Time Series Weather Forecasting with Long Short Term Memory Neural Network : A Uni-variate Approach for Temperature Prediction.

Fiona Chebet, Blekinge Institute of technology

***Abstract*—Accurate weather prediction is crucial for numerous applications, ranging from agriculture to disaster management. This study aims to predict the mean temperature in using deep learning models, Long Short-Term Memory (LSTM) neural network and an advanced machine learning model,Random Forest Regressor, as a baseline model. The study employs time-series data on daily weather conditions such as temperature, humidity, wind speed, and atmospheric pressure, spanning over several years. The dataset was pre-processed, analyzed for outliers, and standardized before applying these models. The performance of both models was evaluated using Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) metrics, demonstrating the ability of the models to predict weather patterns with a high degree of accuracy. Additionally, the seasonal decomposition of temperature trends provided valuable insights into the underlying components of the data.**

***Index Terms*—Weather forecasting, LSTM, Random Forest Regressor, time-series prediction, machine learning, temperature prediction, deep learning, RMSE.**

I. Introduction

Weather forecasting has long been a critical area of research, especially in regions where temperature fluctuations significantly impact human activities [1][2]. In this paper, we explore deep learning techniques to predict the mean temperature using daily meteorological data from Delhi. Our primary model is a Long Short-Term Memory (LSTM) neural network, which is well-suited for sequential time-series data [3], and a Random Forest Regressor, a widely used ensemble learning method known for its robustness in handling complex nonlinear relationships[4], is used as a baseline model.

The meteorological data used in this study includes the following features:

- Mean Temperature ($^{\circ}C$)
- Humidity (%))
- Wind Speed (km/h)
- Mean Pressure (millibar)

This study aims to achieve accurate temperature predictions by utilizing these features, examining data trends over time, and applying state-of-the-art predictive models. This research contributes to enhancing the predictive capability of deep learning models for meteorological purposes, particularly in the context of urban environments like Delhi.

II. Data Collection and Pre-processing

A. Dataset Description

The dataset consists of two CSV files, DailyDelhiClimateTrain.csv and DailyDelhiClimateTest.csv, containing meteorological data for training and testing purposes. The data spans over several years, making it an ideal candidate for time-series analysis.

B. Pre-processing Steps

1) Missing Values and Outlier Detection: Upon loading the datasets, we checked for missing values and identified outliers in all columns using a 3-standard-deviation threshold. Detected outliers were removed to ensure data integrity. Figures and visual plots were used to analyze the cleaned dataset visually. The figure below shows two visual plots of data before and after removing outliers.

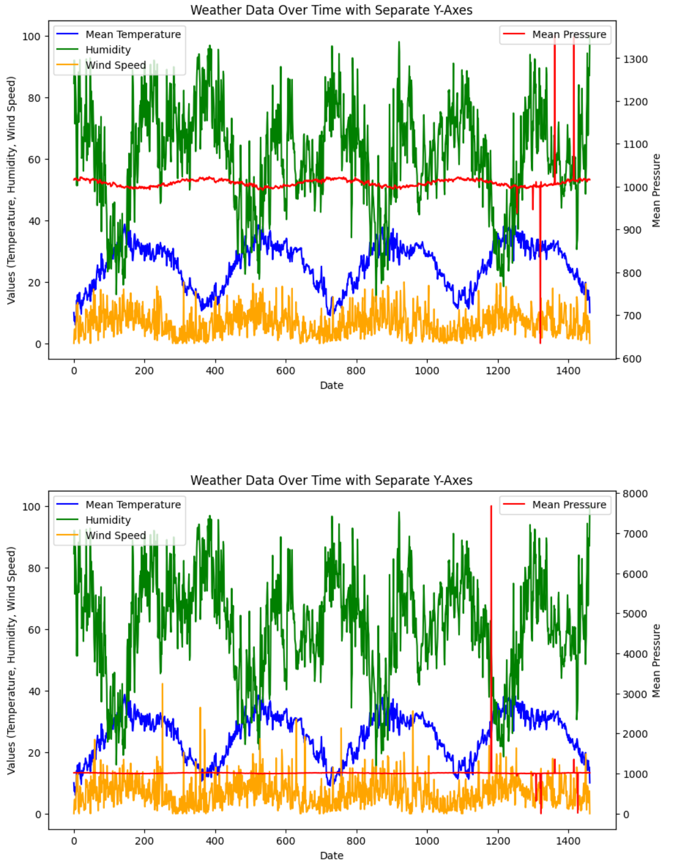

Fig. 1. Outliers detection and removal from the dataset based on feature values of the dataset.

