Comparative Study of Machine Learning Models for Weather Prediction Using Historical Temperature Data

Meghan Mane[1], Umesh Shinde[2], Sheetal Popat[3], Pranjal Sancheti[4]

K. J. Somaiya Institute of Technology, Sion, Mumbai, India

Abstract

Accurate weather prediction is essential for informed decision-making in a variety of sectors, such as agriculture, disaster management, energy management, and others. The precision of weather forecasting is particularly optimized with the advancement of machine learning techniques. This paper presents a comparative analysis of weather prediction models using historical temperature data from Colaba, Mumbai, from 2017 to 2022. We compare the Root Mean Squared Error (RMSE) values of three algorithms: CatBoost, Stacked GRU, and BiLSTM GRU. CatBoost attains an RMSE of 1.41, closely followed by Stacked GRU at 1.47 and BiLSTM GRU at 1.54. Attention Networks are technically more useful compared to LSTM networks. Their combination with various Deep Learning algorithms is giving better results. The results not only highlight the relative performance of the model but also illustrate the vital role machine learning plays in enhancing weather prediction methods.

Keywords: Weather prediction, Attention model, Temporal Machine Learning model, Deep Learning.

I. INTRODUCTION

Weather is an important aspect that needs to be considered for various important tasks. Weather prediction is of prime importance, especially in India, so it needs to be carried out with precision and maximum accuracy.

Our problem statement deals with carrying out weather prediction by plugging the gaps between the two weather stations and increasing the accuracy; thus, by decreasing the gaps between the stations, more data sets could be added and, thus, by getting accurate results.

Our system involves matching the predicted result with the previous data provided, thus ensuring accuracy. We made use of time series (temporal) models in our research work. First, the raw data is cleaned, and the important data is segregated and used for further purposes. The cleaned data is used to make the data model, which is used for further purposes**.**

Weather prediction remains a serious concern and has attracted the attention of governments, industries, risk management entities, and the scientific community.. We have worked on a weather prediction model for the Colaba weather station in Mumbai for temperature. And the remaining work for rainfall and other parameters is going on.

II. LITERATURE SURVEY

The application of machine learning algorithms in weather prediction has gained significant attention, driven by the potential to enhance forecast accuracy. Initially, linear regression models were among the pioneering techniques applied to historical temperature data for weather prediction. These models aimed to establish linear relationships between temperature and meteorological variables. However, their simplicity hindered

theirability to capture complex non-linear patterns, especially for long-term forecasts.
Multiple linear regression, while accommodating multiple input variables, still assumed linear relationships and struggled with modeling non-linear interactions among meteorological factors, limiting its accuracy in capturing temperature fluctuations. Ensemble models, though powerful, posed challenges due to computational complexity and the need for extensive feature engineering. Additionally, their interpretability was constrained, making it difficult to interpret model predictions. Support Vector Machines (SVMs) demanded careful selection of hyperparameters and kernel functions, posing challenges for different datasets and weather conditions. They were also less suitable for large-scale data due to computational demands.
Later, algorithms like Convolutional Neural Networks, Artificial Neural Networks, and Recurrent Neural networks were used for weather prediction CNNs excelled in spatial analysis but struggled with temporal dependencies, especially for rapidly changing temperature patterns. ANNs encountered the vanishing gradient problem, hampering their effectiveness in modeling long-term temperature dependencies. Their deep variants required substantial data and computational resources, limiting their applicability. RNNs improved sequential data modeling compared to ANNs but were susceptible to gradient issues, restricting their ability to capture long-term dependencies effectively.
Long Short-Term Memory (LSTM) networks addressed RNN limitations, enabling the modeling of long-term dependencies. They gained prominence in time-series forecasting, including weather prediction. Bi-LSTM networks further extended LSTM's capabilities by considering information from both past and future time steps. This bidirectional approach improved temperature predictions by capturing context effectively. Attention mechanisms enhanced deep learning models by enabling them to focus on relevant input data. In weather prediction, attention networks improved the model's ability to weigh the importance of meteorological variables.
Thus, from basic linear regression to advanced deep learning models like LSTM, Bi-LSTM, and attention networks, each algorithm has contributed to enhancing forecast accuracy. Our forthcoming research aims to conduct a comparative analysis of these algorithms, considering factors such as model performance and computational efficiency, to identifythe most suitable approach for accurate weather prediction.

