A Comparative Analysis of Univariate Deep Learning-based Time-series Models for Temperature Forecasting of the Bhubaneshwar

1st Naba Krushna Sabat
Dept. of Electronics & Communication Engineering
National Institue of Technology, Rourkela
Rourkela, India
nabakrushna4u@gmail.com

2nd Rashmiranjan Nayak
Dept. of Electronics & Communication Engineering
National Institue of Technology, Rourkela
Rourkela, India
rashmiranjan.et@gmail.com

3rd Umesh Chandra Pati
Dept. of Electronics & Communication Engineering
National Institute of Technology, Rourkela
Rourkela, India
ucpati@nitrkl.ac.in

4th Santos Kumar Das
Dept. of Electronics & Communication Engineering
National Institute of Technology, Rourkela
Rourkela, India
dassk@nitrkl.ac.in

***Abstract*—Meteorological variables such as temperature, humidity, and pressure significantly impact living things. Because of the ambiguity and rapid climatic change in the environment, weather prediction with higher accuracy is essential. With the help of deep learning models, the prediction of weather parameters becomes easier and more accurate as compared to traditional methods. This paper investigates various deep learning models such as Long Short Term Memory (LSTM), Gated Recurrent Unit (GRU), Bidirectional Long Short Term Memory (BiLSTM), Bidirectional Gated Recurrent Unit (BiGRU), and Neural Basis Expansion Analysis for Time Series (NBEATS) for the prediction of the temperature of the city of Bhubaneswar. The comparative analysis of these developed models in terms of various performance metrics, such as MAE, MSE, RMSE, and R^2 score, concludes that the prediction of the BiGRU model is more accurate as compared to the other implemented models.**

***Index Terms*—Weather Prediction, LSTM and BiLSTM Models, N-Beats Model, GRU and BiGRU Models, Deep Learning Method.**

I. Introduction

Weather parameters, such as temperature, pressure, rainfall, and humidity, are changing abruptly because of tree cutting, rapid construction of buildings and enterprises such as chemical factories, etc. changes in weather parameters have an immediate impact on humans and the environment, from daily life, both directly and indirectly. For example, people often consult weather forecasts to decide how to dress daily. In the agriculture field, a few districts are highly affected in the state of Odisha in the summer and rainy seasons. Since the weather and its seasonal impact are unpredictable, it is highly essential to forecast weather parameters to protect and save human lives. Several methodologies are used for weather prediction, such as numerical methods [1], statistical methods, etc. Traditional methods are complex and time-consuming for prediction. The statistical methods such as autoregressive integrated moving average (ARIMA) [2], seasonal autoregressive integrated moving average (SARIMA) [3], Fb-prophet [4], etc. provide inadequate results in terms of poor prediction if the data has deteriorated or for non-linearity of data.

With the advancement of artificial intelligence technology, machine learning, and deep learning methods, time series data prediction has become easier and also provides adequate results. Recently, several machine learning methods such as linear regression [5], support vector machine (SVM) [6], convolutional neural network (CNN) [7], support vector regression (SVR) [8], and deep learning methods such as long short term memory (LSTM) [9], gated recurrent unit (GRU) [10] are widely used for weather prediction. Deep learning techniques have gained popularity due to their extensive usage in sequential models and their ability to handle multidimensional time-series data. This research uses different deep learning models for temperature prediction to find the best suitable model.

The rest of this paper is structured as follows. Section II describes the problem statement. The methodologies and related experiments with the outcome are elaborated in Sections III and IV, respectively. Finally, the conclusion and future scope are presented in Section V.

II. PROBLEM STATEMENT

The present research aims to predict one of the meteorological parameters, i.e., temperature, by using the historical time-series temperature data of Bhubaneswar. Further, the time series modeling of the temperature data will be carried out using various widely used models such as LSTM, BiLSTM, GRU, BiGRU, and NBEATS. Finally, the research is aimed at selecting the best-performing time-series models for the prediction of the temperature of Bhubaneswar.

III. METHODOLOGIES

Following steps are used in the methodologies to perform this research.

A. Data Collection and Preprocessing

Initially, the scope of the research is determined, and the corresponding data source is identified for collecting the weather data. Thereafter, the raw meteorological data is processed to convert the data into a suitable format for the deep learning model.

