Volume: 16 Issue: 06 Pages: 6677 - 6682 (2025) ISSN: 0975-0290

Predictive Modelling of Meteorological Data Using Time Series Methods

Hymavathi. M
Department of Computer Science and Engineering, N.B.K.R.I.S.T, Vidyanagar, Andhra Pradesh, India.
Sandeepu Kumar. K
Dept of Computer Science, Gayatri Vidya Parishad College for Degree and P.G Courses (A), Rushikonda, Visakhapatnam, Andhra Pradesh, India.
Jahnavi. Y *
Department of Computer Science, Government Degree College - Naidupet, Tirupathi (Dt), Andhra Pradesh, India.

ABSTRACT

Accurate weather forecasting plays a vital role in sectors such as agriculture, transportation, energy, and disaster management. With the increasing availability of historical meteorological data, time series forecasting has emerged as a powerful approach to predicting weather conditions. This study explores the application of various time series models including statistical, machine learning, and deep learning techniques to forecast key weather parameters such as temperature, humidity, and precipitation. The model is trained using Adam optimizer with 0.001 learning rate and calculate the loss using Mean Squared Error. Historical weather data is pre-processed to address issues like missing values, seasonality, and noise, and models are evaluated based on metrics such as Mean Squared Error. Training Loss and Validation Loss is calculated for various epochs. The prediction is done by using validation set. Results demonstrate that while statistical models perform adequately for short-term forecasts, deep learning methods capture complex temporal patterns more effectively.

Keywords - **Time Series Forecasting, Deep Learning, LSTM, Training and Validation Loss, Mean Squared Error (MSE).**

Date of Submission: March 20, 2025 Date of Acceptance: April 28,2025

I. INTRODUCTION

Weather prediction is an essential aspect of modern society, influencing a wide range of sectors including agriculture, aviation, transportation, energy management, and disaster preparedness. With climate variability and extreme weather events becoming more frequent, the demand for accurate and timely weather forecasts has grown significantly. Traditionally, weather forecasting has relied on numerical weather prediction (NWP) models that use complex mathematical equations and physical laws of atmospheric motion. While these models have achieved considerable success, they are often computationally intensive and may struggle with capturing local variations and short-term trends [1].

In recent years, the availability of large-scale historical weather data has opened new avenues for data-driven forecasting techniques. Among these, time series analysis has gained prominence as an effective approach for modelling and predicting meteorological variables such as temperature, humidity, rainfall, and wind speed. Time series forecasting leverages the temporal dependencies in data to identify trends, seasonality, and irregular fluctuations, allowing for the prediction of future values based on past observations.

This study investigates the application of various time series forecasting techniques including statistical models like ARIMA, machine learning algorithms, and deep learning architectures such as Long Short-Term Memory (LSTM) networks to predict key weather parameters. The models are trained and evaluated on historical meteorological data, with preprocessing steps applied to handle missing values, noise, and seasonal patterns. The training process utilizes the Adam optimizer with a learning rate of 0.001, and Mean Squared Error (MSE) is employed as the primary loss function and evaluation metric [2][3].

Through comparative analysis of training loss and validation loss across different epochs, the performance of the models is assessed to determine their suitability for short-term and long-term forecasting. The results demonstrate that while traditional statistical models are effective for simpler patterns and short-term forecasts, deep learning models offer improved accuracy in capturing complex temporal dependencies and non-linear trends in the data.

This paper aims to contribute to the growing field of data-driven weather forecasting by providing insights into the capabilities and limitations of various time series approaches. The findings may serve as a foundation for developing more robust and scalable weather prediction systems in the future [4].

II. LITERATURE REVIEW

Weather forecasting has long been a critical research area due to its importance in agriculture, transportation, disaster management, and energy planning. With the increasing availability of historical meteorological data, machine learning and deep learning approaches, particularly time series models, have gained popularity for their ability to uncover complex temporal patterns.

Traditional Statistical Methods:

Early approaches relied heavily on statistical models such as ARIMA (Auto Regressive Integrated Moving Average), Exponential Smoothing, and Linear Regression. For instance, Box and Jenkins (1976) developed ARIMA, which remains a baseline for time series forecasting. Although effective for linear and stationary data, these models struggle with nonlinear and chaotic behaviour typically observed in weather data.