Sr. No.	Paper Name	Summary	Methodology	Limitation
1	Deep Learning-Based Effective Fine-Grained Weather Forecasting Model	Presents a deep learning-based model for fine-grained weather forecasting. Requires substantial data and computation resources.	Deep Learning	Limited to fine-grained weather forecasting. May require substantial data and computational resources.
2	SA-JSTN: Self-Attention Joint Spatiotemporal Network for Temperature Forecasting	Introduces SA-JSTN, a self-attention-based spatiotemporal network for temperature forecasting. Complex architecture	Self-Attention, Spatiotempora lAnalysis	Complex architecture may require significant training data and computation.

		maydemand significant data and computation.		
3	Prediction of Sea Surface Temperature in the South China Sea byArtificial Neural Networks	Utilizes artificial neural networks to predict sea surface temperatures. Relies on accurate input data but may not capturecomplex oceanographic phenomena.	Artificial Neural Networks	Relies on accurate input data, may not account forcomplex oceanographic phenomena.
4	Experimental Studyon Temperature Data of Mass Concrete Construction in HotWeather Conditions	Conducts an experimentalstudy on temperature data during mass concrete construction in hot weather. Provides insightsbut may have limited generalizability.	Experimen talData Analysis	Limited to a specific construction scenario andmay not generalize well.
5	Estimation of Dense Time Seriesof Urban Air Temperatures from Multitemporal Geostationary Satellite Data	Focuses on estimating urban air temperatures from satellite data. Reliant on data availability and may not capture micro-scale temperaturevariations.	Satellite Data Analysis, Time Series Estimation	Reliant on satellite data availability and may notcapture micro-scale temperature variations.
6	Temperature Prediction using Machine Learning Approaches	Discusses temperature prediction using various machine learning approaches. Specific methodologies and algorithms may vary acrossstudies.	Machine Learning Approach es	Methodologies and algorithms vary acrossstudies.

III. DATASET PROCESSING

The analysis of weather data from the Indian Meteorological Department spanning from 2018 to 2022 has revealed intriguing insights regarding the stationarity and variability of temperature values. This data, which is essential for various applications, experiences significant upheavals due to the dynamic nature of Indian weather.

The Augmented Dickey - Fuller test (ADF) was conducted for the determination of whether the data is time series. The differenced weather data showed a positive result to the null hypothesis which indicated that the data is stationary and does not change over time. Whereas when checked for the variance graph (where data points of past 20 days were considered from the current data for each point),the test shows a non - stationary trend which suggeststhat it is difficult to model an algorithm for this data.

The original temperature data has a non-linear trend, which is typical of weather datasets. This non-linearity results from the intricate interplay of meteorological phenomena, seasonal fluctuations, and external variables influencing temperature trends. To prepare the data for rigorous analysis and forecasting, a critical data pretreatment procedure called differencing was performed. Differencing, or determining the difference between consecutive data values, aids in the removal of patterns and seasonality.

While the ADF test confirmed the data's stationarity, a deeper examination of the variance graph revealed an unexpected pattern. In this investigation, the variance was calculated by taking into account data points from the previous 20 days for each present data point. The variance graph revealed a non-stationary trend, showing that temperature variability fluctuates over time. The non-stationary trend in the variance graph is an important finding. It implies that, while the data's central tendency stays constant (as evidenced by the stationary ADF test findings), the temperature range and variability change. This variance could be due to changing weather patterns, seasonal shifts, or external forces.

