Journal of Scientific & Industrial Research
Vol. 85, July 2026, pp. 602-609
DOI: 10.56042/jsir.v85i7.29797

# Hybrid Quantum-Classical LSTM for Daily Temperature Forecasting: A Case Study on Delhi Climate Data

Vani A Yelamali*[1], Ritik Sunil Jain[2], S P Rajamohana[3], Karthiganesh Durai[2], Manjunath K Vanahalli[1], Prabhu Prasad B M[1] & Kamal Pandey[4]

[1]Department of Computer Science and Engineering, Indian Institute of Information Technology Dharwad 580 009, Karnataka, India

[2]KwantumG Research Labs Pvt Ltd, Karnataka, 560 064, Bangalore, India

[3]Department of Computer science, School of Engineering & Technology, Pondicherry University, 609 605, Karaikal, Puducherry, India

[4]Indian Institute of Remote Sensing, 248 001, Dehradun, India

*Received 17 March 2026; revised 27 July 2026; accepted 28 September 2026*

Agriculture, public health, energy management, and urban planning all depend on accurate daily temperature forecasts. Using the Delhi climate dataset, the study explores a hybrid quantum-classical Long Short-Term Memory (LSTM) model for multivariate daily temperature prediction. The best-performing classical model is then contrasted with the hybrid Quantum LSTM architecture, which incorporates Variational Quantum Circuits (VQC) into the LSTM's gating mechanisms, following an evaluation of the classical recurrent baseline models, namely RNN, GRU, and LSTM. The model forecasts the mean temperature of the next day using meteorological data collected over a ten-day period. By lowering MAE from 1.522 to 1.466, MSE from 3.595 to 3.266, and RMSE from 1.896 to 1.807 while increasing $R^2$ from 0.910 to 0.919, the experimental results demonstrate that the suggested Quantum LSTM offers slight but steady improvements over the classical LSTM. These results suggest that hybrid quantum-classical recurrent models can estimate temperatures with slightly higher predictive performance than the classical model. The reported performance advantages, however, are small and should be viewed more as a preliminary case study than as definitive evidence of quantum advantage. As a result, these results continue to have a narrow scope and emphasize the necessity of validation using bigger and more varied meteorological datasets.

**Keywords:** Graph neural network, Hybrid QLSTM, Quantum computing, Quantum machine learning, Variational quantum circuits

## Introduction

Weather prediction is essential for multiple decision-making applications such as agriculture, public health, energy management, and city planning. Accurate temperature forecasting is very desirable in large and densely populated urban locations like Delhi since it directly impacts public safety and environmental sustainability. The urban heat island effect, rapid urbanization, industrial growth, and traffic pollution all contribute to Delhi's highly fluctuating climate.[1] Such complex elements in local atmospheric dynamics make the temperature forecast difficult. Traditional numerical and statistical forecasting approaches are still widely used in climate research. They have a limited capacity to describe the nonlinear, chaotic and multiscale dynamics that are characteristic of atmospheric processes.[2] This limitation has motivated the exploration of Quantum Machine Learning (QML) as a new paradigm with higher representational capacity and flexibility for the modelling of complex climate systems[3,4], in addition to the existing data-driven paradigms, such as Machine Learning (ML) and Deep Learning (DL). The evaluation of the climate time series is a particularly challenging task since the meteorological variables are strongly inter-connected and their temporal behaviour are non-stationary.[2] Traditional computing methods generally fail to represent long-term temporal dependencies. Therefore, more advanced learning architectures are designed to be able to better capture complicated spatio-temporal relationships. In this context, deep learning architectures have shown a quite promising performance. CNNs are good at feature extraction, but recurrent models like LSTMs and GRUs are better at

*Author for Correspondence
E-mail: yelamalivani@gmail.com, vani.23phdcs02@iiitdwd.ac.in

learning temporal relationships. The combination of these complementary qualities enabled optimal forecasting by the collaborative modelling of spatial and temporal information[3,5] using hybrid CNN-LSTM architectures.

