MAUSAM, **76**, 3 (July 2025), 717-732

MAUSAM

DOI: https://doi.org/10.54302/mausam.v76i3.6863
Homepage: https://mausamjournal.imd.gov.in/index.php/MAUSAM

UDC No. 551.509 : 004.032.26 (211.213)

Comparative analysis of recurrent neural networks for weather prediction in the Antarctic region

V. SAKTHIVEL SAMY[1*] and VEENA THENKANIDIYOOR[2]

[1]*National Centre for Polar and Ocean Research, Goa, India*

[2]*National Institute of Technology Goa, Goa, India* (**veenat@nitgoa.ac.in**)

(*Received 10 October 2024, Accepted 15 April 2025*)

***Corresponding author's email: vssamy@ncpor.res.in.**

सार – संख्यात्मक मौसम पूर्वानुमान एक सुस्थापित पद्धति है जो वायु, तापमान, दबाव और आर्द्रता समीकरणों को हल करने के लिए वर्तमान वायुमंडलीय स्थितियों को इनपुट के रूप में उपयोग करती है। यह अध्ययन भारती स्टेशन, अंटार्कटिका से ऐतिहासिक डेटा का उपयोग करके मौसम संबंधी पूर्वानुमान के लिए गहन अध्ययन के उपयोग की जांच करता है। गहन अध्ययन फ्रेमवर्क का उपयोग करके विभिन्न अद्वितीय आवर्तक तंत्रिका नेटवर्क मॉडल विकसित किए गए और अगले 24 से 48 घंटों की मौसम स्थितियों के पूर्वानुमान करने के लिए प्रशिक्षित किए गए। हमारे प्रस्तावित दृष्टिकोण की प्रभावशीलता की तुलना अत्याधुनिक न्यूरल नेटवर्क एल्गोरिदम से की गई और परिणाम बेहतर पूर्वानुमान प्रदर्शित करते हैं। इस अध्ययन में, ट्रांसफॉर्मर मॉडल में सबसे कम रूट मीन स्क्वायर त्रुटि (RMSE) 0.000478 है, जो इसे जांचे गए न्यूरल नेटवर्क में सबसे कुशल मॉडलों में से एक बनाता है। यह प्रगति एक अधिक कुशल विकास प्रक्रिया की सुविधा प्रदान करती है, जो बदले में, मौसम पूर्वानुमान की सटीकता को बढ़ाती है।

ABSTRACT. Numerical weather prediction is a well-established method that uses current atmospheric conditions as inputs to solve wind, temperature, pressure and humidity equations. This study examines the use of deep learning for meteorological forecasting using historical data from the Bharati Station, Antarctica. Different unique recurrent neural network models have been developed using a deep learning framework and explicitly trained to predict the weather conditions of the next 24 to 48 hours. The effectiveness of our proposed approach is compared against state-of-the-art neural network algorithms, and the results demonstrate better forecasting performance. In this study, the Transformer model has the lowest Root Mean Square Error (RMSE) of 0.000478, making it one of the most efficient models in the neural networks investigated. This progress facilitates a more efficient development process, which, in turn, enhances the accuracy of weather forecasts.

Key words – Weather Forecasting, Machine Learning, Deep Learning, Neural Networks, Polar Weather Data, Data Science.

1. Introduction

In the Antarctic region, scientific research and logistics activities are strongly reliant on the precision and reliability of environmental forecasting systems for efficient operation and coordination. Predicting temperature remains a significant area of interest. However, the unique and complex geography of East Antarctica presents substantial challenges for real-time forecasting in smaller, localized areas. Existing meteorological stations and sensor data processing methods mainly rely on numerical modelling techniques.

The journey of weather forecasting began in 1922 when Lewis Fry Richardson attempted manual numerical weather prediction (NWP) in Britain (Bauer *et al*. 2015). By 1950, computer-assisted weather forecasts were produced, marking a significant milestone in meteorology (Lynch, 2008). These forecasts quickly became operational in countries like Sweden, the United States, and Japan. Over time, NWP methods (Bauer *et al*. 2015) steadily evolved. In contrast, the development of machine learning (ML) (Schmidhuber, 2015), particularly neural networks (NNs) (McCulloch and Pitts, 1943), followed a more tumultuous path. McCulloch and Pitts proposed the initial concept of NNs in 1943 (McCulloch & Pitts, 1943), but it was not until the invention of back-propagation in the 1970s that significant advancements were made, leading to the second wave of ML applications (Linnainmaa, 1970; LeCun, 1988). Polar meteorology

(Radok, 1979) plays a significant role in the global climate system. In addition, global warming has renewed interest in polar research, as changes in the Polar Regions and sea ice significantly affect the movement of the atmosphere. The term "weather" describes the recurrent patterns of variation in the Earth's climate at a specific location and time Lal *et al*. 2006. It is an ongoing process that is multifaceted, chaotic, and data intensive. Weather forecasting models today rely heavily on physical model performance, which needs substantial computing systems (Fathi *et al*. 2022) for intensive computations. In recent years, weather data has become increasingly influential in our daily lives.

This study is motivated by frequent sudden and unexpected atmospheric changes in Antarctica and Polar Regions. Predicting these changes could provide significant benefits. Weather forecasting uses qualitative data from the existing environment to predict the future state of the atmosphere. It is one of the world's most challenging issues. Weather forecasting is real-time forecasting in which model capabilities are required, among other things, for daily or weekly forecast schedules. As a result, the accuracy of the data is crucial in making this prediction (Srivastava *et al*. 2004); expensive, technically complex models often fail due to inaccurate observations and a lack of understanding of how the atmosphere operates. Moreover, solving these models takes time. Because of its significance in the public and social spheres, innumerable intellectuals and scientists from various disciplines have studied foresight (Kulandaivelu and Dang, 2003).

The application of ML in meteorology has increased dramatically over the last decade. The NNs and DL, in particular, have seen unprecedented utilization. To fill the shortage of resources covering NNs with a meteorological lens, many articles and studies have suggested weather forecasting techniques using ML and DL to predict the weather and help solve weather forecasting problems. Numerous traditional strategies can be utilized to enhance the model accuracy because the data used in weather forecasting is nonlinear and notices some unusual patterns and trends. However, they cannot always accurately predict the weather (Mohammed *et al*. 2018). They applied both a linear regression and a functional linear regression model. They found that operational weather forecasting services beat both models for up to seven-day forecasts.

On the other hand, their model did well when forecasting over more ex-tended periods. Krasnopolsky and Rabinowitz suggested a hybrid model that uses NNs to simulate the physics of weather forecasting (Krasnopolsky and Fox-Rabinovitz, 2006). Radhika and Shashi (2009) used Support Vector Ma-chines (SVM) for weather prediction as a classification issue (Lütkepohl, 2013). They suggested a data mining-based forecasting algorithm for identifying shifting trends in meteorological conditions. Patterns from prior data are used to forecast the following weather conditions. The suggested data model predicts meteorological conditions using the Hidden Markov Model and extracts them using k-means clustering. Grover et al. studied weather forecasting using a hybrid technique that combined discriminatory model training with deep neural networks to mimic a combination of statistics for various weather-related variables (Guidotti *et al*. 2018).

Weather encompasses the state of the Earth's atmosphere at a particular location and time and is a complex and dynamic system that presents numerous challenges for accurate forecasting. This multifaceted process is driven by continuous data inputs and is inherently chaotic, making precise predictions difficult. Forecasters rely on past datasets to predict future conditions, but the complexity and variability of the atmosphere often lead to inaccuracies. Weather prediction is considered one of the most formidable scientific and technological challenges, demanding precise and timely insights from meteorologists. Modern weather forecasting primarily utilizes intricate physical models that require high-performance computing systems to operate effectively. These powerful computers are essential for solving the complex equations that describe atmospheric behavior. Despite the substantial investment in these advanced technologies, forecasts can still be inaccurate due to insufficient initial observations and the inherent difficulty in fully understanding atmospheric dynamics. Additionally, the time-consuming nature of working with such sophisticated models further complicates the forecasting process.