Finally, the examination of Indian Meteorological Department temperature data emphasizes the dynamic and non-linear nature of Indian weather. While differencing effectively established stationarity in the data's central tendency, the fluctuating variance needs more advanced modeling approaches to assure accurate projections and insights into the region's unique patterns of temperature changes. This study opens up new possibilities for further research into Indian weather patterns and the creation of robust forecasting algorithms.

```
ADF Statistic: -1.7407019209220944
p-value: 0.4102729572845543
Critical Values:
   1%: -3.431661944885779
   5%: -2.862119737440825
   10%: -2.5670785949998973
Fail to reject the null hypothesis - Time series is non-stationary
```

Figure 1: Variance graph shows a non-stationary trend

When computing variance with a 15-day window around each data point, a U-shaped graph is observed. This U-shaped pattern in the variance graph indicates that temperature variability does not remain constant throughout time. Seasonal variations, climate events, and harsh weather conditions are all elements that contribute to this phenomenon. Seasonsof Indian weather include monsoons, winter, and summer, each with its own set of temperature and weather patterns. These seasonal variations cause temperature ranges to vary, influencing the U-shaped variance curve. Furthermore, extreme occurrences such as heatwaves and cold spells can cause sudden temperature changes, adding to the observed oscillations in variance.

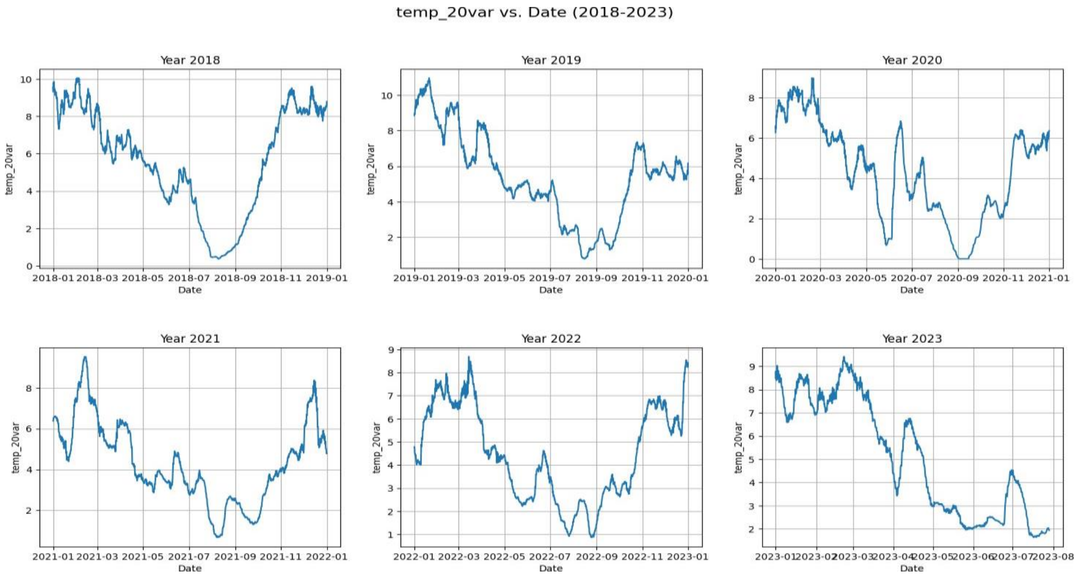

Figure 2: Yearly Graph of Variance from past 15 days vs. date

Temperature data analysis using several time intervals, notably the last 5, 10, and 20 days, has found surprising trends. The variance, which measures the dispersion or fluctuation of temperature data, can reveal important information about the constancy of meteorological conditions throughout time. When the variance is calculated using these multiple time frames, specific characteristics emerge. When the investigation is expanded to include numerous years, common trends emerge. The U-shaped form that develops in the variance graphs issuggestive of seasonal temperature changes. These seasonal variations are a common aspect of weather data, especially in areas with significant seasonal shifts. These differences in India might be ascribed toevents such as monsoons, which cause major changesin temperature and climatic conditions.

Temperature variance analysis employing multiple time intervals provides a thorough perspective of temperature patterns in the Indian weather dataset. The impact of seasonal fluctuationsis shown by the U-shaped shape, while deviations from this shape highlight the influence of external variables and the dynamic nature of Indian weather. This data is critical for constructing reliable and flexible weather prediction models as well as comprehending the region's complex link between climate and temperature.