However, many challenges still exist, particularly in the recognition of long-term dependencies and the preservation of accuracy across lengthy forecasting horizons. These limitations motivated further exploration into recurrent neural network variations, Graph Neural Networks (GNNs), Transformer-based models, and hybrid deep learning architectures.[5] Recent advances in ML and DL have resulted in a greater diversity of data-driven approaches to temperature prediction. Models like Artificial Neural Networks (ANNs), CNNs, RNNs, LSTMs and GRUs have shown the ability to find novel temporal structures and nonlinear correlations in meteorological data.[5,6] These approaches typically beat standard linear models such as Autoregressive Integrated Moving Average (ARIMA) model in terms of predicted accuracy and generalization ability. Also, the integration of optimization and meta learning approaches such as Variational Mode Decomposition (VMD) and Genetic Algorithms (GA) has increased the model robustness and convergence efficiency, leading to more accurate forecasting outcomes.[5,6]

The combination of Quantum Computing and Machine Learning offers the new option for climate data analysis. QML is particularly suited for modelling complicated climatic time series, as it can handle high dimensional data in parallel and allow for more expressive feature representations through quantum notions such as superposition and entanglement.[4,7] Quantum-inspired models such as Quantum Neural Networks (QNNs) and hybrid Quantum Convolutional Neural Networks (HQ-CNNs) have exhibited better forecasting accuracy and computing efficiency in temperature prediction tasks.[6] Finally, models can increase long-term forecasting accuracy by using Variational Quantum Circuits (VQCs) to find exact correlations that are frequently hard to classical models to identify.

Furthermore, there has been a rising interest in enhancing traditional statistical models with hybrid frameworks that integrate deep learning or quantum-inspired methods. A successful combination of linear and nonlinear modelling capabilities[5] enhances prediction accuracy and interpretability through methods such as ARIMA-LSTM and ARIMA-GRU. These methods are hybrid architectures that combine domain expertise and data-driven learning, providing greater robustness and predictive capacity, according to recent studies in DL for climate applications. In line with international goals for climate resilience and sustainable development, these advancements represent a more comprehensive shift away from strictly deterministic, physics-based forecasting toward probabilistic and quantum-enhanced data-driven models.[1,4]

In this regard, the current study examines whether daily mean temperature forecasting on the Delhi climate dataset may be enhanced by a hybrid quantum-classical LSTM. In order to determine the strongest baseline, classical recurrent models such as RNN, GRU, and LSTM are first assessed. Next, a hybrid QLSTM is directly contrasted with the top-performing classical model. The main contributions of this study are:

- The development of a QLSTM architecture that incorporates variational quantum circuits into recurrent gating operations.
- Evaluating classical and hybrid recurrent models for multivariate temperature forecasting.
- An empirical investigation indicates that proposed QLSTM exhibits competitive performance and marginally surpasses results on the Delhi dataset. The objective of the study is to present an initial, practical case analysis of urban temperature forecasting instead of asserting a generalizable conclusion regarding scalability in all climate prediction endeavours.

## Materials and Methods

### Data Description

The research employs the Delhi weather dataset as its experimental dataset, which is publicly available on Kaggle.[8] The dataset includes daily weather recordings from 2013 to 24 April 2017. Each record contains meteorological characteristics, like mean temperature, humidity, wind speed, and mean pressure. The dataset is divided into train and test sets. The training set has 1,461 samples while the testing set contains 114 samples. Sample of the training set is illustrated in Table 1.

### Data Pre-processing

The pre-processing phase involves three primary steps: data transformation, data specification and data cleansing. The data were cleaned by removing null

Table 1 — Sample Delhi weather data

| SL. No | date | meantemp | humidity | wind_speed | meanpressure |
|---|---|---|---|---|---|
| 0 | 2013-01-01 | 10.000000 | 84.500000 | 0.000000 | 1015.666667 |
| 1 | 2013-01-02 | 7.400000 | 92.000000 | 2.980000 | 1017.800000 |
| 2 | 2013-01-03 | 7.166667 | 87.000000 | 4.633333 | 1018.666667 |
| …. | …. | …. | …. | …. | …. |
| 1460 | 2016-12-31 | 15.052632 | 87.000000 | 7.325000 | 1016.100000 |
| 1461 | 2017-01-01 | 15.913043 | 85.869565 | 2.743478 | 59.0 |