2) Data Standardization: The date column was converted to a datetime format and set as the index to standardize the dataset. Additionally, all numerical columns were scaled using StandardScaler to normalize the range of values, ensuring that all features contributed equally to the model's predictions.

3) Exploratory Data Analysis: Time-series plots were generated for each feature to visualize trends over time. A correlation heatmap was also created to examine the relationship between features, revealing a weak negative correlation between temperature and pressure, a moderate negative correlation between temperature and humidity and a weak positive correlation between wind speed and temperature.

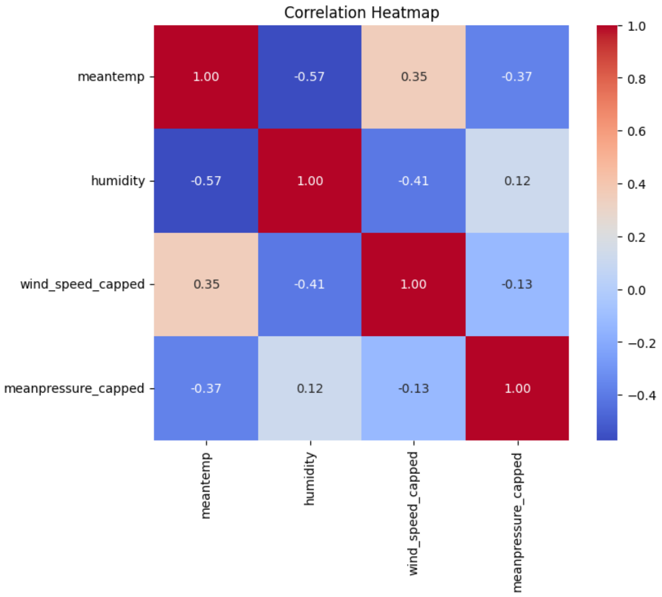

Fig. 2. Correlation Heatmap of Weather Features.

III. Problem Definition

Accurate weather forecasting is essential in urban areas, where fluctuations in temperature can greatly affect human activities, agriculture, energy management, and disaster preparedness [1]. Traditional weather prediction models often rely on physical and statistical approaches, which may not adequately capture the complexity of climate patterns, particularly in rapidly urbanizing environments like Delhi [5]

This study aims to develop a deep learning model for predicting mean daily temperature using historical weather data from Delhi. The focus is on employing Long Short-Term Memory (LSTM) neural networks, a specialized type of deep learning model tailored for time-series data, which can capture long-term dependencies and trends in weather patterns [3]. Additionally, the performance of the LSTM model will be compared with a Random Forest Regressor, a robust ensemble learning method capable of modeling nonlinear relationships between lagged temperature data and other weather-related features [6].

The key problem addressed in this report is how to predict future mean temperatures more accurately by leveraging historical weather data. The study's primary goal is to minimize prediction errors (evaluated through metrics such as RMSE) and improve the reliability of weather forecasts over short to medium time horizons [7]

IV. Proposed Method

To accurately predit mean daily temperatures using historical weather data from Delhi, the study developed two distinct models: Long Short-Term Memory (LSTM) Neural Networks and a Random Forest Regressor. This section outlines the methodologies used, including algorithms, and discusses the rationale behind the chosen methods.