Notably, while the U-like shape remains stable throughout time, the amount of divergence from it fluctuates. Changes in climate patterns, regional climate events, and even localized weather phenomena can all cause these variances. Understanding these variances can provide useful information for forecasting and responding to distinctweather patterns in different years.

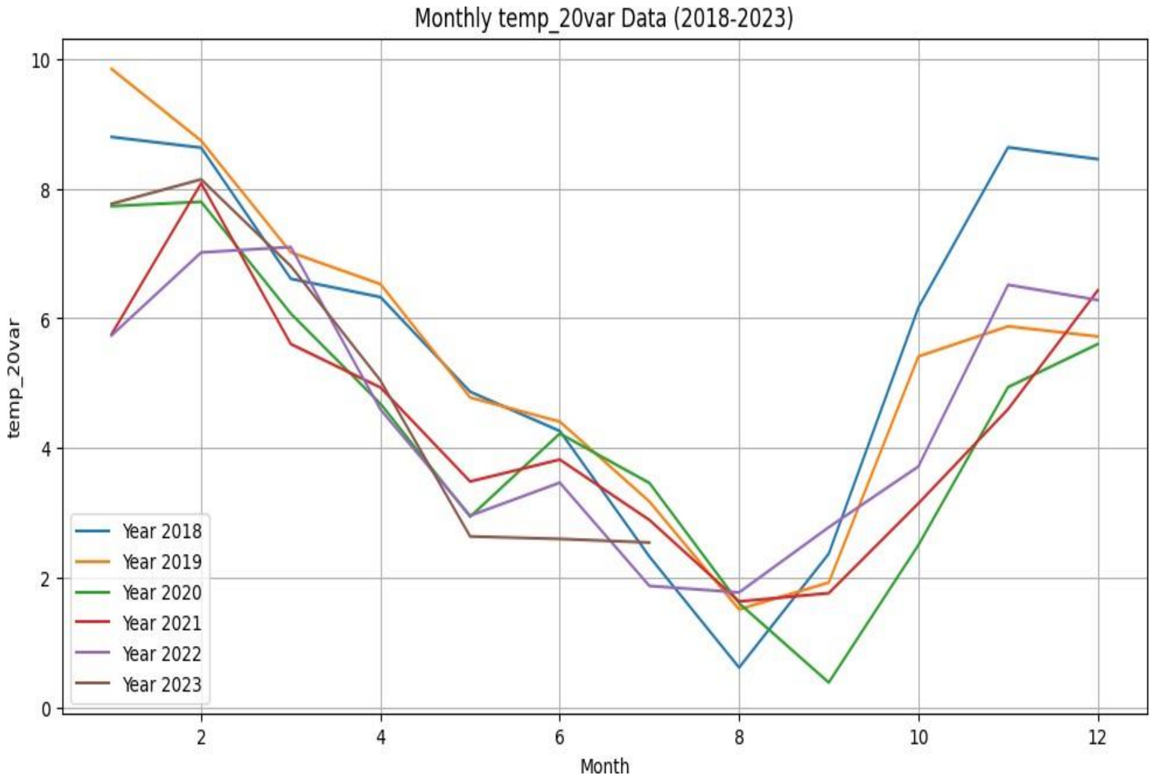

Figure 3: Yearly Graph of Variance from past 20 days vs. month

Following an examination of the variance graphs, it was established that using data from the last 20 days yields the most robust results. Missing values were pre-processed by linear interpolation using a forward technique to maintain data integrity. This thoroughly processed dataset is used to develop temporal deep learning models, which improve accuracy and dependability in weather forecasting and analysis.

IV. TEMPORAL DEEP LEARNING MODELS

Temporal deep learning models are intended to manage and analyze time-series data effectively. These models excel in understanding and capturing evolving patterns, trends, and dependencies, making them important for applications such as weather forecasting, stock market predictions, and natural language processing.