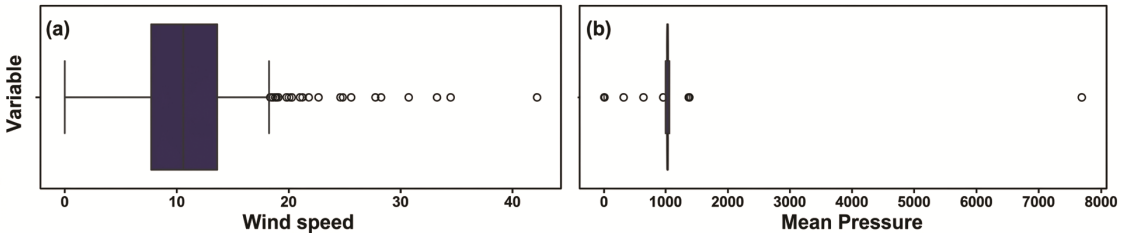


Fig. 1 — Outliers in: (a) wind speed and (b) mean pressure

values and checking for missing values. To avoid data leaking between the training and testing stages, the last item on 1 January 2017 is excluded since it belongs to the test dataset. Furthermore, an outlier analysis is done which shows the outliers in the wind_speed and meanpressure variables as shown in Fig. 1. For instance, the raw data record for January 1, 2017, contains a physically anomalous mean pressure value of 59.0 (Table 1). To prevent such raw data errors from skewing model training, these outliers are substituted with the median values of their corresponding features to stabilize the data distribution.

Each feature is then independently normalized using the MinMaxScaler() to scale its value to the range of 0 and 1. Employing separate scalers ensured that the scaling process did not introduce inter-feature dependencies. Since both LSTM and hybrid QLSTM architectures are designed to capture temporal patterns, the data are segmented into sequential 10-day windows. For each sequence, the values of four input features from the preceding 10 days are used to predict the mean temperature on the following day. Subsequently, the training and testing datasets are reshaped into the format (number of samples, number of features, sequence length) to match the input requirement of the models. Finally, the processed NumPy arrays are converted into PyTorch tensors to facilitate efficient model training and evaluation.

**Data Visualization**

Data visualization was performed to illustrate the temporal trends of the meteorological variables over the four-year period, as shown in Fig. 2. A visual

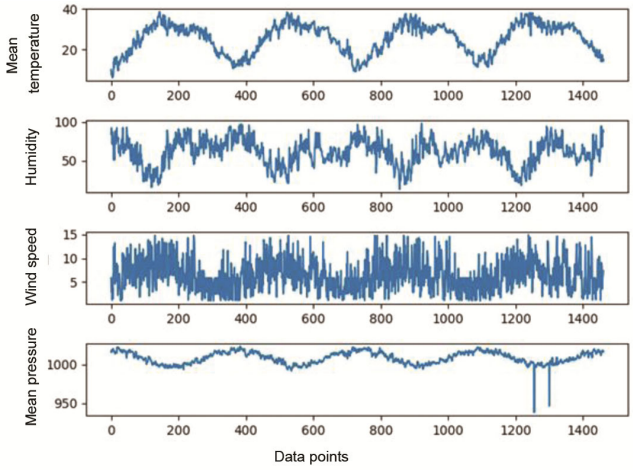


Fig. 2 — Data visualization of meteorological variables

analysis of these graphs reveals distinct time trends among the characteristics. The mean temperature has a marked and consistently regular seasonal cyclical pattern, reflecting expected annual climatic fluctuations. Humidity similarly demonstrates seasonal fluctuations but is characterized by significantly higher short-term variance. On the other hand, wind speed has a far more unpredictable, high-frequency distribution with no obvious long-term cyclical patterns. Ultimately, the average atmospheric pressure exhibits only a handful of discrete fluctuations and minor seasonal changes. The varied behaviours, ranging from stable seasonal trends to unpredictable, non-linear fluctuations, highlight the complex multi-scale dynamics of the climate dataset, thereby justifying the use of advanced sequence-modelling techniques like LSTMs to effectively capture both long-term dependencies and short-lived anomalies.

## Methodology

### Long Short-Term Memory Model (LSTM)

Hochreiter & Schmidhuber developed LSTM networks in 1997 to overcome the limitations of conventional RNNs, giving an enhanced architecture for building long-term dependencies.[9] Traditional RNNs tend to forget information over long sequences. LSTM overcome this constraint by proposing a particular memory cell and three gating techniques (input gate, forget gate, and output gate) to manage the storage, updating and retrieval of information over time.

At time-step $t$, the LSTM evaluates the present input vector $x_t$ along with the prior hidden state $h_{t-1}$ and cell state $c_{t-1}$ from the preceding time step. The sequence of computations performed as follows: The forget gate determines which elements from the prior cell state need to be eliminated.

$$f_t = \sigma\left(W_f[h_{t-1}, x_t] + b_f\right) \quad \ldots (1)$$

The input gate determines the volume of newly acquired information to be added to the memory cell.