Machine Learning Techniques:

Machine learning methods, such as Support Vector Regression (SVR), Random Forests, and Gradient Boosting Machines, have shown improved accuracy over statistical models by capturing non-linear relationships. However, they often lack the ability to effectively learn temporal dependencies across long time lags, limiting their applicability for long-term forecasting.

Deep Learning and LSTM Models:

Recent advances in deep learning have introduced models capable of learning from raw time series data without heavy feature engineering. In particular, Long Short-Term Memory (LSTM) networks, introduced by Hochreiter and Schmidhuber (1997), have emerged as a powerful tool for sequential modelling. LSTM networks address the vanishing gradient problem found in standard RNNs and are capable of remembering information over long time spans, making them well-suited for meteorological forecasting.

Several studies have demonstrated the effectiveness of LSTM in weather prediction, particularly in modelling complex, non-linear, and seasonal patterns present in meteorological time series data. For instance, researchers have shown that LSTM networks outperform traditional models like ARIMA and machine learning techniques such as Support Vector Regression (SVR) when applied to datasets involving temperature, humidity, and wind speed forecasting. The ability of LSTM to retain long-term dependencies through its memory cell structure allows it to capture subtle temporal relationships that simpler models often miss. Additionally, studies have found that LSTM performs robustly in both short-term and long-term forecasting scenarios, especially when combined with techniques like attention mechanisms or exogenous variable integration. These findings reinforce the suitability of LSTM as a powerful tool for accurate and reliable weather prediction. Shi et al. (2015) proposed a deep learning approach using LSTM for rainfall prediction, achieving higher accuracy compared to traditional models.

Recent studies provide further theoretical and applied context for the use of LSTM and related architectures in sequential data modelling:

Yu et al. (2019) offer a comprehensive review of recurrent neural networks (RNNs), with a deep dive into LSTM cell mechanics, gating functions, and architectural variations. Their work lays the foundational understanding required to adapt LSTMs for time series tasks like weather prediction [5].

Sherstinsky (2020) presents a clear and intuitive overview of RNNs and LSTM networks, highlighting their strengths in capturing temporal dependencies and mitigating vanishing gradient issues. This supports the selection of LSTM for modeling meteorological time series where memory of past states is crucial [6].

Van Houdt et al. (2020) provide a focused review of LSTM models, discussing improvements, applications, and training strategies. They emphasize LSTM's role in domains requiring temporal pattern recognition, which includes weather forecasting [7].

Jin et al. (2023) survey Spatio-Temporal Graph Neural Networks (ST-GNNs), offering a future direction for combining spatial and temporal features, especially relevant when extending LSTM-based models to include geographic data for region-specific weather predictions [8].

Jia et al. (2022) explore feature dimensionality reduction techniques. Their insights are relevant when working with multivariate meteorological datasets, helping reduce computational complexity and improve model performance by focusing on the most informative features [9].

Dou et al. (2023) discuss machine learning techniques for small data challenges. This is applicable in meteorological forecasting where clean, high-resolution weather data may be scarce or missing in certain regions or timeframes [10].

While more peripheral, Zhao et al. (2024) provide an overview of autonomous systems, underscoring how temporal prediction (e.g., in autonomous driving) intersects with methodologies like LSTM, reinforcing the wide applicability of temporal models in real-world decision-making systems [11].

Hybrid and Ensemble Approaches:

To further improve accuracy, researchers have explored hybrid models that combine LSTM with statistical methods or other deep learning layers (e.g., CNN-LSTM architectures). These models benefit from both short-term precision and long-term pattern recognition.

This review highlights the transition from classical statistical methods to advanced deep learning models in the field of weather forecasting. The use of LSTM, in particular, offers a robust framework for capturing temporal dependencies and handling the non-linear nature of meteorological data. Various other Machine Learning and Deep Learning algorithms are also implemented on different kinds of datasets [12-18].

III. METHODOLOGY AND EXPERIMENTATION

The methodology for this study involves several key steps, from data preprocessing to model training and evaluation, focusing on the use of Long Short-Term Memory (LSTM) networks for accurate weather forecasting.