The models used in this analysis were chosen for their capacity to understand the temporal components of the data, as weather data frequently reveals patterns and trends that are influenced by the passage of time. These models are capable of adapting to changing weather conditions, accounting for seasonal fluctuations, and capturing extensive data linkages. Researchers can extract useful insights from data by utilizing deep learning techniques and these specialized models. This includes forecasting, detecting long-term climate patterns, and dealing with the intricacies of Indian weather, which is distinguished by dynamic and non-standard patterns. The application of temporal deep learning models to weather analysis is a cutting-edge strategy that has the potential to improve forecasting accuracy and better comprehend the complexities of climate behavior.

The models used in this analysis are:

CATBOOST:

Catboost stands for the ‘Categorical Boosting’ algorithm. It is well-known for its robustness and advanced performance. It is an advancement of the gradient-boosting algorithm. This algorithm handles the problem of overfitting as it uses regularizers and optimizers.

It is an open-source machine learning method that improves decision trees by using gradient boosting. CatBoost is particularly well-suited for structured data problems due to its efficient management of categorical characteristics.

Gradient Boosting: To make predictions, CatBoost employs the gradient boosting technique, which constructs an ensemble of decision trees. This ensemble method frequently yields highly accurate models.

Support for Categorical Features One of CatBoost's assets is its seamless handling of categorical data. It employs techniques like ordered boosting, which sorts categorical feature values, making efficient divisions in the trees. This avoids the need for significant preprocessing of categorical variables.

CatBoost is versatile and can be used for both regression (predicting numerical values) and classification (classifying data) tasks.

It supports GPUs, which makes training faster and suitable for enormous datasets.

CatBoost offers a variety of hyperparameters for fine-tuning the efficacy of models.

It provides feature importance scores and instrumentsfor model interpretation, enabling users to comprehend the factors driving predictions.

Community and Resources: CatBoost is accessible to data scientists and machine learning practitioners due to its active community and online tutorials and examples.

CatBoost's gradient-boosting approach combines weak learners (decision trees) to produce a robust predictive model, despite the fact that its mathematical details and formulas are complex.

Imagine a decision tree, like a flowchart where each step helps make a choice. CatBoost uses a bunch of these trees and trains them to work together. It's like having a team of experts making decisions.

CatBoost is excellent at dealing with categories, like colors or types of cars. It knows how to handle them, so you don't have to do tricky math to turn colors intonumbers.

In a nutshell, CatBoost is a clever tool that makes machines smarter at making choices, especially when dealing with categories. It uses decision trees, learns from mistakes, and does all the tricky math for you. So, it's like having a team of experts and a math genius on your side when you're working with data.

STACKED GRU:

Stacked GRU (Gated Recurrent Unit) is a neural network architecture in which multiple GRU layers are stacked on top of one another. GRU is a recurrent neural network (RNN) known for its capacity to capture sequential dependencies in data. When these GRU layers are layered, a neural network with multiple levels of GRU units is produced.

Stacking GRU layers is intended to generate a more robust and hierarchical model for tasks requiring the comprehension of intricate sequences. Each layer in the stack processes the input data sequentially, and the output from one layer functions as the input to the next layer. This enables the network to acquire increasingly abstract and sophisticated data representations as it advances through the layers.

Mathematically, each GRU layer's computations entail gate operations, similar to those of a single GRU unit. At each time step, these gates govern the flow of information and determine what to remember and what to discard.

Various sequential data tasks, such as natural language processing, speech recognition, and time-series forecasting, frequently employ stacked GRU networks. They provide increased model capacity and the ability to incorporate long-range data dependencies, making them suitable for difficult sequence prediction problems.

Hidden State (Ht) = Reset Gate (Rt) * Previous Hidden State (Ht-1) + New Information Gate (Zt) * New Information (Xt)

- Ht is what STACKED GRU remembers.
- Rt decides what to keep from the past.
- Zt helps decide what's important from the newinformation.
- Xt is the new information it's looking at.

STACKED GRU uses these calculations to remember and learn patterns in data over time. It stacks these calculations to become even smarter!
In a nutshell, STACKED GRU is like a data detective that learns from the past, predicts the future, and it's used for all sorts of cool things like weather forecasting and speech recognition. Its secret sauce is those calculations that help it remember and learn.