1.1. *Research objectives*

The novelty of this methodology is the use of extensive historical data from Bharati Station, Antarctica, to create age-free direct input-output network models. This method is entirely data-based, the model is validated using historical and current data and a Transformer model, including the LSTM variant, is trained for temperature prediction.

The specific objectives are as follows:

(*i*) Development of a framework for a deep neural network to use historical weather data to forecast specific weather features over the desired period.

Fig. 1(a). Bharati Station, Antarctica

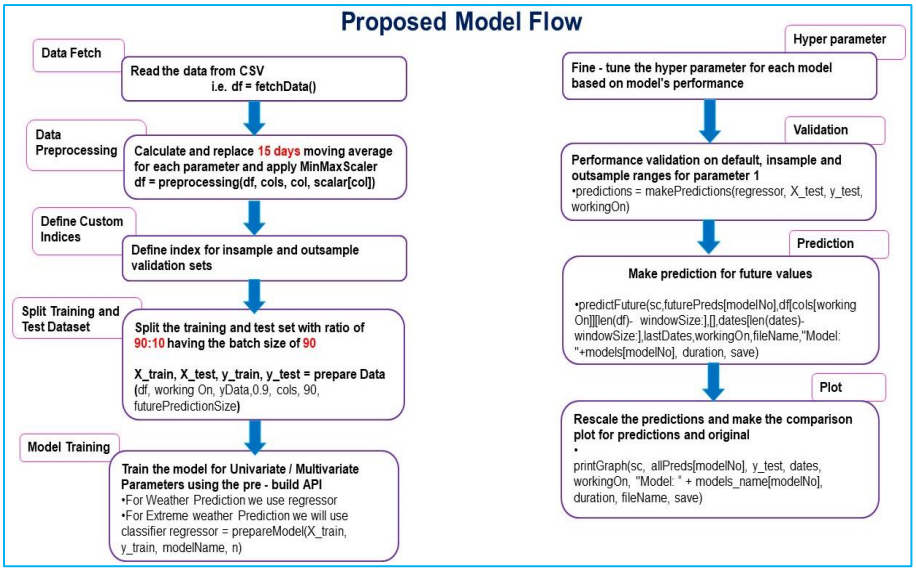

Fig. 1(b). This represents the proposed research model flow for predicting weather parameters, such as temperature, by applying a 15-day moving average and training the model using a 70:30 split *i.e.*, training : 70, validation:20 and testing:10.

(*ii*) Design a model to predict the temperature evolution over 24 hours at Bharati Station in Antarctica.

(*iii*) Analysis of forecast errors, emphasize seasonal variations and forecast duration.

(*iv*) Prepare and curate polar weather datasets for this study, which will be made available to the scientific community upon the publication of this article.

Our research aims to improve weather forecasting by developing and evaluating deep neural network models. The study enhanced the deep learning (DL) model and rigorously evaluated the Transformer using attention mechanisms, while comparing it to LSTM-Multi-input Multi-output (MIMO) and LSTM-One-Input Multioutput (SIMO) models.

2. Data and methodology

2.1. *Study areas*

Fig. 1(a) depicts the geographical location of Bharati Station, Antarctica (69.40° S, 76.00° E), situated

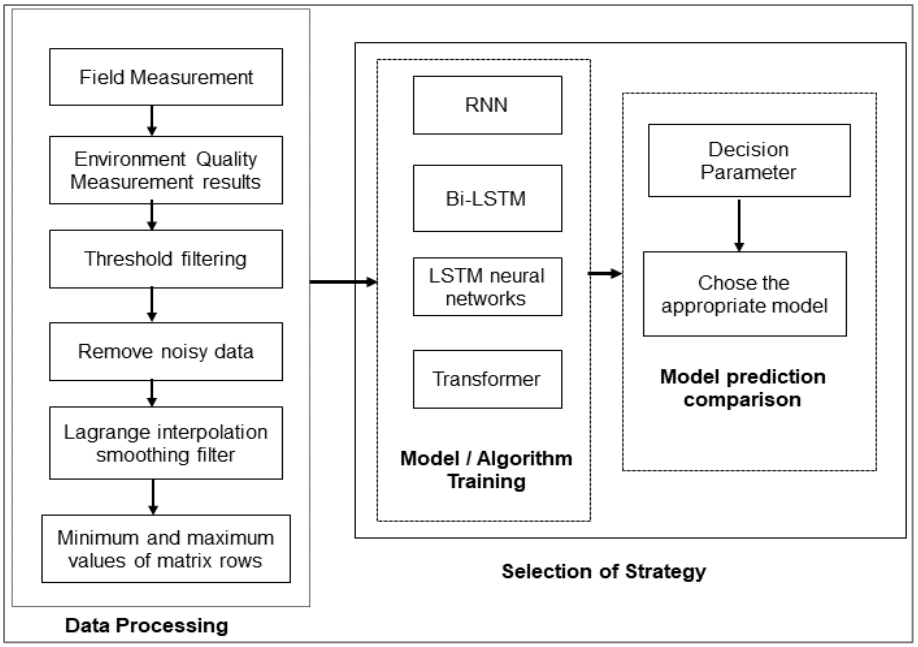

Fig. 1(c). Data cleaning and standardization prior to input into the network, followed by machine learning training

approximately 3000 km to the east of Maitri, Thala Fjord, and Quilty Bay and east of the Stornes Peninsula, at coordinates 69° 24.41' S, 76° 11.72' E. The station is positioned at an elevation of approximately 35 meters above sea level. The station with a very small footprint was commissioned on 18 March, 2012 to facilitate year-round scientific research activity by the Indian Antarctic program. The communication is through dedicated satellite channels providing connectivity for voice, video and data with India mainland. It serves as a modern research facility focused on various disciplines, including climate change, atmospheric sciences, and biological research. Meteorological observations are needed to analyse the course of wind and temperature changes in an area.

2.2. *Proposed models*

Most existing weather forecasting approaches predominantly rely on ML and DL algorithms, yet they still exhibit certain inherent flaws to be addressed (Bochenek and Ustrnul, 2022). ML, a branch of data science, generates models from training datasets (Mohammed and Kora, 2023). Each record within the dataset is assigned appropriate weights (typically between 0 and 1) for each variable, indicating how each variable is connected to the target value. Sufficient training data is necessary to calculate the optimal weights for all variables. When these weights are learned accurately, the model can predict the correct output or target value based on test data.

Weather forecasting is a highly researched topic within the fields of Artificial Intelligence (AI) and ML. Various classic and hybrid techniques are employed in weather forecasting. Models referenced in Yu *et al*. (2017) can be constructed utilizing both ML and DL methodologies. Specifically, these include Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks. This diverse array of approaches facilitates the exploration of complex patterns and relationships within the data, enhancing predictive accuracy and model robustness. The RNN have been utilized for data modeling and prediction since the 1980s, gradually gaining widespread adoption. However, RNNs exhibit several limitations, particularly in handling long-range dependencies, which render them unsuitable for many applications. To address the issue of vanishing gradients, Long Short-Term Memory (LSTM) networks were introduced. Subsequently, numerous LSTM variants, such as Bidirectional LSTM (Bi-LSTM), Stacked Autoencoder LSTM (SAE LSTM) and others, have been developed to enhance the efficiency and effectiveness of feature learning.

In this study, we performed a comparative analysis of several deep neural network models, specifically highlighting different variations of LSTM networks (Hewage *et al.* 2021). This investigation aimed to evaluate their performance and effectiveness in capturing temporal dependencies and patterns within the dataset. They developed these deep neural network models and

TABLE 1

Shows the sample Automatic Weather Station datasets from Bharati Station, Antarctica, used in this study, covering the period from 2015 to 2022

OBSTIME	TEMP	AP	WS	WD	RH
11/1/2015 0:00	-12.52	973.92	29.3	86.38	43.13
11/1/2015 1:00	-11.88	973.54	26	87.63	44.22
11/1/2015 2:00	-10.6	973.14	24.8	86.27	43.35
11/1/2015 3:00	-9.46	972.51	23.96	87.77	43.95
11/1/2015 4:00	-7.8	971.8	20.9	72.77	44.03

evaluated their performance in relation to their respective Root Mean Square Error (RMSE) values. Subsequently, we fine-tuned each model with respect to our dataset. All models, including the neural network models, were trained over a minimum of 200 epochs to ensure robust performance. We evaluated two different forecasting models and its flow is shown in Fig. 1(b). In the first configuration, we utilized 80% of the dataset for training and reserved 20% for testing, with predictions extended to 24 to 48 hours ahead. In the second configuration, 90% of the data was allocated to training and 10% to testing, with the same 24 to 48-hour forecast prediction. The results indicated that the first configuration achieved higher predictive accuracy than the second.