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$$

$$\widetilde{c_t} = tanh(W_c[h_{t-1}, x_t] + b_c) \quad \ldots (2)$$

The updated cell state is a mix of the information from the prior cell state that is stored and the information that is obtained newly.

$$c_t = f_t \odot c_{t-1} + i_t \odot \widetilde{c_t} \quad \ldots (3)$$

The output gate regulates the portion of the cell state that is propagated to the next layer or the subsequent time step.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad \ldots (4)$$

$$h_t = o_t(c_t) \quad \ldots (5)$$

where, $\sigma$ denotes the sigmoid activation function, *tanh* represents the hyperbolic tangent, $W$ and $b$ correspond to the learnable weights and biases, and $\odot$ indicates element-wise multiplication. At each time step, the LSTM strategically retains, modifies, or disregards information through these gating mechanisms, which enhances its remarkable performance in fields like as natural language processing, time series prediction, and speech recognition. For the classical implementation, the PyTorch nn.LSTM layer was employed to efficiently model sequential dependencies and extract features for regression-based prediction tasks.

### Hybrid Quantum LSTM

The proposed hybrid QLSTM model is implemented using the PennyLane framework, which facilitates the integration of parameterized quantum circuits with classical deep learning models within a unified optimization environment. Within this framework, the qml.QNode interface enables the definition and execution of quantum computations, while qml.qnn.TorchLayer provides compatibility with PyTorch, allowing quantum circuits to be embedded as trainable layers within classical neural network architectures. This interoperability enables the development of hybrid quantum-classical models suitable for sequential learning applications. In the proposed design, hybrid QLSTM model expands the typical LSTM architecture by replacing the classical forget, input, update and output gates with parameterized quantum circuits, implemented as Variational Quantum Circuits (VQCs). A separate VQC is dedicated to each gate to control the transformation of the input state by a set of quantum operations. These circuits usually involve rotation gates, such as RY, and entangling operations with controlled-NOT (CNOT) gates, allowing the quantum state to undergo a structured parameterized transformation.

The hybrid QLSTM framework exploits quantum phenomena like superposition and entanglement to improve its capacity to describe complex non-linear interactions as an alternative to the conventional affine transformations used within LSTM gating operations. The general architecture is a hybrid quantum-classical scheme where conventional layers are employed for input pre-processing and sequence management, and quantum circuits are inserted into each gating unit for processing intermediate representations. As shown in Fig. 3, the final predictions are produced by passing the resulting quantum outputs through classical post-processing layers. This hybrid architecture provides a compact yet expressive representation to model the temporal dependencies of sequential data. The inclusion of

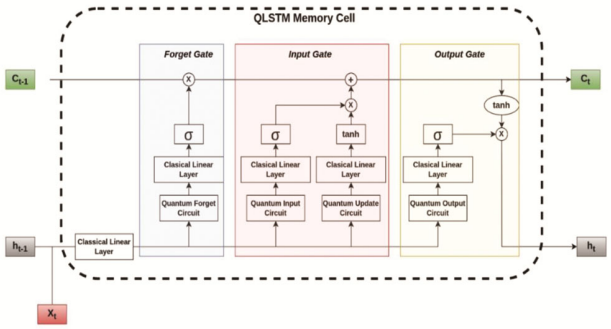


Fig. 3 — Architecture of the Proposed Hybrid Quantum Long Short-Term Memory (QLSTM) Model

parameterized quantum transformations in the recurrent framework may improve the description of complex feature interactions, and can boost the learning efficiency and convergence in solving complicated time-series forecasting problems.

At each time-step $t$, a combined vector is formed by concatenating the previous hidden state $h_{t-1}$ with the current input vector $x_t$:

$$v_t = [x_t, h_{t-1}] \quad \ldots (6)$$

This vector is then mapped to the required number of qubits ($n_q$) by passing it through a classical linear layer $clayer_{in}$, as illustrated in Fig. 3.

- If dim $(v_t) > n_q$, the layer compresses the input to preserve only the most salient features.
- If dim $(v_t) < n_q$, the layer expands the vector into a higher-dimensional representation.

This pre-processing step optimizes the use of quantum resources by minimizing information loss while ensuring dimensional compatibility with the quantum circuit.