BI-LSTM GRU:
For sequential data processing, BI-LSTM (Bidirectional Long Short-Term Memory) and GRU (Gated Recurrent Unit) are both types of neural networks.
BI-LSTM (Bidirectional Long Short-Term Memory): This network works in both directions. It processes both historical and prospective data, capturing dependencies in both directions. It excels at sentiment analysis and machine translation duties.
GRU (Gated Recurrent Unit): GRU is a tad simpler than LSTM, but it is still highly effective. It is easier to train and has fewer gates. It is frequently used when high performance is required but a more efficient network is desired.
Both BI-LSTM and GRU entail complex mathematical calculations involving weights and gates. These calculations assist the networks in remembering pertinent information and forgetting irrelevant data.
BI-LSTM combines two LSTM networks, one processing data in a forward direction and the other in a reverse direction.
To control information flow, the formulas include gate operations such as the Input Gate (It), Forget Gate (Ft), and Output Gate (Ot).
For GRU: It also has information management gates such as the Reset Gate (Rt) and Update Gate (Zt).
GRU has fewer parameters than LSTM, which may result in quicker training.
BI-LSTM and GRU are both utilized in numerous applications, including natural language processing, time-series forecasting, and more. The choice between them depends on the specific mission, available resources, and complexity versus efficiency tradeoffs desired.
Let's dive into the mechanics with a simplified formula:

Hidden State (Ht) = Forget Gate (Ft) * Previous Hidden State (Ht-1) + Input Gate (It) * New Information (Xt)

- Ht is what BI-LSTM GRU remembers.
- Ft helps it decide what to forget from the past.
- It tells it what new information to add.

Here's a simple formula for the bidirectional part: Bidirectional Hidden State (Ht) = Forward Hidden State (Ht_forward) + Backward Hidden State (Ht_backward)

- Ht_forward is information from left to right.
- Ht_backward is information from right to left.

By combining all these calculations, BI-LSTM GRU can understand and predict sequences better.

BI-ATTENTION:

Bidirectional Attention Networks (BI-ATTENTION), also referred to as Bidirectional Attention Flow (BiDAF), are sophisticated models for natural language processing (NLP) and question-answering tasks. They are designed to understand the relationships between various parts of a text and provide answers that are contextually relevant.

BI-ATTENTION is an architecture for neural networks that uses bidirectional attention mechanisms to enhance text comprehension. It was developed to address the shortcomings of traditional models for tasks such as question-answering and reading comprehension. BI-ATTENTION models can answer complex queries by attending to relevant portions of the text both forwards and backwards, thereby capturing context effectively.

It enables models to capture context from both past and future portions of the text, thereby augmenting their comprehension of the relationships between words and phrases.

Question-Answering: BI-ATTENTION is especially effective in question-answering duties, where it assists with recognizing the relevant data in a passage in order to answer questions accurately.

Natural Language Processing: It has applications in context-dependent NLP tasks such as machine translation, sentiment analysis, and text summarization.

Improved Accuracy: BI-ATTENTION models frequently outperform conventional models, particularly for complex tasks requiring a profound understanding of language.

BI-ATTENTION involves several mathematical calculations and components:

Contextual Embeddings: The input text is first embedded into contextual word representations using pre-trained word embeddings like Word2Vec or GloVe.

Attention Mechanisms: To attribute importance scores to words in the passage and question, BI-ATTENTION employs attention mechanisms. It computes two attention matrices: attention from passage to question and attention from question to passage.

Attention Score (Passage to Question): Determine the degree of relevance between each word in the passage and each word in the question by calculating their similarity. This is accomplished using the dot product or other measures of similarity.

Attention Score (Question to Passage): Analogous to the attention score from passage to question, but calculated in the opposite direction.

Contextual Embedding Fusion: Combine the embeddings of the passage and the query with their respective attention weights to generate enhanced contextual embeddings.

Bidirectional Attention Flow: Perform bidirectional attention flow by allowing information from the passage to flow into the question and vice versa.