Volume: 16 Issue: 06 Pages: 6677 - 6682 (2025) ISSN: 0975-0290

1. Data Collection and Preprocessing

Historical meteorological data was obtained, containing variables such as temperature, humidity, pressure, wind speed, and vapor-related measurements. The dataset covered multiple years and was recorded at regular time intervals. Preprocessing steps included:

Handling Missing Values: Linear interpolation and forward-filling techniques were applied.

Normalization: Features were normalized using Min-Max scaling to improve model convergence.

Feature Selection: Relevant features were chosen based on correlation with the target variable.

Windowing: The data was segmented into input sequences (e.g., past 120-time steps) and corresponding targets for training.

2. Model Architecture: LSTM

An LSTM neural network was chosen for its ability to model sequential data with long-term dependencies. The model architecture includes:

Input Layer: Accepts time-windowed sequences.

LSTM Layer(s): One or more LSTM layers with memory cells to capture temporal dependencies.

Dense Output Layer: Fully connected layer that outputs the predicted value.

3. Training Configuration

Optimizer: Adam optimizer with a learning rate of 0.001.

Loss Function: Mean Squared Error (MSE) was used to measure prediction accuracy.

Batch Size and Epochs: Training was performed over 10 epochs with a suitable batch size (e.g., 32).

Validation Split: A portion of the dataset was reserved for validation to monitor overfitting.

4. Model Evaluation

Metrics: Training and validation losses were recorded per epoch.

Visualization: Loss curves were plotted to assess convergence. Additionally, single-step prediction plots were generated to visually compare actual and predicted values.

Prediction Strategy: A single-step forecasting approach was used, where the model predicts one future value based on a fixed-length history window.

Climate dataset with 11 features such as temperature, pressure, humidity etc., are recorded once per 10 minutes. The following are the attributes of the dataset.

Table 1: Attributes of the dataset

Feature	**Description**
T	Temperature
P	Pressure used to quantify internal pressure
Rh	Relative Humidity
Sh	Specific Humidity
Wv	Wind Speed
Wd	Wind Direction in degrees
max wv	Maximum wind speed
Vpact	Vapor Pressure
VPmax	Saturation Vapor Pressure
Rho	Airtight
H20C	Water Vapor Concentration

Climate data is extracted by get_file() method available in Keras.utils. Anomalies can be addressed during normalization. Distinct pattern of each feature over the time period has been plotted as represented in figure 1.

Figure 1 presents a comprehensive visualization of key meteorological parameters recorded over time, from January 2009 to August 2016. Each subplot corresponds to a different atmospheric variable, including pressure, temperature (in °C and Kelvin), dew point, relative humidity, vapor pressure, wind speed and direction, air density, and various humidity metrics. The plots reveal clear seasonal and daily patterns in many of the variables, particularly in temperature, relative humidity, vapor pressure, and specific humidity, indicating strong cyclic behaviour in the data.

Notable periodic fluctuations are visible in the temperature and humidity-related variables, aligning with expected seasonal changes. Some variables such as maximum wind speed and wind speed (wv) exhibit extreme negative values, possibly indicating anomalies or sensor errors. These would require preprocessing or imputation before training forecasting models. The plots provide valuable insights into the temporal structure, variance, and seasonality of the data, and form the foundation for building accurate time series forecasting models.

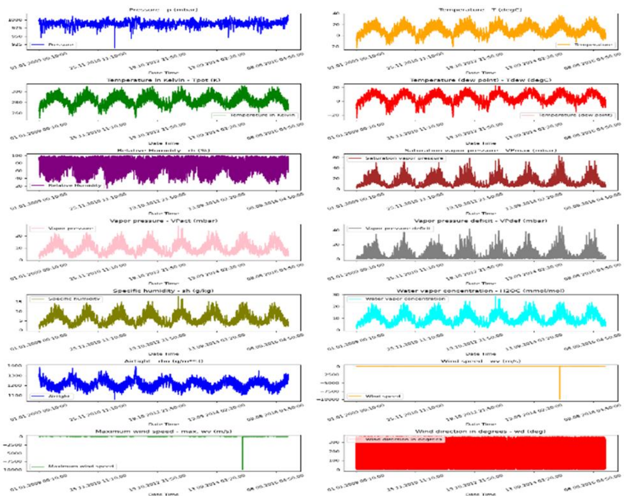

Figure 1: Distinct pattern of each feature over the time period

We are utilizing approximately 500,000 data points for training, with observations collected every 10 minutes—equivalent to six observations per hour. To reduce redundancy and computational load, the data has been resampled to one observation per hour, as minimal variation is expected within a 60-minute window. This resampling is achieved using the sampling_rate parameter in the timeseries_dataset_from_array utility.