The next component of the proposed architecture is the variational quantum circuit (VQC) embedded within each LSTM gating unit, namely the forget, input, and output gates. In this quantum-enhanced formulation, the conventional linear transformations of the classical LSTM are replaced by VQCs. The forget gate

$$f_t = \sigma\big(VQC_i(v_t)\big) \quad \ldots (7)$$

regulates the proportion of the previous cell state $c_{t-1}$ to be retained, while the input gate

$$i_t = \sigma\big(VQC_i(v_t)\big) \quad \ldots (8)$$

controls the amount of new information to be written into the cell state.

The update (candidate) gate generates the candidate information used to update the cell state and is defined as:

$$g_t = tanh\left(VQC_g(v_t)\right) \quad \ldots (9)$$

similarly, the output gate

$$o_t = \sigma\big(VQC_o(v_t)\big) \quad \ldots (10)$$

determines the extent to which the updated cell state is exposed to the hidden state output.

Within each VQC, the classical feature vector $v_t$, obtained from the preceding linear layer, is encoded into a quantum state using angle embedding. Each feature value corresponds to the rotation angle of an individual qubit via a parameterized rotation around the Y-axis, as depicted in Fig. 4. The corresponding single-qubit rotation operation is defined as

$$RY(\theta) = e^{-i\frac{\theta}{2}Y} \quad \ldots (11)$$

where, the Pauli-$Y$ matrix:

$Y = |0 - ii0|$

For a system of $n_q$ qubits, the full embedding operation prepares the quantum state

$$|\Psi\rangle = RY(\theta_1) \otimes RY(\theta_2) \otimes \ldots\ldots \otimes RY\left(\theta_{n_q}\right) | 0\rangle \otimes^{n_q} \quad \ldots (12)$$

placing the qubits into a superposition that encodes the classical features into quantum amplitudes and phases.

Next, entanglement layers are applied to capture dependencies and correlations among the encoded features. Controlled-NOT (CNOT) gates enable interactions between neighbouring qubits and are arranged in a structured topology, such as a linear chain or ring configuration. The CNOT operation is represented as

CNOT = [1000010000010010]

which introduces quantum correlations analogous to shared weights and hidden unit interactions in classical neural networks.

Finally, the quantum state is projected back into the classical domain through measurement**.** The expectation values of the Pauli-$Z$ operator are computed for each qubit to produce classical outputs:

$$\langle Z_i \rangle = \langle \Psi | Z_i | \Psi \rangle \quad \ldots (13)$$

where, $Z$ denotes the Pauli-$Z$ matrix.

$Z$=[100 − 1]

Compared to traditional linear gates, this quantum-enhanced gating mechanism allows for richer feature interactions and better representational capacity, which makes the hybrid QLSTM especially well-suited for challenging time-series prediction applications.

The measured outputs from each quantum circuit are then passed through classical layers $clayer_{out}$ with either sigmoid ($\sigma$) or *tanh* activation functions. Each

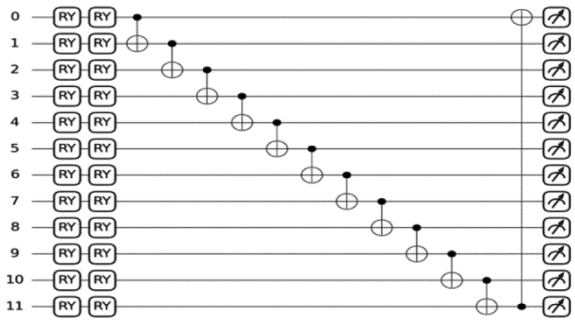


Fig. 4 — 12-Qubit Parameterized quantum circuit used in the hybrid QLSTM architecture

LSTM gate has one of these four layers, enabling independent parameter adjustment.

The cell state and hidden state are updated as:

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \quad \ldots (14)$$

$$h_t = o_t \odot tanh\ tanh(c_t) \quad \ldots (15)$$

where, $\odot$ denotes element-wise multiplication.

Once passing through a linear layer, the final hidden state $h_t$ yields the scalar prediction $v_t$, which in this instance represents the anticipated daily temperature. Depending on the application, this framework can handle both regression and classification tasks.

**Training and Hyperparameters**

To perform a balanced comparison between the conventional LSTM model and the hybrid QLSTM architecture, the same training setups were implemented. To train the models, a time- series dataset was divided using a sliding-window approach with a sequence length of 10-days, where each sequence was treated as a single training sample. At each time step, the input vector consists of four meteorological variables, including mean temperature, humidity, wind speed, and mean atmospheric pressure.