2.3. *Data details and pre-processing*

In Antarctica, the automatic weather station datasets/parameters such as (*i*) Air Temperature (TEMP) (*ii*) Air Pressure (AP), (*iii*) Wind Speed (WS), (*iv*) Wind direction (WD), (*v*) Relative Humidity (RH) are regularly observed at Bharati station (69° 24' 41" S, 76° 11'72" E). Using the data collected over the years, data analysis was done on the data count of Bharati station from 2015-22.

2.4. *Preprocessing and experimental setup*

The datasets used in this study require preparation before they can be used for model training. The data missing values are the first to be resolved. In order to solve these missing values, some preprocessing techniques have been applied. Data loss and error are inevitable due to data collection problems and work to prepare data before entering the neural network is known as data cleaning or preprocessing. The steps of this process are indicated in Fig. 1(c). Initially, a constant atmospheric value of 999.9 was assigned to the missing values of parameters such as TEMP, WS, WD, RH and AP. In response, various preprocessing methods have been used to eliminate errors and ensure that all columns contain numerical values. The program eliminates the noise from original data and filters irrelevant information. In order to handle the missing value, the study uses an interpolation technique where functions replace the missing value with an estimate based on the known value around the missing data point. In Table 1, the sample Automatic Weather Station datasets from Bharati Station, Antarctica, used in this study, covering the period from 2015 to 2022.

Experimental Setup : We evaluated the performance of the developed model using the following AI - based High Performance Computing facility of Ministry of Earth Sciences (MoES) available at Indian Institute of Tropical Meteorology (IITM), Pune.

(*i*) HPE XL675d Gen10+ CTO Server, node 1,

(*ii*) AMD EPYC 7402 processor,

(*iii*) 24 core processor for a total of 96 cores,

(*iv*) 1024 GB RAM and 97 TB storage capacity,

(*v*) Cent OS Linux version 7.9.2009, scheduler PBS 19.1.3

For executing the proposed model, we employed NVIDIA CUDA 11.3, Docker 20.10.7, Tensor Flow and Kubernetes.

2.5. *Long Short-Term Memory*

The proposed model is based on LSTM networks and utilizes time-series weather data to forecast weather conditions. As outlined in Section 2.1, we specifically explored the capabilities of a variant of LSTM, a type of recurrent neural network designed to handle sequential data such as time-series. LSTM networks are chosen for their ability to capture long-term dependencies through specialized memory cells. These memory cells enable the network to remember past information, forget irrelevant details, and update current states dynamically (Jozefowicz

et al. 2015; Hinton, 2006; Gulli & Pal, 2017). The Stacked LSTM is a DL architecture with multiple LSTM layers stacked sequentially. Each layer processes information from different temporal scales, allowing the model to learn complex patterns in time-series data. The output of one LSTM layer serves as input to the next, enabling the network to learn hierarchical representations of temporal dependencies.

Stacked LSTM is crucial in time-series forecasting because it captures long-term dependencies and intricate patterns across various time scales. This is particularly effective in tasks where capturing temporal relationships is essential for accuracy.

We handle univariate series in the model development phase with a single feature per variable. When constructing the dataset using the split_sequence() method, we specify the number of time steps as input. Additionally, the input_shape parameter in the first hidden layer's specification defines the input shape for each sample. Given the typically large number of samples, the model anticipates the input component of the training data to adhere to the illustrated size or format.

(*i*) During model fitting, we fine-tune several hyperparameters, batch size, the number of data sequences we send simultaneously.

(*ii*) *Window size* : We consider the number of days that we expect to predict the temperate environment of our case.

(*iii*) *Units* : The unit used in our LSTM cell.

(*iv*) *Epochs* : This is the number of iterations (forward & backward propagation) that our model needs to do.

The proposed study aims to advance weather forecasting by developing and testing deep neural network models tailored for regression tasks. This entails adjusting and fine-tuning these models to properly handle complicated, non-linear interactions in meteorological data in order to increase prediction accuracy. This study improves deep learning models by carefully analyzing LSTM networks in multi-input multi-output (MIMO) and single-input multi-output (SIMO) configurations and Transformer models with attention mechanisms. These models are intended to give a comprehensive approach to weather forecasting.

2.6. *The LSTM-MIMO*

The LSTM-MIMO architecture (Long Short Term Memory - Multiple Input Multiple Output) is characterized by two LSTM layers with multiple inputs and 128 units each but shows higher efficiency than traditional LSTM. This progress is due to integrating different parameters, which improves the models' ability to capture complex relationships and dynamic patterns in datasets. These improvements highlight the potential of LSTM-MIMO to increase predictive accuracy and robustness in time series forecasting applications.

Mathematical equations for the stacked multiple input multiple output LSTM architecture layer by layer.

First LSTM Layer

Input Shape : (16, 16, 5) and Output Shape : (16, 16, 128)

Let :

–(X_t) be the input at time step (t) with shape (16, 5)

–(h_{t-1}) be the hidden state from the previous time step with shape (16, 128)

–(C_{t-1}) be the cell state from the previous time step with shape (16, 128)

–(W), (U) and (b) be the weights and biases for the LSTM cell.

The LSTM equations at time step (t) are:

$f_t = \sigma\,(W_f\,X_t + U_f\,h_{t-1} + b_f)$ with shape [16, 128]

$i_t = \sigma\,(W_i\,X_t + U_i\,h_{t-1} + b_i)$ with shape [16, 128]

$\tilde{c}_t = \tanh\,(W_c\,X_t + U_c\,h_{t-1} + b_c)$ with shape [16, 128]

$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$ with shape [16, 128]

$o_t = \sigma\,(W_o\,X_t + U_o\,h_{t-1} + b_o)$ with shape [16, 128]

$h_t = o_t \odot \tanh\,(c_t)$ with shape [16, 128]

First dropout layer

This layer randomly sets a fraction$^{(rate)}$ of input units to 0 at each update during training time to prevent over fitting.

$h_t^{\text{dropout}} = \text{Dropout}\left(h_t, \text{rate} = 0.2\right)$ with shape [16, 16, 128]

Second LSTM Layer

– Input shape : (16, 16, 128) and – Output shape : (16, 128)

This LSTM layer processes the entire sequence and returns only the output of the last time step. The equations are similar to the first LSTM layer but applied on the output of the first LSTM layer.

$h_t^{(2)} = \text{LSTM}\left(h_t^{\text{dropout}}, \text{lstm\_units} = 128\right)$ with shape [16, 128]

Second Dropout Layer

Like the first dropout layer, this sets a fraction$^{(\text{rate})}$ of input units to 0.

$h_t^{(2)\text{dropout}} = \text{Dropout}\left(h_t^{(2)}, \text{rate} = 0.2\right)$ with shape [16, 128]

Dense Layer

This layer takes the output from the second LSTM layer and projects it to the desired output shape.

$Y = \text{Dense}\left[h_t^{(2)\text{dropout}}, \text{uints} = 2\right]$ with shape [16, 2]

Combining these layers, the overall model architecture processes the input sequence with five features through stacked LSTM layers with dropout. It projects the output to have two features at each time step.

2.7. *The LSTM - SISO*

The LSTM-SISO with a single input and output, utilizing two LSTM layers with 128 units each, demonstrates superior performance compared to a basic LSTM. The additional layer enables the model to capture intricate patterns and dependencies in the data effectively, enhancing its accuracy and generalization ability. This increased depth significantly improves the model's performance in tasks that necessitate detailed sequence analysis, such as time series forecasting. The architecture of the developed single input single output - LSTM-SISO model is provided below, layer by layer. This model is specifically designed to predict temperatures for 24 to 48 hours.