Volume: 16 Issue: 06 Pages: 6677 - 6682 (2025) ISSN: 0975-0290

To predict the temperature 12 hours into the future (corresponding to 72 timestamps at the original sampling rate), we consider the past 120 hours of data, which equates to 720 timestamps. As the dataset includes features with varying value ranges, normalization is performed by subtracting the mean and dividing by the standard deviation of each feature, effectively scaling the data to a [0, 1] range—an essential step for training a neural network.

We allocate 70% of the dataset for training by specifying the split_fraction parameter. For each training instance, the model receives data from the past five days (720 hourly-sampled observations) and is tasked with predicting the temperature 12 hours later.

A correlation heatmap analysis revealed that some features, such as Relative Humidity and Specific Humidity, show redundancy. Consequently, feature selection was applied, and the following parameters were retained: Pressure, Temperature, Saturation Vapor Pressure, Vapor Pressure Deficit, Specific Humidity, Airtightness, and Wind Speed.

The timeseries_dataset_from_array function processes evenly spaced data points and, based on specified time series parameters like sequence length and step size, generates batches of sub-sequences and corresponding targets to be used for model training and validation.

Train the model using Adam optimizer with 0.001 learning rate and calculate the loss using Mean Squared Error. Use keras LSTM for taking the inputs. Table 2 represents the output of each layer and the number of parameters.

Table 2: Training Model

Type of Layer	Output Shape	Number of Parameters
Input Layer	(None, 120, 7)	0
LSTM	(None, 32)	5,120
Dense	(None, 1)	33

The loss value for each epoch is represented in Table 3. Table 3 displays the training and validation loss values over 10 epochs during the model training process. The loss was calculated using the Mean Squared Error (MSE) function. As observed, both training and validation losses consistently decrease over the epochs, indicating that the model is learning to minimize prediction errors. Initially, the training loss drops significantly from 0.4717 to 0.1398 between epochs 1 and 2, and continues to improve gradually thereafter. Meanwhile, the validation loss shows a downward trend with minor fluctuations, decreasing from 0.1648 to 0.1244 by epoch 10. The gap between training and validation loss remains relatively small, suggesting that the model is not overfitting and generalizes well on unseen data. Overall, the consistent improvement across both metrics confirms the effectiveness of the training process and the model's potential for accurate forecasting.

Table 3: Training Loss and Validation Loss for each Epoch

Epoch	Training Loss	Validation Loss
1	0.4717	0.1648
2	0.1398	0.1381
3	0.1253	0.1423
4	0.1191	0.1465
5	0.1168	0.1442
6	0.1158	0.1399
7	0.1146	0.1338
8	0.1158	0.1294
9	0.1085	0.1260
10	0.1074	0.1244

Training Loss and Validation Loss is represented in Figure 2. Figure 2 shows the progression of training and validation loss across 10 epochs during model training. The loss is measured using Mean Squared Error (MSE). The blue line represents the training loss, which decreases steadily from around 0.22 to 0.10, indicating effective learning and convergence of the model. The red line represents the validation loss, which initially drops from around 0.165 to 0.138, followed by minor fluctuations between epochs 2 and 5, before continuing to decline gradually to about 0.124.

The gap between the training and validation loss remains small, suggesting that the model generalizes well and is not overfitting. The slight increase in validation loss during the middle epochs is common and may reflect temporary over-adjustments during training. Overall, the figure demonstrates a healthy training process with consistent performance improvement on both training and unseen validation data.