For both models, the dimension of the hidden layer was set to 8 to ensure comparable model capacity. To ensure convergence during training, a learning rate of 0.005, a batch size of 64, and 100 training epochs were used. The hybrid QLSTM architecture employed the same optimization parameters to ensure a fair comparison with the classical baseline. Furthermore, the hybrid model incorporates a variational quantum layer consisting of 12 qubits within the recurrent gating mechanism. Because the concatenated input vector $v_t = [x_t, h_{t-1}]$ has a dimension of exactly 12 (4 input features + 8 hidden dimensions), it maps perfectly to the 12-qubit system. This intentional mathematical alignment ensures a 1-to-1 feature-to-qubit encoding, thereby eliminating the need for additional dimensional compression or expansion in the classical pre-processing layers.

**Evaluation Metrics**

To analyse a model's efficacy, conventional metrics like MSE, RMSE, MAE, and R2 are used to evaluate predictive performance and overall model effectiveness. The subsequent mathematical formulations delineate these metrics:

***Mean Absolute Error (MAE)***

Mean Absolute Error (MAE) is the mean of the absolute differences between the expected and actual values.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|e_i| \quad \ldots (16)$$

where, $e_i = y_i - \hat{y}_i$

***Mean Squared Error (MSE):***

The Mean Square Error (MSE) is the average of the squared errors between the predicted value and the actual value. It is a way of measuring how well the regression model is performing.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n} e_i^2 \quad \ldots (17)$$

***Root Mean Squared Error (RMSE)***

The Root Mean Squared Error (RMSE) is the square root of the MSE and measures the error of prediction. The smaller the RMSE, the better the model is at predicting.

$$\text{RMSE} = \sqrt{\frac{1}{\Pi}\sum_{i=1}^{n} e_i^2} \quad \ldots (18)$$

***$R^2$ (R squared)***

$R^2$, also called the coefficient of determination, quantifies the proportion of the variation in the dependent variable that is explained by the regression model.

$$R^2 = 1 - \frac{SSR}{SST} \quad \ldots (19)$$

where,
n = number of data points,
$e_k$ = (true value) $_k$ - (estimated value) $_k$,
SSR = Residual Sum of Squares,
SST = Total Sum of Squares.

## Experimental Results and Discussions

The experiments were conducted on a system running Windows 10, equipped with an Intel® Core™ i5-10210U processor operating at 2.11 GHz and 8 GB of RAM. The proposed hybrid quantum-classical model was implemented using PennyLane version 0.39.0, which enables the development and training of parameterized quantum circuits integrated with classical deep learning frameworks. Four meteorological factors mean temperature, humidity, wind speed, and mean pressure were used in 10-day windows to predict the mean temperature of the following day. To assess the predictive performance of the models, several statistical evaluation metrics were used. These metrics included the coefficient of determination ($R^2$), Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). Together these metrics provide complementary insights into model performance by

evaluating both the goodness-of-fit and the magnitude of prediction errors.

The predictive performance of the four evaluated models, namely RNN, GRU, LSTM, and the proposed hybrid QLSTM, is summarized in Table 2. Among the classical baseline models, the LSTM achieved the best overall performance, with a MAE of 1.522, an MSE of 3.595, an RMSE of 1.896, and an $R^2$ value of 0.910. The proposed hybrid QLSTM further improved these results, achieving a MAE of 1.466, an MSE of 3.266, an RMSE of 1.807, and an $R^2$ value of 0.919. In contrast, the RNN and GRU models produced lower $R^2$ values and higher prediction errors. These results indicate that, compared with the best performing classical baseline, the proposed hybrid architecture provides a small but consistent improvement in predictive performance. The improvement is shown across all reported evaluation criteria, despite the small numerical increases, indicating that the incorporation of variational quantum circuits does not reduce predictive capability and may improve the model's representation of nonlinear temporal interactions.

The training-loss curves shown in Fig. 5 illustrate the optimization behaviour of the two best-performing models. Both the hybrid QLSTM and the classical LSTM exhibit a rapid decrease in training loss during the initial training epochs, indicating that each model effectively learns the dominant temporal patterns in the dataset. The hybrid QLSTM initially exhibits a little higher training loss, which makes sense given the extra parameter space that the quantum layer adds. However, by approximately the fifth epoch, its loss trajectory closely aligns with that of the classical LSTM. Both models approach near-zero training loss after approximately ten epochs and remain stable throughout the remainder of the training process. This behaviour suggests that the hybrid model maintains the optimization stability of the underlying LSTM framework and that the inclusion of the quantum layer does not hinder convergence.