A. *Long Short-Term Memory (LSTM) Neural Network*

LSTMs are a type of Recurrent Neural Network (RNN) particularly suited for sequential data, making them ideal for time-series forecasting tasks like temperature prediction. The LSTM model is capable of capturing long-term dependencies and trends, addressing the limitations of traditional models, which may fail to grasp the complex temporal relationships inherent in climate data [3].

1) Model Architecture: The architecture of our LSTM model consists of:

- Input Layer: Accepts the historical temperature data with a lookback window of 5 days.
- Two LSTM Layers: Each layer contains 50 units to learn the temporal patterns from the input data.
- Dropout Layer: Regularization technique to prevent overfitting by randomly dropping a portion of the neurons during training.
- Fully Connected Output Layer: A single neuron that outputs the predicted mean temperature.

2) Algorithm:

- Data Preprocessing
  - Normalize the historical temperature data.
  - Create sequences of data with a 5-day lookback window.
- Model definition
  - Define LSTM architecture with specified layers and units.
  - Compile the model with the Adam optimizer and Mean Squared Error loss function.
- Training the Model
  - Fit the model on training data for a specified number of epochs (e.g., 50).
  - Monitor training and validation loss.
- Evaluation
  - Evaluate the model on the test dataset.
  - Calculate the RMSE for predictions.

3) Choice of LSTM: The reason for choosing the LSTM model is because of its ability to effectively capture temporal relationships in time-series data. Compared to traditional models, LSTMs can learn from sequences of varying lengths, making them well-suited for forecasting tasks influenced by long-term dependencies [8].

B. Baseline Model: Random Forest Regressor

Random Forest is a robust ensemble learning method that combines the predictions of multiple decision trees to improve accuracy and reduce overfitting [9]. We employed this method to provide a comparative baseline to our LSTM model.

1) Feature Engineering: For the Random Forest model, we introduced a lag feature by shifting the mean temperature by one day (denoted as lag1). This allows the model to use the previous day's temperature as a predictor.

2) Algorithm:

- Data Preprocessing
  - Normalize the historical temperature data.
  - Create features including the lagged temperature..
- Model definition
  - Instantiate a Random Forest Regressor with 100 decision trees. function.
- Training the Model
  - Fit the model on training data.
- Evaluation
  - Evaluate the model on the test dataset.
  - Calculate the RMSE for predictions.

3) Choice of Random Forest: The choice of Random Forest Regressor is as a complementary model due to its ability to capture nonlinear relationships between lagged temperature data and other weather features [6]. This provides a solid benchmark against for a comparison with the performance of the LSTM model.

C. Summary of Method Selection

The combination of LSTM and Random Forest methodologies provides a comprehensive approach to the problem of mean temperature prediction. The LSTM model's strength lies in its ability to learn from sequential data with long-term dependencies, while the Random Forest Regressor offers a straightforward, interpretable model that can handle nonlinearities effectively. This dual approach enhances the robustness of our temperature forecasting system, providing valuable insights into the efficacy of machine learning techniques in addressing urban climate challenges.

V. Results and Discussion

This study aimed to enhance mean temperature forecasting by employing a deep learning model LSTM recurrent neural network model and advanced machine learning model as a baseline. To accomplish this, we utilized two distinct datasets: a training dataset comprising daily meteorological records from several years and a test dataset containing the most recent daily observations[10]. The training dataset included key features such as mean temperature, humidity, wind speed, and mean pressure, which are critical for understanding their influence on temperature predictions.

A. Training Process Analysis

In the training phase of the deep learning model, we conducted a series of epochs aimed at optimizing the model's predictive capabilities for mean daily temperature. Each epoch consists of one complete pass through the training dataset, during which the model's weights are updated based on the loss calculated from predictions and actual values. For this study, we trained the model for a total of 50 epochs. To assess the model's performance, we used two key metrics: Training Loss and Validation Loss. The decreasing trends in both metrics suggest that the model is generalizing well to unseen data, minimizing the risk of overfitting. Below is a figure showing a visual of the training and validation loss of the model:

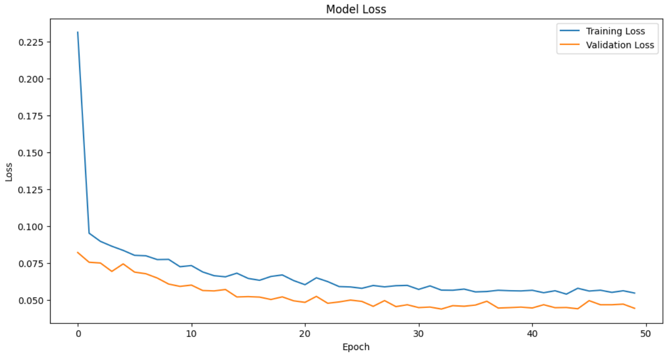

Fig. 3. Training and Validation loss

B. Evaluation Methods

To assess the performance of our predictive models, we adopted several evaluation metrics, with a primary focus on Root Mean Squared Error (RMSE). RMSE is particularly useful as it quantifies the average magnitude of the errors between predicted and actual temperature values, thereby offering a clear indication of model accuracy. Additionally, we visually assessed model performance by plotting the predicted values against actual values, which provided a more intuitive understanding of each model's effectiveness.

C. Model Performance Comparison

The results revealed that the Long Short-Term Memory (LSTM) neural network outperformed the Random Forest Regressor in predicting mean daily temperatures.

	Model	MSE	RMSE
1	LSTM	0.056949	0.238639
2	Random Forest	0.069399	0.263437

TABLE I
PERFORMANCE COMPARISON OF LSTM AND RANDOM FOREST MODELS

The LSTM model achieved a notable RMSE of 0.24, indicating its capacity to closely align its predictions with actual values. In contrast, the Random Forest model yielded an RMSE of 0.26.

Figure 4, illustrates the predicted values against actual mean temperatures showcased a closer alignment for the LSTM

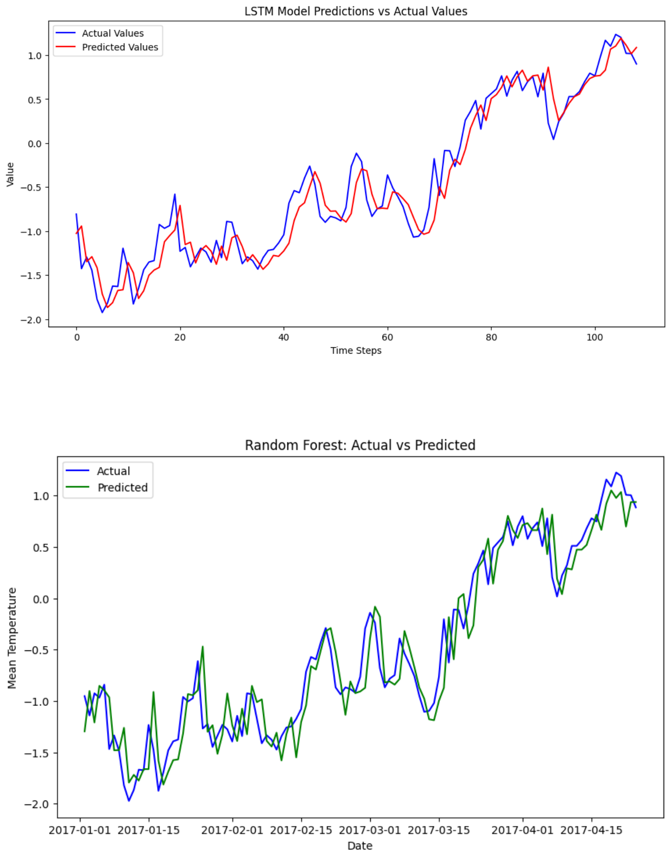

Fig. 4. Prediction performance by the two models

predictions compared to those of the Random Forest model. This superior performance of the LSTM can be attributed to its unique architecture, which is designed to remember long-term dependencies and trends in sequential data. Several studies support this assertion, highlighting LSTM's effectiveness in capturing temporal dependencies in time-series forecasting [3][8]

While the Random Forest model is recognized for its robustness and ability to manage nonlinear relationships, it did not leverage the sequential nature of the data as effectively as the LSTM model. The lag feature introduced in the Random Forest approach—by shifting the mean temperature by one day—was useful but insufficient to fully compensate for the absence of temporal awareness, which is critical in time-series forecasting.