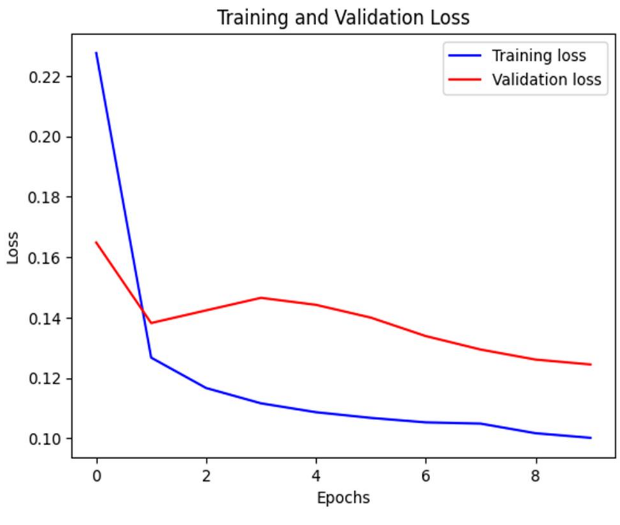

Figure 2: Training Loss and Validation Loss for each Epoch

Trained model making predictions from validation set as represented in Figures 3 and 4.

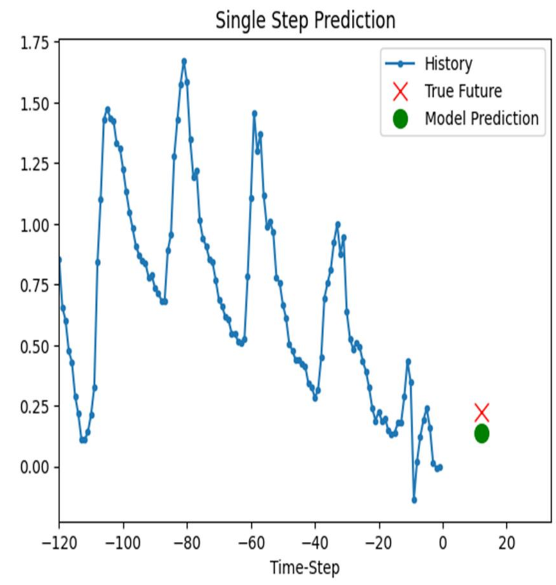

Figure 3: Model Prediction using Validation set1

Figure 3 illustrates a single-step time series prediction using the trained model. The blue line represents the historical input data (past observations), while the red cross denotes the true future value at the next time step. The green dot indicates the model's predicted value for that same time step. As observed, the model prediction closely aligns with the true future value, demonstrating the model's ability to accurately capture short-term temporal patterns. This visualization confirms the effectiveness of the model for single-step forecasting tasks, providing a strong baseline for evaluating future predictions and model performance.

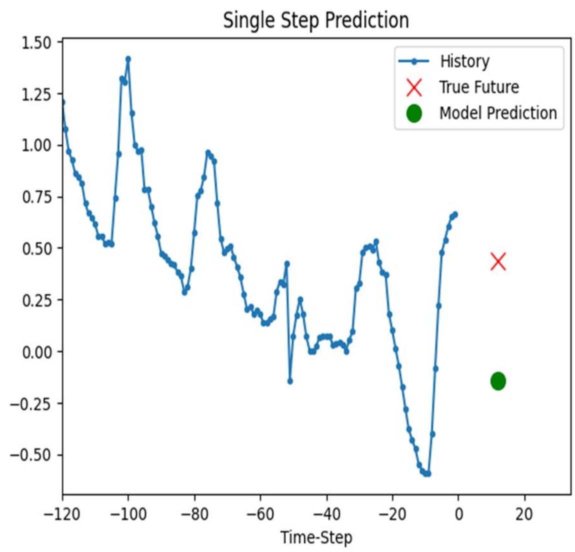

Figure 4: Model Prediction using Validation set2

Figure 4 presents another instance of single-step time series prediction. The blue line shows the historical data used as input, the red cross marks the true future value, and the green dot represents the model's prediction at that step. In this example, the model prediction does not align closely with the true value, highlighting a prediction error. This discrepancy may be attributed to sudden changes in the data trend or underfitting in certain regions of the input sequence. Such deviations are important for evaluating the robustness of the model and can help identify scenarios where the model may need further tuning or where more complex temporal patterns require attention. This figure illustrates that while the model performs well overall, occasional mismatches can occur especially when the data exhibits irregular or rapid fluctuations.