Validation-loss curves in Fig. 6 follow a similar pattern. Over the training period, the hybrid QLSTM has slightly lower and more steady validation-loss values than the conventional LSTM with less evident swings. Furthermore, it appears that the hybrid model appears to stabilize sooner in the first training epochs. This pattern indicates a minor improvement of generalization performance and improved model stability. However, there is still a slight difference between the two models and so it is essential to interpret the results with care. The present results might be viewed as evidence of competitive performance in a case-study situation rather than clear proof of general superiority since the study is based on a single Delhi climatic dataset with 1461 training samples and 114 test samples. In this context, however, the hybrid model's lower validation loss suggests that the quantum-enhanced gating mechanism may help capture interactions among the meteorological variables more effectively than the purely classical formulation.

Moreover, the small testing subset of 114 samples limits the statistical power to detect subtle overfitting in the quantum layers. Since the variational quantum circuits introduce additional trainable parameters and a highly expressive hypothesis space, they carry an elevated risk of fitting noise in the training data, and a

Table 2 — Predictive performance of the evaluated models

| Metric | RNN | GRU | LSTM | Hybrid QLSTM |
|---|---|---|---|---|
| MAE | 1.687 | 1.575 | 1.522 | 1.466 |
| MSE | 4.216 | 3.888 | 3.595 | 3.266 |
| RMSE | 2.053 | 1.972 | 1.896 | 1.807 |
| R2 | 0.894 | 0.903 | 0.910 | 0.919 |

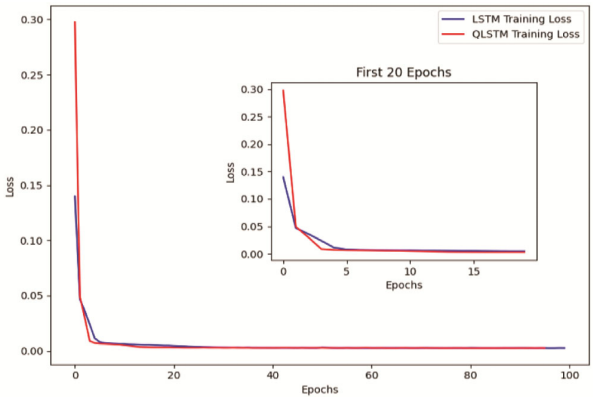


Fig. 5 — Comparison of training loss for the LSTM and hybrid QLSTM models

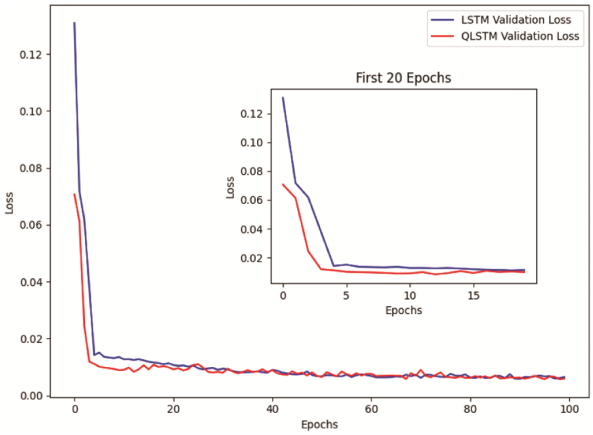


Fig. 6 — Comparison of validation loss for the LSTM and hybrid QLSTM models

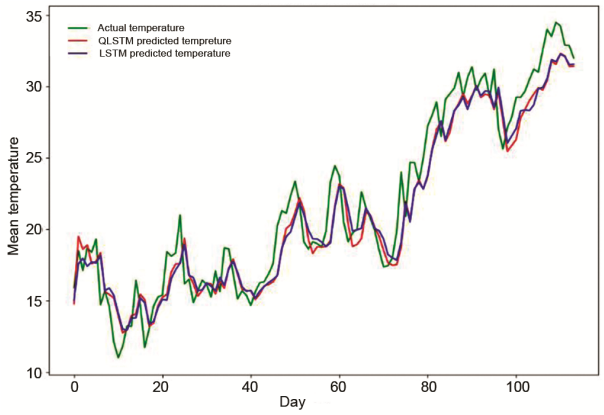


Fig. 7 — Comparison of predicted and actual values for the LSTM and hybrid QLSTM models

test set of this size may not reliably distinguish genuine generalization gains from favourable sampling variance. The reported improvements should therefore be interpreted with this limitation in mind.