1) Study Area: Bhubaneswar is the capital of Odisha state which is situated in the eastern part of India. This city is located at 20.296059 latitudes and 85.824539 longitudes with an elevation of 58m and is found at the GPS coordinates 20°17' 45.8124" N and 85°49' 28.3404" E. Fig. 1 depicts the study area with a broad borderline and a smaller green circle. This state has nineteen weather stations, one of which is situated in the district of Khordha. Subsequently, the temperature parameter of the weather data of Bhubaneswar city is considered for this research work.

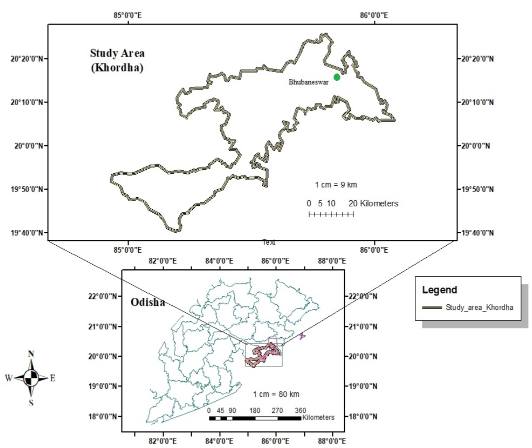

Fig. 1. Research area.

2) Data Preprocessing: The dataset has 2808 samples from the years 2015 to 2022, and the data is available on a daily basis. This data is preprocessed in different stages, as described in this section. The daily average temperature replaces these daily temperature samples to reduce the dimensionality. Subsequently, the data outlier is detected and removed using the quantile regression method to prevent the biasing of the data during modeling. Then, the empty locations of the time series temperature data are again filled with the median imputation method. Data normalization is applied to transform the original data in the range of 0 to 1 from their large-scale variation range for the smooth training of the deep learning-based model. Hence, the Min-Max normalization approach is used to scale up temperature data from 0 to 1. Subsequently, the whole data is segregated into three parts such as training data (80% samples of the complete data), 10% validation data(10% samples of the complete data), and 10% testing data (10% samples of the complete data).

B. Developed Deep Learning Model

In this stage, different widely used deep learning models such as LSTM, BiLSTM, GRU, BiGRU, and NBEATS are used to model the time-series temperature data of the Bhubaneswar as depicted in Fig. 2.

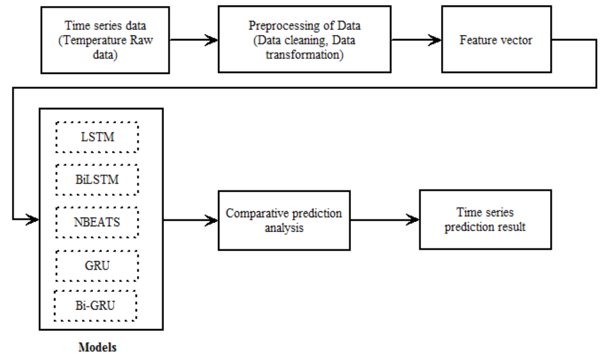

Fig. 2. Block diagram of deep learning model for temperature prediction.

1) LSTM and BiLSTM model: Recurrent Neural Networks (RNNs) can learn long-term dependencies of the sequential data. However, RNNs suffer from gradient explosion problems when the time series data are excessively lengthy. Subsequently, it is infeasible to train the RNNs. Hence, a special type of RNN, i.e., LSTM, is used to model the time series data effectively. LSTM is capable of processing and analyzing the complete data sequence with the help of its feedback connection. The information is stored in a memory cell called a "cell state," which keeps its state across time. The memory cell consists of three gates, such as input, output, and forget gates. Information flows in, out, or is memorized in the cell and is regulated by the gates [11], [12]. The typical architecture of the LSTM cell used to build the LSTM network is shown in Fig. 3.