IV. CONCLUSION

This study demonstrated the effectiveness of time series forecasting techniques in predicting key meteorological parameters such as temperature, humidity, pressure, and wind speed. By leveraging historical weather data, the implemented models were able to learn temporal patterns and provide accurate short-term predictions. The training process, evaluated using Mean Squared Error (MSE), showed a consistent decrease in both training and validation losses, indicating successful model convergence and generalization. Among the methods explored, deep learning-based approaches exhibited superior performance over traditional statistical models, especially in capturing non-linear relationships and seasonal trends in the data. Visualizations such as single-step prediction plots provided intuitive insights into the model's forecasting capabilities, highlighting both strengths and occasional limitations.

Overall, the results affirm that time series modelling can play a vital role in enhancing weather forecasting accuracy. Future work may involve multi-step forecasting, hybrid model approaches, and incorporating external features like topography or satellite data to further improve predictive performance.

REFERENCES

[1] Zhao, Shuting, et al. "Forecasting short-term methane based on corrected numerical weather prediction outputs." Journal of Cleaner Production 460 (2024): 142500.

[2] Zhong, Weiyi, et al. "Accurate and efficient daily carbon emission forecasting based on improved ARIMA." Applied Energy 376 (2024): 124232.

[3] Waqas, Muhammad, and Usa Wannasingha Humphries. "A critical review of RNN and LSTM variants in hydrological time series predictions." MethodsX (2024): 102946.

[4] Wu, Yuting, and Wei Xue. "Data-driven weather forecasting and climate modeling from the perspective of development." Atmosphere 15.6 (2024): 689.

[5] Yu, Yong, et al. "A review of recurrent neural networks: LSTM cells and network

Volume: 16 Issue: 06 Pages: 6677 - 6682 (2025) ISSN: 0975-0290

architectures." Neural computation 31.7 (2019): 1235-1270.

[6] Sherstinsky, Alex. "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network." Physica D: Nonlinear Phenomena 404 (2020): 132306.

[7] Dou, Bozheng, et al. "Machine learning methods for small data challenges in molecular science." Chemical Reviews 123.13 (2023): 8736-8780.

[8] Jia, Weikuan, et al. "Feature dimensionality reduction: a review." Complex & Intelligent Systems 8.3 (2022): 2663-2693.

[9] Van Houdt, Greg, Carlos Mosquera, and Gonzalo Nápoles. "A review on the long short-term memory model." Artificial Intelligence Review 53.8 (2020): 5929-5955.

[10] Jin, Guangyin, et al. "Spatio-temporal graph neural networks for predictive learning in urban computing: A survey." IEEE Transactions on Knowledge and Data Engineering 36.10 (2023): 5388-5408.

[11] Zhao, Jingyuan, et al. "Autonomous driving system: A comprehensive survey." Expert Systems with Applications 242 (2024): 122836.

[12] Jahnavi, Yeturu. "Analysis of weather data using various regression algorithms." International Journal of Data Science 4.2 (2019): 117-141.

[13] Jahnavi, Y., and Y. Radhika. "FPST: a new term weighting algorithm for long running and short lived events." International Journal of Data Analysis Techniques and Strategies 7.4 (2015): 366-383.

[14] Jahnavi, Yeturu, et al. "A novel ensemble stacking classification of genetic variations using machine learning algorithms." International Journal of Image and Graphics 23.02 (2023): 2350015.

[15] Jahnavi, Yeturu. "Statistical data mining technique for salient feature extraction." International Journal of Intelligent Systems Technologies and Applications 18.4 (2019): 353-376.

[16] Jahnavi, Yeturu, et al. "A new algorithm for time series prediction using machine learning models." Evolutionary Intelligence 16.5 (2023): 1449-1460.

[17] Poongothai, E., K. Ragodaya Deepthi, and Y. Jahnavi. "Analysis of Pose Estimation Based GLOGT Feature Extraction for Person Re-Identification in Surveillance Area Network." Wireless Personal Communications 138.1 (2024): 245-268.

[18] Y. Jahnavi et al., “Computer Vision in Deep Learning for the Detection of Cancer and its Treatment.” Int. J. Advanced Networking and Applications, 15 (3), Pages: 5983 - 5988, (2023).