Layer of Modeling: Model the enhanced contextual embeddings using a bidirectional LSTM or another recurrent neural network (RNN).

Answer Pointer: Use a pointer network or other mechanisms to locate the start and end positions of the answer span in the passage.

ATTENTION - BASED TRANSFORMER:

Various natural language processing (NLP) tasks have been revolutionized by the Attention-Based Transformer, a potent machine learning model. It is a neural network architecture presented in the 2017 paper "Attention Is All You Need" by Vaswani et al. The Transformer model relies on self-attention mechanisms, eschewing recurrence and convolutions that were prevalent in earlier models.

Self-Attention Mechanism: At the core of the Transformer is the self-attention mechanism, which enables the model to determine the relative importance of various input sequence segments when generating an output. This attentiveness mechanism is used to identify dependencies between words in a sequence, which makes it highly effective for tasks such as machine translation, text summarization, and question-answering.

Multiple Attention Heads: Transformers use multiple attention heads to capture various types of inter-word relationships. Each head concentrates on a unique aspect of the input sequence, enabling the model to learn intricate patterns.

Positional Encoding: Because the Transformer lacks positional information (unlike recurrent models), positional encoding is applied to the input embeddings to inform the model of the position of each word in the sequence.

Transformers consist of encoder and decoder stacks for processing input and output sequences, respectively. Each layer of both stacks consists of a multi-head self-attention mechanism followed by feedforward neural networks.

Residual Connections and Layer Normalization: Residual connections (skip connections) are utilized to facilitate the passage of gradients during training, while layer normalization is used to stabilize training. After the attention mechanism, a feedforward neural network is applied to each sequence position to further process the data.

Due to its ability to model long-range dependencies efficiently, the Attention-Based Transformer has obtained state-of-the-art results in numerous NLP tasks. It has also served as the basis for subsequent models such as BERT, GPT, and RoBERTa, which have contributed to the advancement of natural language comprehension and generation.

V. METHODOLOGY

Data preparation is an important step in the study and modeling of weather data, and in this case, data gathered from the Indian Meteorological Department (IMD) underwent Forward Linear Interpolation to resolve missing data points. This method is critical for ensuring that the data is acceptable for accurate analysis and forecasting. Forward Linear Interpolation is the process of predicting missing data points from neighboring data. When used with weather data, it fills in missing temperature values in the dataset. This is especially critical in meteorological datasets, which may have missing values owing to sensor failure, data transmission difficulties, or other factors. A more full and continuous dataset is created by linearly interpolating the missing values.