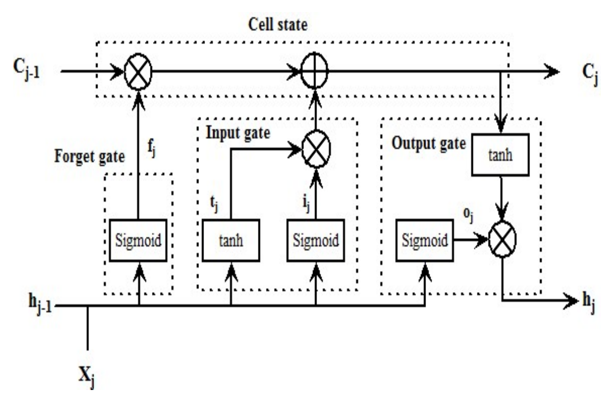

Fig. 3. Architecture of the LSTM cell.

Here, X_j is the input, h_{j-1} is the previous hidden state output given to the LSTM model. The sigmaod and tanh activation functions are used to regulates the value flowing throw the network.The mathematical formulas guiding the LSTM cell's performance in accordance with the input's (i_j), output's (o_j), forget (f_j) and cell (c_j) for j^{th} state is expressed in Eq. 1.

$$\begin{aligned} f_j &= \sigma(W_f.[h_{j-1}, X_j]) + b_f \\ i_j &= \sigma(W_i.[h_{j-1}, X_j]) + b_i \\ t_j &= tanh(W_t.[h_{j-1}, X_j]) + b_t \\ o_j &= \sigma(W_o.[h_{j-1}, X_j]) + b_o \\ c_j &= f_j * c_{j-1} + t_j * i_j \end{aligned} \tag{1}$$

Here, the parameters W_f, W_i, W_t , W_o and b_f, b_i, b_t, b_o are the weights of bias of forget, input and output states. The output of LSTM model is represented as h$_j$ = o$_j$ $*$ tanh c_j. Similarly, the Bi-directional LSTM (BiLSTM) is made up of two LSTM units: one that processes input going ahead (forward propagation) and the other going backward (backward propagation) [13]. Hence, the BiLSTM model can fit the input data more accurately as compared to the LSTM model.

2) NBEATS model: Neural Basis Expansion Analysis for Time Series (NBEATS) model consists of a number of stack levels, each of which is made up of a number of blocks. The blocks are connected to the feed-forward network through the forecast and backcast links. The block focuses on the residual error to produce only a partial forecast. The stack transfers the accumulated forecast value of each block to the subsequent stack in the stack layer. Finally, partial forecast results are combined to get the unified forecast value of the temperature [14]. A typical block-level architecture of the NBEATS model is illustrated in Fig. 4.

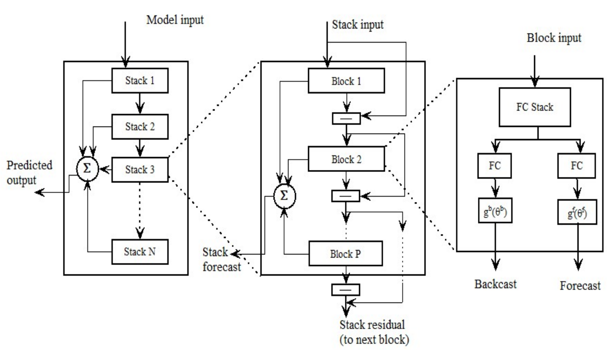

Fig. 4. Block-level architecture of NBEATS model [14].

3) GRU and BiGRU model: The Gated Recurrent Unit (GRU) is another type of RNN model. Like LSTM, GRU has two gates, such as reset and update gates, except forget gate. The model uses these gates to improve the output. These two have control of the flow of data through the model. In this approach, the reset gate is applied to reject the data from prior hidden states, whereas the update gate is used to regulate the data from previous hidden states that is taken over into the present state [10]. The architecture of GRU is depicted in Fig. 5. The mathematical formulas guiding the GRU cell's performance in accordance with the reset (r_j), update (z_j), hidden (h_j) and new hidden ($\tilde{h_j}$) for j^{th} state is expressed as

$$\begin{aligned} r_j &= \sigma(W_r.[h_{j-1}, X_j]) + b_r \\ z_j &= \sigma(W_z.[h_{j-1}, X_j]) + b_z \\ \tilde{h_j} &= tanh(W_h.[r_j * h_{j-1}, X_j]) + b_h \\ h_j &= (1 - z_j) * h_{j-1} + z_j * \tilde{h_j} \end{aligned} \tag{2}$$

where, the parameters W_r, W_z, W_h and b_r, b_z, b_h are the weights of bias of reset, update and hidden states. Further, the Bidirectional GRU (BiGRU) provides better performance due to its ability to process the information accumulated from both forward and backward directions. However, the increase in performance of the BiGRU model is achieved at the cost of additional computation [15].