Mathematical equations for the stacked LSTM-SISO architecture layer by layer.

The LSTM layer input shape : (16, 16, 1) and its output shape : (16, 16, 128)

Let :

–(X_t) be the input at time step (t) with shape (16, 1)

–(h_{t-1}) be the hidden state from the previous time step with shape (16, 128)

–(c_{t-1}) be the cell state from the previous time step with shape (16, 128)

–(W), (U) and (b) be the weights and biases for the LSTM cell

The LSTM equations at time step (t) are :

$f_t = \sigma\ (W_f\ X_t + U_f\ h_{t-1} + b_f)$ with shape [16, 128]

$i_t = \sigma\ (W_i\ X_t + U_i\ h_{t-1} + b_i)$ with shape [16, 128]

$\tilde{c}_t = \tanh\ (W_c\ X_t + U_c\ h_{t-1} + b_c)$ with shape [16, 128]

$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$ with shape [16, 128]

$o_t = \sigma\ (W_o\ X_t + U_o\ h_{t-1} + b_o)$ with shape [16, 128]

$h_t = o_t \odot \tanh\ (c_t)$ with shape [16, 128]

Dropout layer : This layer randomly sets a fraction rate of input units to 0 at each update during training time to prevent over fitting.

$h_t^{\text{dropout}} = \text{Dropout}\left(h_t, \text{rate} = 0.2\right)$ with shape [16, 16, 128]

Dense Layer : The input shape: (16, 128) and its corresponding output shape is : [16, 1]

This layer takes the output from the LSTM layer and projects it to the desired output shape.

$Y = \text{Dense}\left[h_t^{\text{dropout}}, \text{uints} = 1\right]$ with shape [16, 1]

Combining these layers, the overall model architecture processes the input sequence with one feature through an LSTM layer with dropout. It projects the output to have one feature at each time step.

2.8. *The Bidirectional LSTM-SISO*

The Bidirectional LSTM-SISO contains two bidirectional LSTM layers with 128 units each, resulting in higher performance than the basic LSTM model by using bidirectional processing to capture both past and future contexts in the sequence. This architecture significantly improves the modeling of complex temporal

patterns and leads to more accurate time series predictions. Each layer contains 128 LSTM units, increasing the capacity to capture contextual information and data dependencies in detail. This additional memory enables the model to learn complex relationships and improve its predictive accuracy. As a result, the model efficiently manages and integrates different types of information and generates more robust and reliable predictions than the standard LSTM model.

Mathematical equations for the Bidirectional LSTM-SISO Layers architecture layer by layer.

First Bidirectional LSTM Layer :

$$\vec{h}_t^{(1)}, \vec{c}_t^{(1)} = \text{LSTM}\left[x_t, \vec{h}_{t-1}^{(1)}, \vec{c}_{t-1}^{(1)} \right]$$

$$\overleftarrow{h}_t^{(1)}, \overleftarrow{c}_t^{(1)} = \text{LSTM}\left[x_t, \overleftarrow{h}_{t+1}^{(1)}, \overleftarrow{c}_{t+1}^{(1)} \right]$$

$$h_t^{(1)} = \left[\vec{h}_t^{(1)}, \overleftarrow{h}_t^{(1)} \right]$$

Dropout Layer after the First Bidirectional LSTM Layer:

$$h_t^{(1)} = \text{Dropout}\left[h_t^{(1)}, \text{rate} = 0.2 \right]$$

Second Bidirectional LSTM Layer :

$$\vec{h}_t^{(2)}, \vec{c}_t^{(2)} = \text{LSTM}\left[h_t^{(1)}, \vec{h}_{t-1}^{(2)}, \vec{c}_{t-1}^{(2)} \right]$$

$$\overleftarrow{h}_t^{(2)}, \overleftarrow{c}_t^{(2)} = \text{LSTM}\left[\overleftarrow{h}_t^{(1)}, \overleftarrow{h}_{t+1}^{(2)}, \overleftarrow{c}_{t+1}^{(2)} \right]$$

$$h_t^{(2)} = \left[\vec{h}_t^{(2)}, \overleftarrow{h}_t^{(2)} \right]$$

Dropout Layer after the Second Bidirectional LSTM Layer:

$$h_t^{(2)} = \text{Droupout}\left[h_t^{(2)}, \text{rate} = 0.2 \right]$$

Dense Layer:

$$Y = \text{Dense}\left(h_t^{(2)} \right) = h_t^{(2)} \; W_0 + b_0$$

2.9. *The Transformer Model*

The customized architecture of the Transformer model enables efficient parallel processing and long-distance data dependency handling. In weather forecasting, transformers are effective in capturing complex temporal patterns and correlations in large-scale datasets, improving the accuracy and efficiency of forecasting weather patterns and helping to better prepare and respond to weather-related events.

Transformers outperform encoder decoder LSTMs in forecasts due to their ability to capture long-term dependencies and parallel computations.

(*i*) The self-attention mechanism of transformers allows them to simultaneously focus on different parts of the input sequence and to model complex temporal patterns better than LSTMs processing data sequentially. This results in faster training and calculation times.

(*ii*) Transformers can handle variable-length input sequences without causing disappearing gradient problems commonly seen in LSTMs.

These advantages make transformers particularly suitable for forecasting tasks that require understanding complex relationships over a long period of time. The architecture of the developed transformer model is provided below, layer by layer. This model is specifically designed to predict temperatures for 24-48 hour.

Mathematical equations for the Transformer model architecture layer by layer.

Bidirectional LSTM Layer 1 and Layer 2

$$\vec{h}_t^{(1)} = \text{LSTM}\left[x_t, \vec{h}_{t-1}^{(1)} \right]$$

$$\overleftarrow{h}_t^{(1)} = \text{LSTM}\left[x_t, \overleftarrow{h}_{t+1}^{(1)} \right]$$

$$h_t^{(1)} = \left[\vec{h}_t^{(1)}, \overleftarrow{h}_t^{(1)} \right]$$

$$h_t^{(2)} = \left[\vec{h}_t^{(2)}, \overleftarrow{h}_t^{(2)} \right]$$

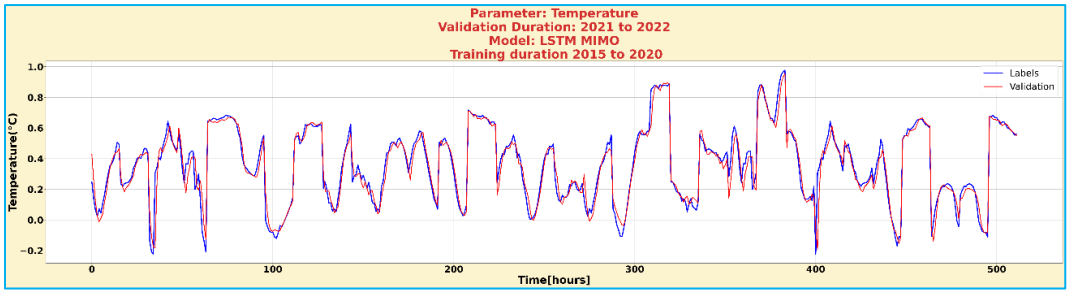

Fig. 2(a). Shows that the graph generated using LSTM-MIMO model for temperature validation, such as the red and the blue lines indicating the actual datasets

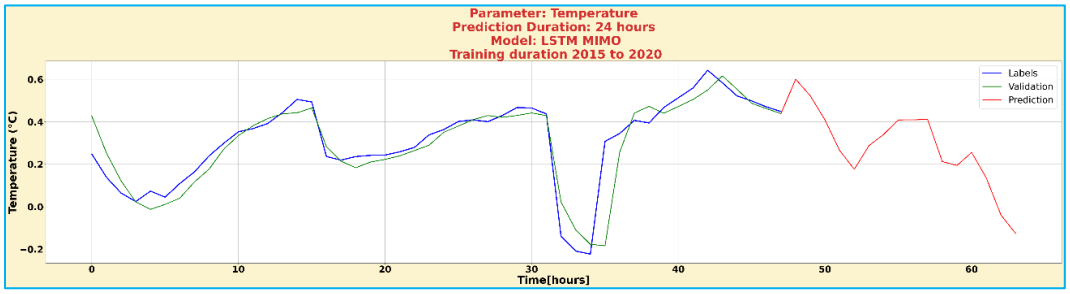

Fig. 2(b). Displays the graph generated by the LSTM-MIMO model for 24-hour temperature prediction, where the red line represents the predicted values, the blue line indicates the actual datasets and the green line illustrates the model's validation