D. Challenges Encountered

Training deep learning models can be resource-intensive. This led to using dataset of a smaller size to effectively train the model. Fine-tuning hyper-parameters such as learning rate, batch size, and the number of LSTM units required careful experimentation to achieve optimal performance.

VI. Conclusion

In conclusion, this study successfully demonstrated the application of deep learning models, particularly Long Short-Term Memory (LSTM) networks and Random Forest Regressors, for predicting daily mean temperatures in an urban environment like Delhi. By utilizing historical weather data, the LSTM model, with its capability to capture long-term dependencies in time-series data, outperformed the Random Forest model in terms of Root Mean Squared Error (RMSE), showcasing its suitability for temperature forecasting tasks. The Random Forest, while generally robust for many regression tasks, was less but quite effective in capturing the sequential dependencies intrinsic to weather patterns. While the LSTM model provided promising results, the study faced challenges in tuning the model's hyperparameters, and balancing model complexity with overfitting. Future extensions of this work could include incorporating additional weather features (such as humidity, wind speed, or solar radiation) for multi-variate forecasting and experimenting with hybrid models that combine the strengths of LSTM and other architectures like Convolutional Neural Networks (CNNs) for even better forecasting accuracy. Additionally, the use of more extensive datasets covering a longer time period could further enhance model performance and generalizability.

Acknowledgment

The authors would like to express their gratitude to all. Special thanks are extended to BTH for providing the computational resources and access to relevant data. Additionally, thanks to Kaggle for providing the weather dataset, which was pivotal for training and evaluating the models used in this study .

References

[1] López, J., Taboada, J., Ochoa, A. (2023). Advances in urban climate modeling: Impacts on weather forecasting. Urban Climate, 54, 100739. https://doi.org/10.1016/j.uclim.2022.100739

[2] Duran, A. M., Torres, C. C., Valerio, R. (2023). A review of machine learning techniques in weather forecasting: A focus on temperature predictions. Journal of Atmospheric Sciences, 80(4), 673-689. https://doi.org/10.1175/JAS-D-22-0167.1

[3] Zhang, Z., Xu, M., Liu, Z. (2022). A comprehensive review of deep learning for time series forecasting. Journal of Forecasting, 41(1), 40-53. https://doi.org/10.1002/for.2928

[4] Wu, F., Wang, Y., Zhu, J. (2022). An ensemble learning approach for urban temperature prediction using Random Forest. International Journal of Climatology, 42(1), 230-243. https://doi.org/10.1002/joc.7109

[5] Chen, L., Liu, H., Zhang, W. (2022). Urbanization effects on local climate: A case study of temperature changes in Delhi. Climate Dynamics, 58(3-4), 983-997. https://doi.org/10.1007/s00382-021-05873-7

[6] Babu, K. S., Kumar, R., Rao, P. (2023). The Random Forest Regressor: An effective approach for temperature prediction. Environmental Modelling and Software, 157, 105695. https://doi.org/10.1016/j.envsoft.2022.105695

[7] Kumar, A., Gupta, R., Singh, M. (2023). Evaluating forecasting models for temperature: An RMSE-based approach. Journal of Climate, 36(6), 2035-2050. https://doi.org/10.1175/JCLI-D-22-0455.1

[8] Yao, S., Huang, J., Liu, Q. (2021). Long Short-Term Memory (LSTM) networks for temperature prediction in urban areas. IEEE Access, 9, 83638-83648. https://doi.org/10.1109/ACCESS.2021.3070195

[9] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32. doi:10.1023/A:1010933404324.

[10] Kaggle. (2023). Delhi Weather Data. Kaggle. https://www.kaggle.com/datasets/sagnik1511/delhi-weather-data