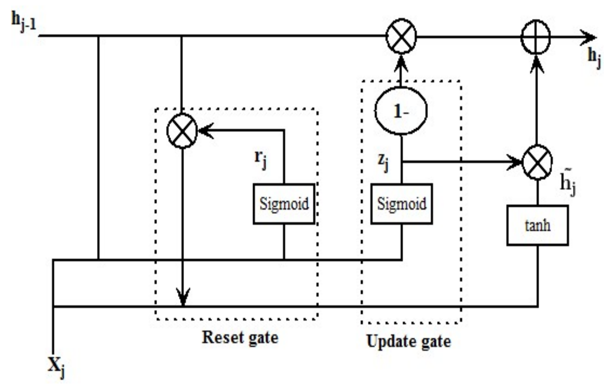

Fig. 5. Architecture of GRU model.

4) Performance Evaluation Criteria: The performance of the models is measured in terms of various performance metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE) [16] and R^2 score [17] . The error performance are calculated using the Eq. (3 - 6). The model with the lowest MAE, MSE, RMSE, and highest R^2 score is considered as the best model for prediction.

$$MAE = \frac{1}{N}\sum_{k=1}^{N}|\Theta_k| \tag{3}$$

$$MSE = \frac{1}{N}\sum_{k=1}^{N}(\Theta_k)^2 \tag{4}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(\Theta_k)^2} \tag{5}$$

$$R^2 score = 1 - \frac{\sum\left(A_k - \widehat{A_k}\right)^2}{\sum\left(A_k - \overline{A_k}\right)^2} \tag{6}$$

Where, Θ_k denotes the difference between Actual value A_k and predicted value $\widehat{A_k}$, i.e. Θ_k = A_k - $\widehat{A_k}$ and k represent the number of data points. The term $\overline{A_k}$, denotes the mean of actual value.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

The models such as LSTM, BiLSTM, NBEATS, GRU, and BiGRU are executed on the Windows 10 operating system, having the configuration of an Intel Core i7 CPU and RAM of 8 GB. The models are developed using Python programming with deep learning frameworks such as Keras, Tensorflow, Matplotlib, and Pandas. The epochs utilized here are set to 100. The optimizer is set to Adam, and the batch size is set at 64. As a loss function, MSE and an activation function, ReLU (Rectified Linear unit), are employed. All deep learning models have error calculation parameters such as MAE, MSE, RMSE, and R^2 score which are described in Table I. One of the important online performance parameters, like execution times of all the developed deep learning models, is compared in Table I. The LSTM model takes minimum execution time with degraded performances. However, the BiGRU model provides significantly better performances at the cost of a minimal increase in execution time. Fig. [6 - 10] represents the deep learning models LSTM, BiLSTM, NBEATS, GRU, and BiGRU, as well as their test predictions. The error performance comparison is shown in the Fig. 11.

TABLE I
ERROR PERFORMANCE ANALYSIS OF THE DEEP LEARNING MODELS

Model	MAE	MSE	RMSE	R^2 Score	Execution Time (in Second)
LSTM	0.9531	1.5278	1.2360	0.90429	**25.777**
BiLSTM	0.8561	1.2812	1.1319	0.9197	32.7664
NBEATS	0.8377	1.2741	1.1287	0.9193	769.48
GRU	0.8602	1.2442	1.1154	0.9209	44.0276
BiGRU	**0.8097**	**1.1095**	**1.1053**	**0.9288**	46.3783

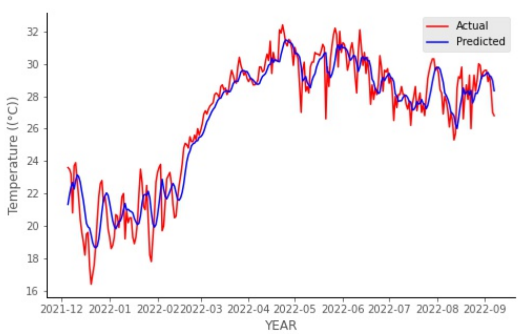

Fig. 6. Prediction analysis of temperature using LSTM model.