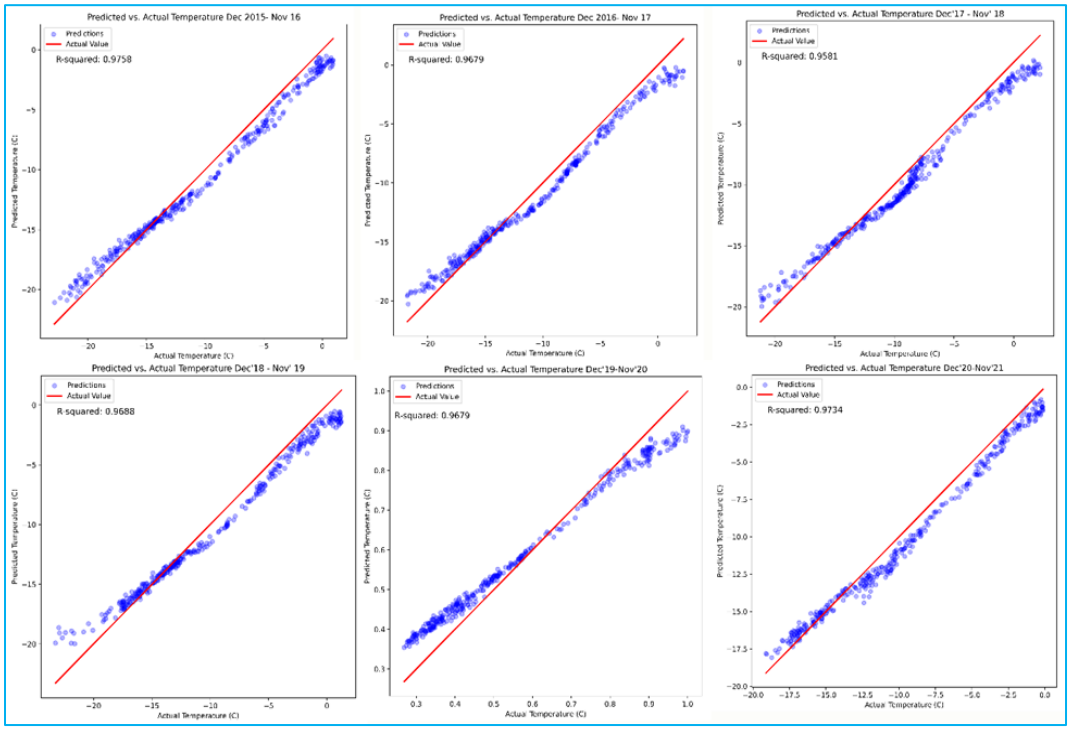

Fig. 2(c). Results of a linear fit between the single-step predicted and actual temperature values for the Bharati station, as obtained from the Transformer Model

TABLE 2

The parameters are used to configure the LSTM-MIMO model to run and predict temperature over a 24-hours period

Model Name	Epochs	Window Size	Hyperparameters	Batch Size	RMSE	Layer 1	Layer 2	Layer 3
LSTM-MIMO	200	16	0.001	32	0.009605	LSTM (256)	LSTM (128)	Dense (2)

Multi-Head Attention Layer

For each multi-head attention block:

$Q = W_Q \cdot x$

$K = W_K \cdot x$

$V = W_V \cdot x$

where (W_Q, W_K, W_V) are the weight matrices for the queries, keys and values respectively.

$$\text{Attention}(Q,K,V) = \text{Software}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where (d_k) is the dimension of the keys.

$$\text{Multi Head}(Q,K,V) = \text{Concat}(\text{head}_1,...,\text{head}_h)W_0$$

where [head$_i$ = Attention (Q_i, K_i, V_i)] and (W_o) is the output weight matrix.

Residual Connection and Layer Normalization

Output$_1$ = Layer Norm {x + Dropout [Multi Head (Q, K, V)]}

Feed Forward Network : FFN (x) = max (0, xW_1 + b_1) W_2 +b_2

where (W_1, W_2) are weight matrices and (b_1, b_2) are biases.

Residual Connection and Layer Normalization:

Output$_2$ = Layer Norm {Output$_1$ + Dropout [FFN (Output$_1$)]}

Global Average Pooling :

$$\text{global\_avg\_pool} = \frac{1}{T}\sum_{t=1}^{T} x_t$$

where (T) is the length of the input sequence.

Output Dense Layer : output = Dense (x) = xW_o + b_0

where (W_o) is the weight matrix and (b_0) is the bias for the output layer.

This model leverages bidirectional LSTMs to capture temporal dependencies, multi-head attention to focus on different parts of the sequence, feed-forward networks for non-linear transformations and finally aggregates information using global average pooling before producing the output.

3. Results and analysis

In this part, we comprehensively evaluate of selected deep neural network techniques trained on the meteorological data. To understand deep models, we usually perform Error Analysis which involves systematically evaluating the errors made by the model(s) to identify patterns, understand limitations, and uncover areas for improvement. This process provides insights into why and where the model underperforms, helping refine the architecture, data, or training strategies. In all our experiments, we use quantitative metrics for performance evaluation which in our case is RMSE. The RMSE is also regularly used to measure the difference between observed and predicted values by any model. It analyzes the distribution of errors to identify consistent trends (*e.g.*, higher errors in specific weather conditions, time periods, or outlier scenarios). The results (below) show that the transformer model outperforms for temperature.

3.1. *The LSTM-MIMO*

The LSTM-MIMO with multiple inputs and two LSTM layers containing 128 units are more effective than basic LSTMs by integrating various parameters, as mentioned at section 2.1. The parameters are used to configure the LSTM-MIMO model to run and predict temperature over 24 hours. Table 2 allows for more rich feature representations and improves the ability of the model to learn complex correlations within the data. Combined with various types of information, the model provides more accurate and robust predictions.

The 24-hour prediction is depicted in red in Fig. 2(b) using LSTM-MIMO model. Fig. 2(a) showcases the validation of the temperature parameter using the LSTM-MIMO model, with its corresponding 24 - hour prediction

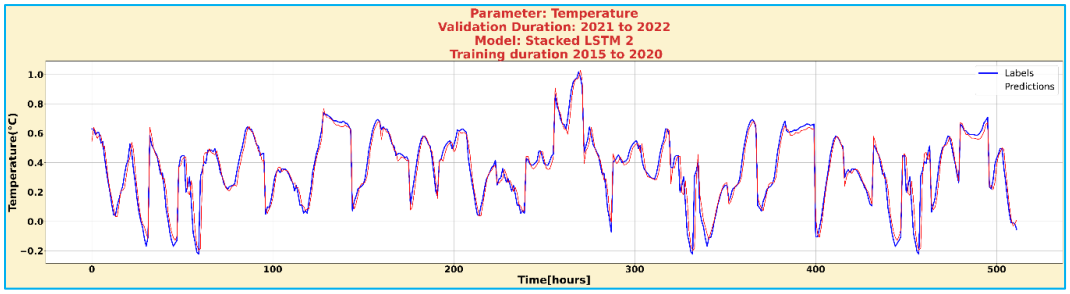

Fig. 3(a). Shows graph generated using LSTM-SISO model for temperature validation, such as red and the blue line indicating the actual datasets

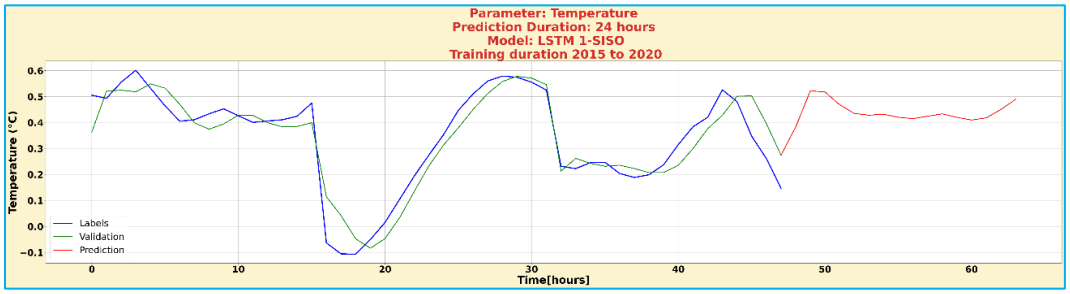

Fig. 3(b). Displays the graph generated by the LSTM-SISO model for 24-hour temperature prediction, where the red line represents the predicted values, the blue line indicates the actual datasets, and the green line illustrates the model's validation

TABLE 3

The parameters are used to configure the LSTM-SISO model to run and predict temperature over a 24-hours period

Model Name	Epochs	Window Size	Hyperparameters	Batch Size	RMSE	Layer 1	Layer 2	Layer 3
LSTM-SISO	200	32	0.001	64	0.041	LSTM (256)	LSTM (128)	Dense (1)

shown in Fig. 2(b). The RMSE values calculated by the model are also presented in Table 2.