The quantitative assessment and the qualitative comparison presented in Fig. 7 are consistent with these findings. Both the LSTM and hybrid QLSTM models closely follow the overall variation in the actual temperature series over the subsequent 100 days, indicating that both approaches successfully capture the dominant forecasting trend. Compared to the standard LSTM, the hybrid QLSTM frequently remains closer to the ground-truth values and appears to follow the observed series more smoothly at several intervals. Nevertheless, both models show local deviations from the actual series, indicating that short-term fluctuations are not fully captured. Therefore, the visual results confirm the numerical findings by indicating that the suggested hybrid model gives a minor but considerable boost in prediction fidelity without generating instability.

In terms of computational cost, the hybrid QLSTM required approximately 360 minutes for 100 training epochs, whereas the classical models required less than one minute under the same experimental setting, with the higher overhead of the QLSTM primarily associated with the simulation and optimization of the parameterized quantum circuit. Although the QLSTM introduces additional computational cost, the primary objective of this study is to investigate its capability to model complex and nonlinear patterns in weather data rather than to minimize training time alone; hence, this overhead is considered in the context of the performance benefits demonstrated by the hybrid model.

## Conclusions

This study used the Delhi climate dataset to present a hybrid quantum-classical LSTM framework for daily mean temperature forecasting. LSTM was chosen as the benchmark for comparison with the suggested hybrid QLSTM because it performed the best among the traditional baselines taken into consideration. In comparison to the classical LSTM, the hybrid model obtained somewhat lower MAE, MSE, and RMSE values as well as a slightly higher $R^2$ score, indicating competitive and enhanced predictive accuracy. These findings imply that variational quantum circuits may be useful for modelling nonlinear temperature dynamics and can be incorporated into recurrent forecasting systems without compromising performance. The improvement on this dataset is minimal, and the present work is viewed as a first empirical case study, rather than definite evidence for quantum advantage. Future works should examine the approach on larger and more diversified meteorology datasets, compare it with more reliable modern baselines, and consider statistical significance, processing cost, and repeat trials.

## References

1 Sevgin F, Machine learning-based temperature forecasting for sustainable climate change adaptation and mitigation, *Sustainability*, **17(5)** (2025) 1812, doi: 10.3390/su17051812.

2 Gong B, Langguth M, Ji Y, Mozaffari A, Stadtler S, Mache K & Schultz M G, Temperature forecasting by deep learning methods," *Geoscientific Model Development,* **15(23)** (2022) 8931–8956, doi: 10.5194/gmd-15-8931-2022

3 Pandey T N, Ravalekar V, Nair S D & Pradhan Sunil K, A Comparative Analysis of Classical Machine Learning Models with Quantum-inspired Models for Predicting World Surface Temperature, *Sci Rep*, **15(1)** (2025) 28443, doi: 10.1038/s41598-025-12515-4.

4 Thi Kieu Tran T, Lee T, Shin J-Y, Kim J-S & Kamruzzaman M, Deep learning-based maximum temperature forecasting assisted with meta-learning for hyperparameter optimization, *Atmosphere*, 11(5) (2020) 487, doi: 10.3390/atmos11050487

5 Mutinda J K, Langat A K & Mwalili S M, Forecasting temperature time series data using combined statistical and deep learning methods: A case study of nairobi county daily temperature, *Int J Math Math Sci*, (2025), doi: 10.1155/ijmm/4795841.

6 Li B & Qian Y**,** Weather prediction using CNN-LSTM for time series analysis: A case study on Delhi temperature data, 2024, doi: 10.48550/arXiv.2409.09414.

7 Nammouchi A, Kassler A & Theocharis A, Quantum machine learning in climate change and sustainability: A short review, in *Association for the Advancement of Artificial Intelligence*, (2023), doi: 10.48550/arXiv.2310.09162.

8 Rao S V, Daily climate time series data, *Kaggle*, 2000, https://www.kaggle.com/datasets/sumanthvrao/daily-climate-time-series-data?select=DailyDelhiClimateTrain.csv.

9 Hochreiter S & Schmidhuber J, Long short-term memory, *Neural Computation*, **9(8)** (1997) 1735–1780.