V. CONCLUSION

In this study, five deep learning models (LSTM, BiLSTM, NBEATS, GRU, and BiGRU) are used to forecast the temperature and meteorological data for the city of Bhubaneswar. The models are provided with preprocessed historical data from 2015 to 2022. The comparative analysis of the models in terms of the performance metrics indicated that the BiGRU model performs better as compared to the other four models. This

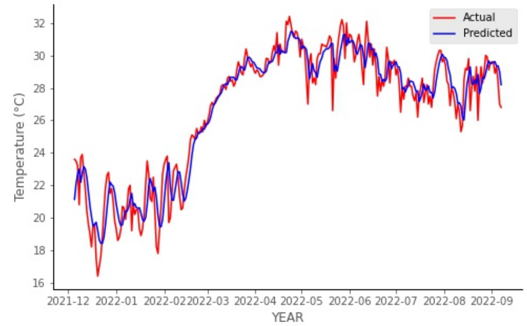

Fig. 7. Prediction analysis of temperature using BiLSTM model.

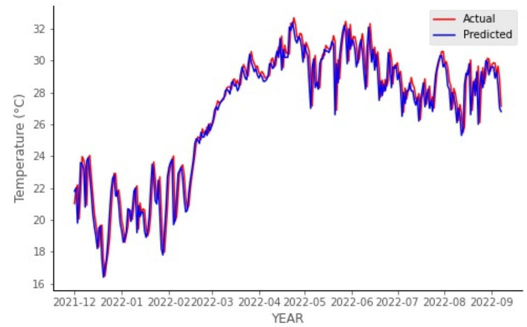

Fig. 8. Prediction analysis of temperature using NBEATS model.

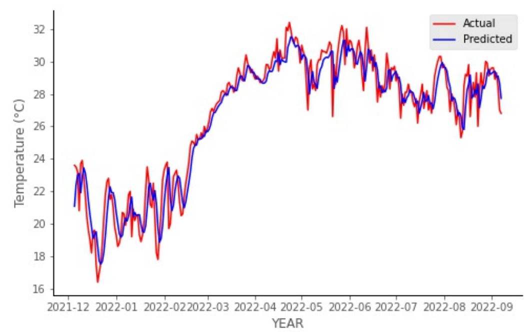

Fig. 9. Prediction analysis of temperature using GRU model.

is because the BiGRU offers the lowest MAE, MSE, RMSE values and highest R^2 score as compared to other models. Further, independent models can be combined to increase the time series forecasting model's accuracy.

.

REFERENCES

[1] P. Chen, A. Niu, D. Liu, W. Jiang, and B. Ma, "Time series forecasting of temperatures using sarima: An example from nanjing," in *IOP Conference Series: Materials Science and Engineering*, vol. 394, no. 5. IOP Publishing, 2018, p. 052024.

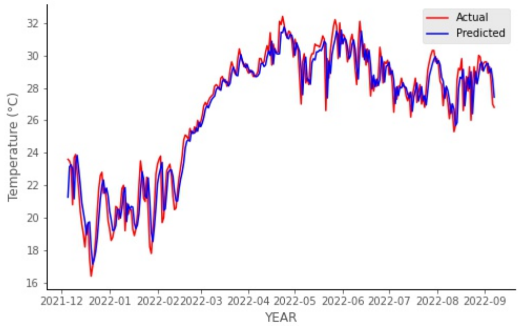

Fig. 10. Prediction analysis of temperature using BiGRU model.