3.2. *The LSTM-SISO*

The LSTM-SISO with a single input and output, utilizing two LSTM layers with 128 units each, demonstrates superior performance compared to a basic LSTM. The additional layer enables the model to capture intricate patterns and dependencies in the data effectively, enhancing its accuracy and generalization ability. This increased depth significantly improves the model's performance in tasks requiring detailed sequence analysis, such as time series forecasting.

The 24-hour prediction is depicted in red in Fig. 3(b) using the LSTM-SISO model. Fig. 3(a) showcases the validation of the temperature parameter using the LSTM-SISO model, with its corresponding 24-hour prediction shown in Fig. 3(b). Table 3 presents the parameters used to configure the LSTM SISO model to run and predict temperature over a 24-hour period and the RMSE value calculated by the model.

3.3. *The Bidirectional LSTM-SISO*

The Bidirectional LSTM-SISO contains two bidirectional LSTM layers with 128 units each, resulting in higher performance than the basic LSTM model by using bidirectional processing to capture both past and future contexts in the sequence. This architecture significantly improves the modeling of complex temporal patterns and leads to more accurate time-series predictions. Each layer contains 128 LSTM units, increasing the capacity to capture contextual information and data dependencies in detail. This additional memory enables the model to learn complex relationships and improve its predictive accuracy.

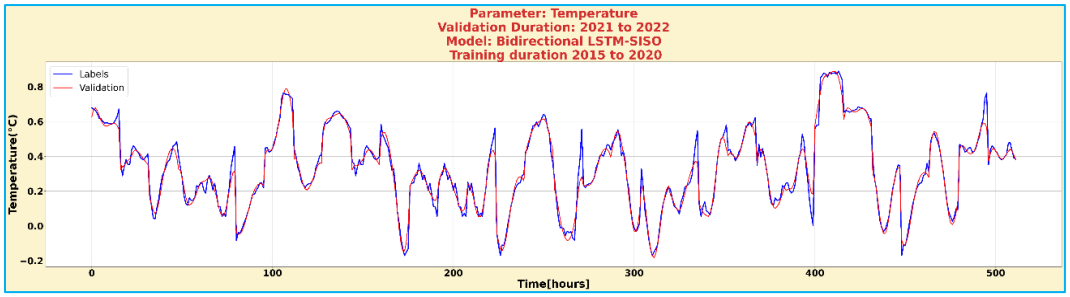

Fig. 4(a). Shows that the graph generated using Bidirectional LSTM-SISO model for temperature validation, such as red and the blue line indicating the actual datasets

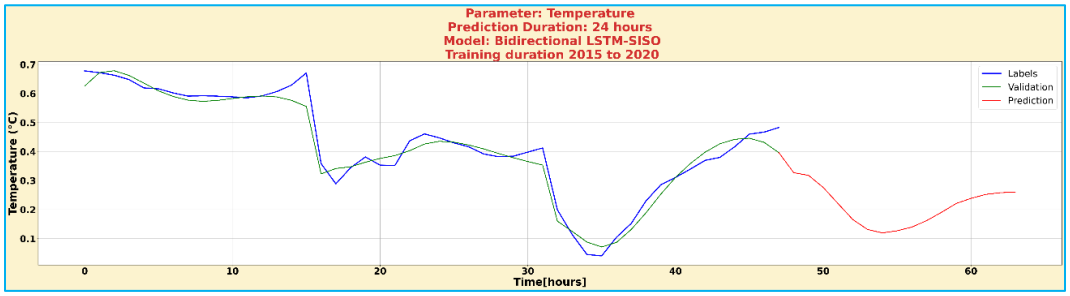

Fig. 4(b). Displays the graph generated by the Bidirectional LSTM-SISO model for 24-hour temperature prediction, where the red line represents the predicted values, the blue line indicates the actual datasets, and the green line illustrates the model's validation

TABLE 4

The parameters are used to configure the Bidirectional LSTM-SISO model to run and predict temperature over a 24-hours period

Model Name	Epochs	Window Size	Hyperparameters	Batch Size	RMSE	Layer 1	Layer 2	Layer 3
Bidirectional LSTM-SISO	200	16	0.001	32	0.000816	LSTM (128)	LSTM (128)	Dense (1)

As a result, the model efficiently manages and integrates different types of information and generates more robust and reliable predictions than the standard LSTM model. The 24-hours prediction is depicted in red in Fig. 4(b), using Bidirectional LSTM-SISO model. Fig. 4(a) showcases the validation of the temperature parameter using the Bidirectional LSTM model, with its corresponding 24-hour prediction shown in Fig. 4(b).

The parameters are used to configure the Bidirectional LSTM model to run and predict temperature over a 24-hours period along with the calculated RMSE value, listed at Table 4.

3.4. *The Transformer Model*

The customized architecture of the Transformer model enables efficient parallel processing and handling of long-distance data dependency. In weather forecasting, transformers effectively capture complex temporal patterns and correlations in large-scale datasets, improve the accuracy and efficiency of forecasting weather patterns, and help to better prepare and respond to weather-related events. The dropout is a widely recognized technique in neural network modeling designed to mitigate over fitting in Table 2.

To predict the 24-hour temperature, Table 5(a) outlines the Transformer model's layer configuration, detailing the specific type of layer flow, the output shape at each stage, and the total number of observation data points used in the prediction process. These are essential for understanding the architecture and performance of the model in temperature forecasting.

The 24-hours prediction is depicted in red in Fig. 5(b), using the Transformer model. Fig. 5(a)

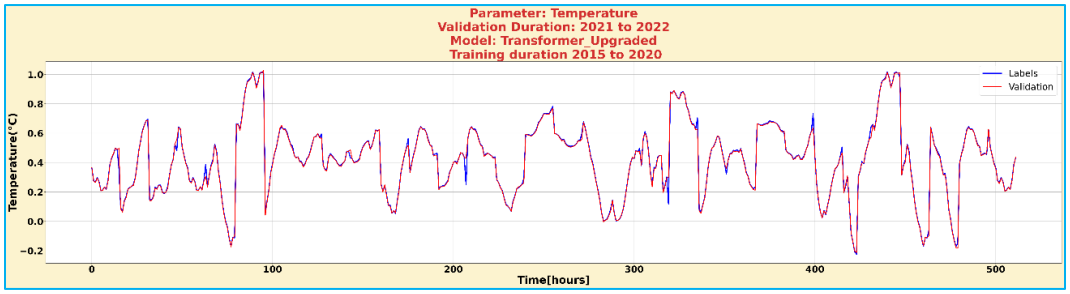

Fig. 5(a). Shows that graph generated using the Transformer model for temperature validation, such as red and the blue line indicating the actual datasets

