAE of 46.34 and RMSE of 47.47 when using QNN, MAE of 42.55 and RMSE of 49.76 when using ANN, MAE of 86.05 and RMSE of 96.07 when using QLSTM, and MAE of 114.73 and RMSE of 121.86 when using classical LSTM. Considering the fact that the lower the loss the better it is, we got appreciable accuracy in terms of MAE and RMSE when using the QNN. It should be noted that the accuracy can be further improved by tuning hyperparameters related to the models.

TABLE III: Prediction from 11th to 15th March

Model	8 th Mar	11 th Mar	12 th Mar	13 th Mar	14 th Mar	15 th Mar
QNN	7701.74	7700.10	7708.77	7712.89	7688.78	7683.45
ANN	7715.72	7713.12	7666.08	7766.65	7769.66	7676.48
LSTM	7613.73	7622.44	7597.53	7610.93	7623.29	7636.1094
QLSTM	7662.46	7662.88	7645.06	7640.86	7660.05	7624.82

Thus, the performance of the quantum approach with fewer learning parameters (low qubit-count and low overall depth of circuits) demonstrates its advantage in recognizing patterns in data or making predictions. The Quantum approach learns more information in the initial few epochs compared to the classical approach. Quantum performance could improve with increasing quantum circuit depth and the number of qubits.

The challenge statement, extended solution-approach document, final defence presentation deck, code base of our solution, and final test results of our solution can be found at: <https://github.com/rbanerjee7/04360111/tree/main/Phase-2>

REFERENCES

- [1] Chen, Samuel Yen-Chi, Shinjae Yoo, and Yao-Lung L. Fang. "Quantum long short-term memory." In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8622-8626. IEEE, 2022.
- [2] Pérez-Salinas, Adrián, Alba Cervera-Lierta, Elies Gil-Fuster, and José I. Latorre. "Data re-uploading for a universal quantum classifier." *Quantum* 4 (2020): 226.
- [3] DikshantDulal. (Year). SoftServe_QLSTM. Retrieved from https://github.com/DikshantDulal/SoftServe_QLSTM/tree/main
- [4] <https://github.com/rbanerjee7/04360111/tree/main/Phase-2/Classical%20method>
- [5] https://github.com/rbanerjee7/04360111/blob/main/Phase-2/Challenge_4_report_04360111.pdf
- [6] https://github.com/rbanerjee7/04360111/tree/main/Phase-2/Final_Prediction