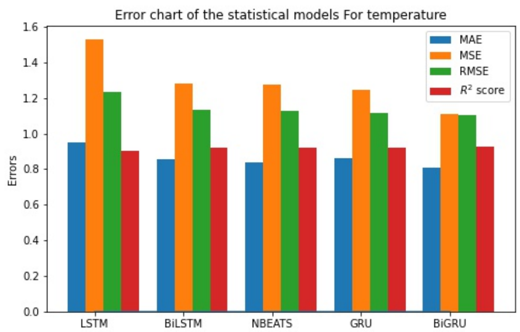

Fig. 11. Comparison of error performance for temperature prediction.

[2] N. Shivhare, A. K. Rahul, S. B. Dwivedi, and P. K. S. Dikshit, "Arima based daily weather forecasting tool: A case study for varanasi," *Mausam*, vol. 70, no. 1, pp. 133–140, 2019.

[3] T. Dimri, S. Ahmad, and M. Sharif, "Time series analysis of climate variables using seasonal arima approach," *Journal of Earth System Science*, vol. 129, no. 1, pp. 1–16, 2020.

[4] Z. Z. Oo and P. Sabai, "Time series prediction based on facebook prophet: A case study, temperature forecasting in myintkyina," *International Journal of Applied Mathematics Electronics and Computers*, vol. 8, no. 4, pp. 263–267, 2020.

[5] S. R. Shams, A. Jahani, S. Kalantary, M. Moeinaddini, and N. Khorasani, "The evaluation on artificial neural networks (ann) and multiple linear regressions (mlr) models for predicting so2 concentration," *Urban Climate*, vol. 37, p. 100837, 2021.

[6] Y. Radhika and M. Shashi, "Atmospheric temperature prediction using support vector machines," *International journal of computer theory and engineering*, vol. 1, no. 1, p. 55, 2009.

[7] M. Qiu, P. Zhao, K. Zhang, J. Huang, X. Shi, X. Wang, and W. Chu, "A short-term rainfall prediction model using multi-task convolutional neural networks," in *2017 IEEE international conference on data mining (ICDM)*. IEEE, 2017, pp. 395–404.

[8] R. I. Rasel, N. Sultana, and P. Meesad, "An application of data mining and machine learning for weather forecasting," in *International conference on computing and information technology*. Springer, 2017, pp. 169–178.

[9] Z. Karevan and J. A. Suykens, "Spatio-temporal stacked lstm for temperature prediction in weather forecasting," *arXiv preprint arXiv:1811.06341*, 2018.

[10] X. Zhou, J. Xu, P. Zeng, and X. Meng, "Air pollutant concentration prediction based on gru method," in *Journal of Physics: Conference Series*, vol. 1168, no. 3. IOP Publishing, 2019, p. 032058.

[11] W. Lu, J. Li, Y. Li, A. Sun, and J. Wang, "A cnn-lstm-based model to forecast stock prices," *Complexity*, vol. 2020, 2020.

[12] A. P. Kogekar, R. Nayak, and U. C. Pati, "A cnn-bilstm-svr based deep hybrid model for water quality forecasting of the river ganga," in *2021 IEEE 18th India Council International Conference (INDICON)*, 2021, pp. 1–6.

[13] F. Shahid, A. Zameer, and M. Muneeb, "Predictions for covid-19 with deep learning models of lstm, gru and bi-lstm," *Chaos, Solitons & Fractals*, vol. 140, p. 110212, 2020.

[14] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-beats: Neural basis expansion analysis for interpretable time series forecasting," *arXiv preprint arXiv:1905.10437*, 2019.

[15] A. P. Kogekar, R. Nayak, and U. C. Pati, "A cnn-gru-svr based deep hybrid model for water quality forecasting of the river ganga," in *2021 International Conference on Artificial Intelligence and Machine Vision (AIMV)*, 2021, pp. 1–6.

[16] K. Wu, J. Wu, L. Feng, B. Yang, R. Liang, S. Yang, and R. Zhao, "An attention-based cnn-lstm-bilstm model for short-term electric load forecasting in integrated energy system," *International Transactions on Electrical Energy Systems*, vol. 31, no. 1, p. e12637, 2021.

[17] A. Ash and M. Shwartz, "R2: a useful measure of model performance when predicting a dichotomous outcome," *Statistics in medicine*, vol. 18, no. 4, pp. 375–384, 1999.