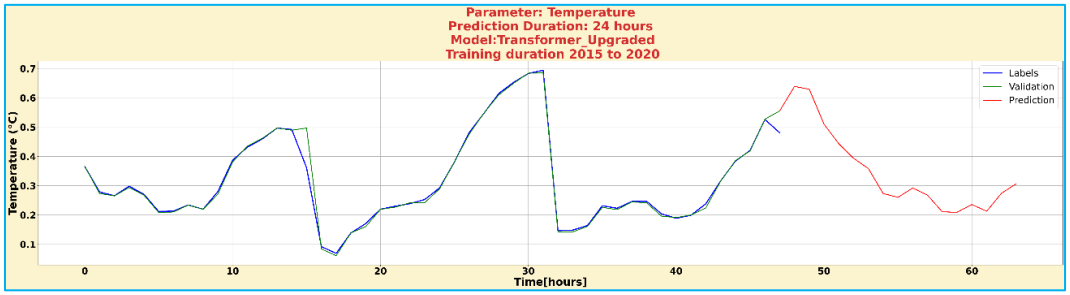

Fig. 5(b). Displays the graph generated by the Transformer model for 24-hour temperature prediction, where the red line represents the predicted values, the blue line indicates the actual datasets, and the green line illustrates the model's validation

TABLE 5(a)

The parameters used in the Transformer for running the model to predict 24 hours temperature

Layer (type)	Output Shape	Parameters
Input Layer	1	0
Bidirectional LSTM 1	256	33,280
Bidirectional LSTM 2	256	98,560
Multi Head Attention	256	4,364
Layer Normalization	256	512
Dense	32	8,224
Dropout	1	0
Layer Normalization_2	256	512
Dense 2	32	8,224
Layer Normalization_3	256	512
Global Average Pooling1D	256	0
Dense 3	1	257

TABLE 5(b)

The overall comparison of the model configuration parameters, along with the calculated RMSE values, is presented

Model Name	Epochs	Window Size	Hyperparameters	Batch Size	RMSE	Layer 1	Layer 2	Layer3
LSTM-MIMO	200	16	0.001	32	0.009605	LSTM (256)	LSTM (128)	Dense
LSTM-SISO	200	32	0.001	64	0.041	LSTM (256)	LSTM (128)	Dense
Bidirectional LSTM	200	16	0.001	32	0.000816	LSTM (128)	LSTM (128)	Dense
Transformer	200	16	0.0001	32	0.000478	Attention LSTM (128)	Attention LSTM (128)	Dense

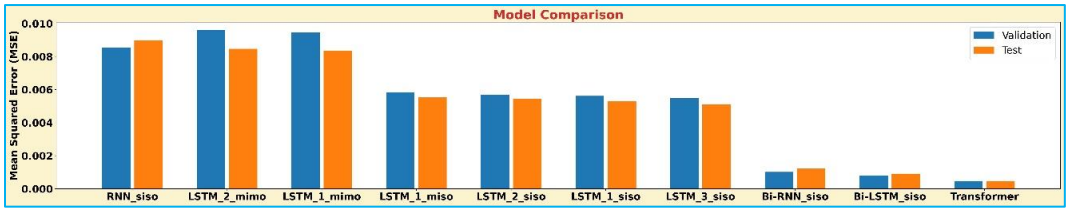

Fig. 6. Clearly demonstrates that the Transformer model outperforms the other models investigated in this study, including the Bi-directional LSTM and Bi-directional RNN, as evidenced by the RMSE values presented in Table 5(b)

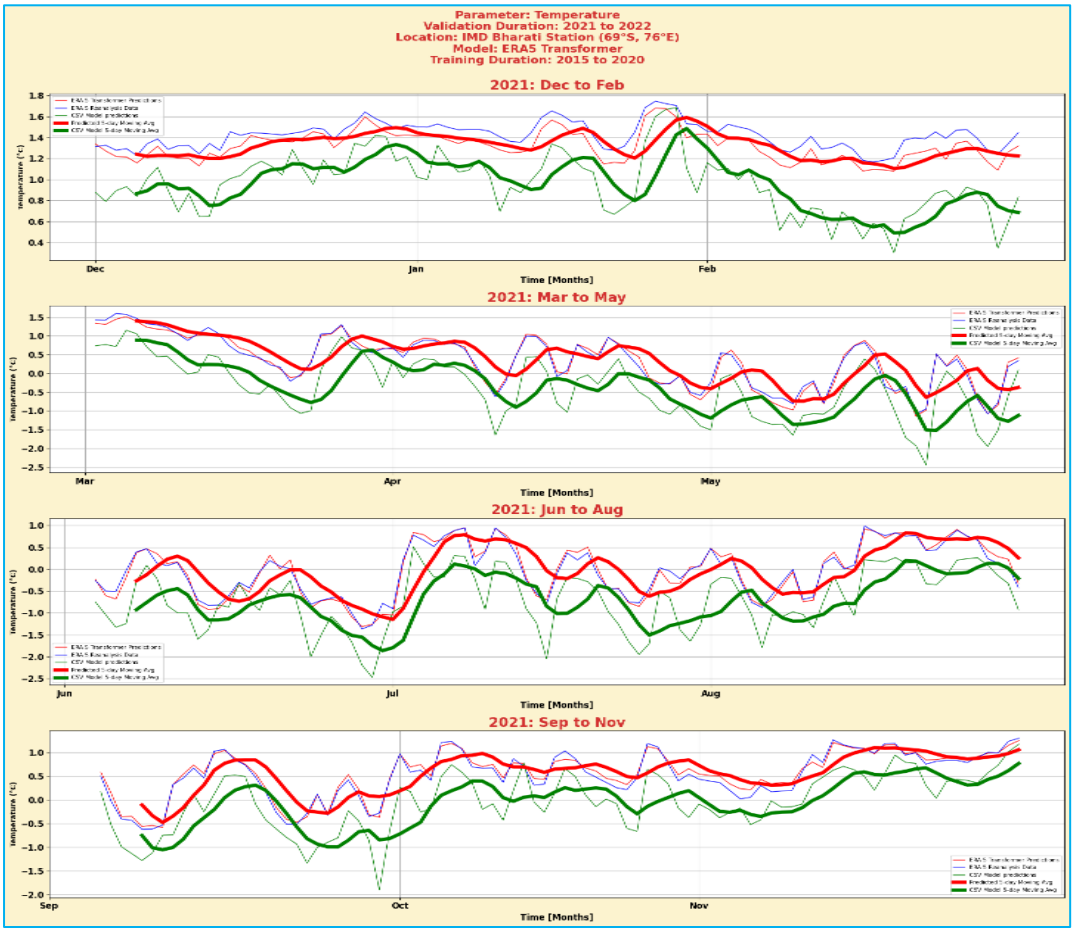

Fig. 7. Shows similar correlation, confirming the Transformer's suitability for nowcasting and predicting conditions over the next 24 to 48 hour

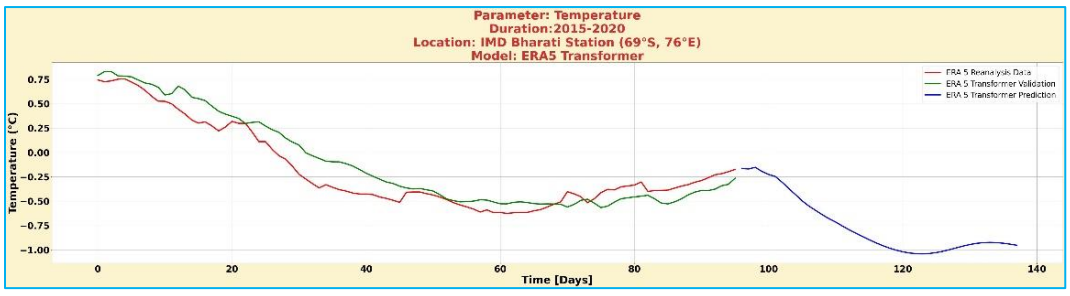

Fig. 8. Displays the ERA-5 data represented by the red line, the ERA-5 validation shown in green, and the 24-hour predictions using the Transformer model depicted by the blue line. This analysis corresponds to the same duration used for the single-point observed dataset at Bharati Station (69° S, 76° E)

showcases the validation of the temperature parameter using the Transformer model, with its corresponding 24-hour prediction shown in Fig. 5(b). The overall comparison of the model configuration parameters, along with the calculated RMSE values, is presented in Table 5(b).

Transformers process sequences through attention mechanisms. Thus, errors can arise if : (*i*) Certain time steps are misinterpreted due to poor attention weights. Or (*ii*) Long-term dependencies are not adequately captured despite the self-attention mechanism.

Fig. 6 presents a comparison analysis that highlights the performance of transformer models above other models studied in the study. This is proven by the RMSE values shown in Table 5(b) which indicates that the transformer model achieves a more accurate and reliable prediction compared to its equivalent.

3.5. *Validation of Transformer model with ERA 5 datasets*

We validated the optimized Transformer model using ERA-5 datasets (ERA5 is the fifth generation ECMWF - European Centre for Medium-Range Weather Forecasts reanalysis, offering global climate and weather data since 1940.Itreplaces the ERA-Interim reanalysis) for our Bharati Station in Antarctica (69° S, 76° E). Its parallel processing capability enhances scalability and speed, making it ideal for complex large-scale datasets. With a mean absolute error of 0.07, it significantly outperforms single-point datasets. The model also excels in spatiotemporal data analysis by effectively managing long-range dependencies. The model excels in spatiotemporal data analysis by effectively managing long-range dependencies. We trained multiple models using single-location data and two with ERA5 data. Fig. 7 below shows similar correlation, confirming the Transformer's suitability for nowcasting and predicting conditions over the next 24 to 48 hours.

In Fig. 7, the thick red line represents the temperature predictions (5-day moving average) using the Transformer model, while the thick green line shows the temperature predictions from single-point datasets (5-day moving average). The thin red line illustrates the temperature predictions using the Transformer model for ERA-5 datasets specifically at the location of Bharati Station (69° S, 76° E). The thin blue line represents the validation of temperature using ERA-5 datasets for the same location. The thin green line indicates the temperature predictions derived from single-point datasets.

Additionally, the seasonal predictions are displayed in Fig. 7 for the following periods: December, January, February (DJB); March, April, May (MAM); June, July, August (JJA); and September, October, November (SON).

We trained multiple models using single-location data and two with ERA5 data. Fig. 8 below confirms the Transformer's suitability for nowcasting and predicting conditions over the next 24 to 48 hours.

Performing several sets of experiments finally concludes by bridging the gap between model outcomes and actionable improvements, ensuring more robust and reliable predictions which is done with the help of error analysis we did in this paper for deep models.

4. Conclusions

This paper introduces a flexible, deep-learning approach for local weather forecasting, enabling quick predictions and cost-effective, reliable short-term forecasts. Unlike previous models that relied on varying degrees of data assimilation, this model is entirely data-driven. Four models were trained to predict atmospheric temperature using a seven-year historical dataset from Bharati Station in Antarctica. The Transformer model is designed for 24 to 48-hour predictions and was trained with parameters like air pressure, relative humidity, and wind speed, achieving an RMSE of 0.000478.

The model's flexibility and speed facilitate short-term local forecasts, particularly in regions where accurate predictions are challenging due to local factors. The results indicate that two-day predictions are comparable in accuracy to expensive numerical weather predictions. However, further accuracy improvements may be achieved by optimizing features such as wind speed and humidity, which will be the focus of future research.

Acknowledgements

We thank The Director of the National Centre for Polar and Ocean Research (NCPOR), The Secretary of the Ministry of Earth Sciences (MoES), New Delhi, and the Director General of Meteorology (DGIMD) for the India Meteorological Department (IMD) for their kind support and encouragement to publish the work.

Funding: National Centre for Polar and Ocean Research, Goa, Ministry of Earth Sciences (MoES).

Data Availability: The weather data supporting the findings of this study is available upon request, and the datasets will be shared accordingly.

Conflicts of Interest: The authors declare no conflict of interest.

Authors' Contributions

V S Samy: Performed writing - original draft and conceptualization, developed model using ML and its Neural Networks.
Veena T: Performed review and editing.

Disclaimer: The contents and views presented in this research article/paper are the views of the authors and do not necessarily reflect the views of the organizations they belong to.

References

Bauer, P., Thorpe, A. and Brunet, G., 2015, "The quiet revolution of numerical weather prediction", *Nature*, **525**, 47-55. doi : 10.1038/ nature14956.

Bochenek, B. and Ustrnul, Z., 2022, "Machine learning in weather prediction and climate analyses-applications and perspectives", *Atmosphere*, **13**, 2, 180.

Fathi, M., Haghi Kashani, M., Jameii, S. M. and Mahdipour, E., 2022, "Big data analytics in weather forecasting : A systematic review", *Arch. Comput. Methods Eng.*, **29**, 2, 1247-1275.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F. and Pedreschi, D., 2018, "A survey of methods for explaining black box models", *ACM Comput. Surv.*, **51**, 5, 1-42.

Gulli, A. and Pal, S., 2017, *Deep Learning with Keras.* Packt Publishing Ltd, Birmingham.

Hewage, P., Trovati, M., Pereira, E. and Behera, A., 2021, "Deep learning-based effective fine-grained weather forecasting model", *Pattern Anal. Appl.*, **24**, 1, 343-366.

Hinton, G. E., 2006, "Reducing the dimensionality of data with neural networks", *Science*, **313**, 5786, 504-507. https://doi.org/ 10.1126/science.1127647.

Jozefowicz, R., Zaremba, W. and Sutskever, I., 2015, "An empirical exploration of recurrent network architectures", In : *Int. Conf. Mach. Learn.*, 2342-2350.

Krasnopolsky, V. M. and Fox-Rabinovitz, M. S., 2006, "Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction", *Neural Netw.*, **19**, 2, 122-134.

Kulandaivelu, E. and Dang, S. P., 2003, "A case study of katabatic winds over Schirmacher Oasis, East Antarctica", *MAUSAM*, **54**, 4, 843-850.

Lal, R. P., 2006, "Short period climatology of Maitri, Schirmacher Oasis, East Antarctica", *MAUSAM*, **57**, 4, 684-688.

LeCun, Y., 1988, "A theoretical framework for back-propagation", In: *Proc. 1988 Connectionist Models Summer School*, CMU, Pittsburgh, PA, 21-28. Morgan Kaufmann.

Linnainmaa, S., 1970, "The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors", Master's Thesis, University of Helsinki.

Lütkepohl, H., 2013, *Introduction to multiple time series analysis*. Springer Science & Business Media.

Lynch, P., 2008, "The origins of computer weather prediction and climate modeling", *J. Comput. Phys.*, **227**, 3431-3444. Doi : 10.1016/j.jcp.2007.02.034.

McCulloch, W. S. and Pitts, W., 1943, "A logical calculus of the ideas immanent in nervous activity", *Bull. Math. Biophys.*, **5**, 115-133. doi:10.1007/BF02478259.

Mohammed, A. and Kora, R., 2023, "A comprehensive review on ensemble deep learning : Opportunities and challenges", *J. King Saud Univ.-Comput. Inf. Sci.*, **35**, 2, 757-774.

Mohammed, A. S., Kareem, S. W., Al Azzawi, A. K. and Sivaram, M., 2018, "Weather prediction using SRE-NAR and SRE-ADALINE", *J. Adv. Res. Dyn. Control Syst.*, **12**, 10.

Radhika, Y. and Shashi, M., 2009, "Atmospheric temperature prediction using support vector machines", *Int. J. Comput. Theory Eng.*, **1**, 1, 55.

Radok, U., 1979, "Polar meteorology and climatology 1975-78", *Rev. Geophys.*, **17**, 7, 1772-1782.

Schmidhuber, J., 2015, "Deep learning in neural networks : An overview", *Neural Netw.*, **61**, 85-117. doi :10.1016/j.neunet. 2014.09.003.

Srivastava, A., Jain, V. K. and Dutta, H. N., 2004, "Surface wind characterization at an Antarctic coastal station Maitri", *MAUSAM*, **55**, 1, 95-102.

Yu, P. S., Yang, T. C., Chen, S. Y., Kuo, C. M. and Tseng, H. W., 2017, "Comparison of random forests and support vector machine for real-time radar-derived rainfall forecasting", *J. Hydrol.*, **552**, 92-104.