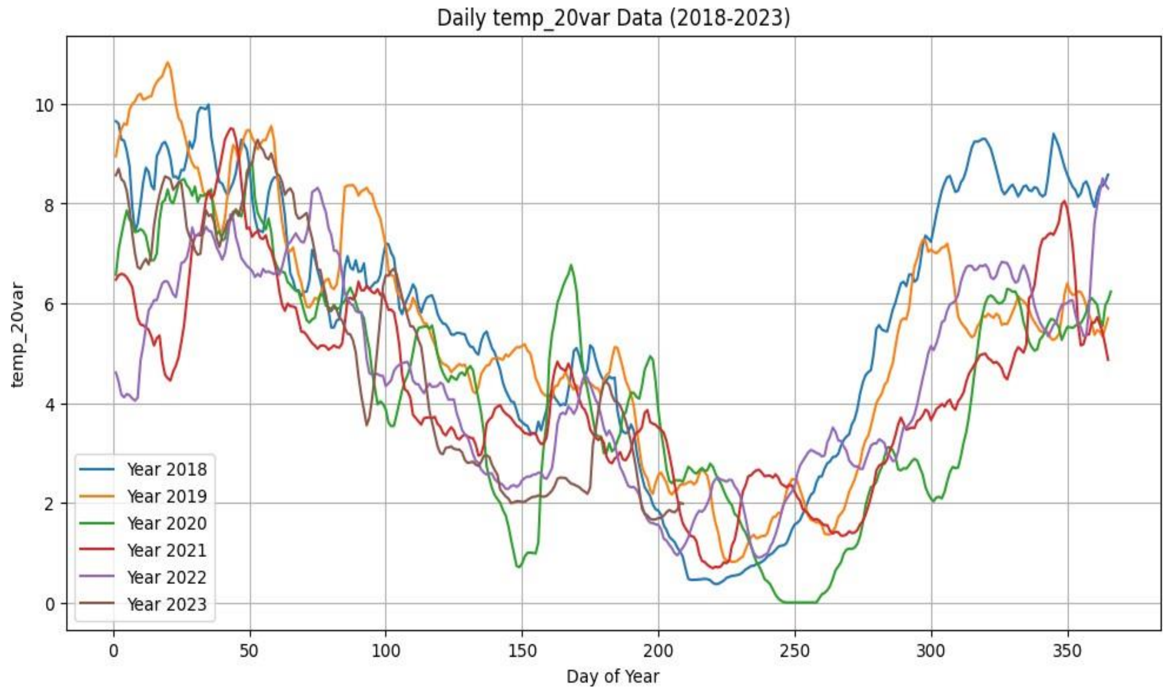

Figure 4: Yearly Graph of Variance from past 20 days vs. days

Furthermore, when considering the last 20 days of data, the observation of a smoother transition in the standard trend is remarkable. A smoother transition indicates that the temperature variation is more consistent during this 20-day period, which is useful for modeling and forecasting. The decision to change the data columns to represent the data points of the previous 20 days while taking the rolling variance of the previous 20 days corresponds with this goal. Since the trend is showing a comparatively smoother transition, the data set is used in referenceto the variance of the past 20 days. Thus the data setis ready for the experimentation.

The Keras library, a sophisticated tool that provides access to a wide range of pre-defined layer structures for building neural networks, was used to develop the deep learning models outlined before. Keras is well-known for its easy-to-use interface and flexibility, making it an excellent choice for building sophisticated neural networks. Because deep learning tasks are computationally intensive, these algorithms were run on GPUs (Graphics computing Units), which provide high-end computing capacity, allowingfor faster training and more efficient model creation.

The deep learning models were analyzed by looking at the results of several experimentation codes. Accuracy and root mean squared error (RMSE) were two key performance indicators evaluated in this investigation. Accuracy quantifies how well the model's predictions match real data, offering insight into the model's overall soundness. RMSE, on the other hand, quantifies the average difference between anticipated and actual values to measure the model's prediction inaccuracy. These measures serve as critical tools for assessing the effectiveness and dependability of deep learning models, allowing us to assess their predicted accuracyand performance quality.

Layer (type)	Output Shape	Param #	Connected to
input_27 (InputLayer)	[(None, 480, 1)]	0	[]
dropout_26 (Dropout)	(None, 480, 1)	0	['input_27[0][0]']
bidirectional_6 (Bidirectional)	(None, 480, 400)	243600	['dropout_26[0][0]']
multi_head_attention_25 (MultiHeadAttention)	(None, 480, 1)	346753	['bidirectional_6[0][0]', 'bidirectional_6[0][0]', 'bidirectional_6[0][0]']
flatten_21 (Flatten)	(None, 480)	0	['multi_head_attention_25[0][0]']
dense_24 (Dense)	(None, 1)	481	['flatten_21[0][0]']

Total params: 590834 (2.25 MB)
Trainable params: 590834 (2.25 MB)
Non-trainable params: 0 (0.00 Byte)

Figure 5: Bi - Attention

The data in this analysis has a particular structure, with the input shape consisting of 480 columns. This is because the data points are captured at 15-minute intervals, and the dataset contains information from the previous 20 days, which is a significant amount of time-series data. This structure ensures that the input dataset has a large number of temperature observations throughout time.

The output dataset, on the other hand, is relatively simple, with simply a single number reflecting the expected weather situation five days in the future. This output is linked to the input data from the appropriate time period, allowing for weather pattern prediction.

Layer (type)	Output Shape	Param #
bidirectional (Bidirectional)	(None, 336, 64)	8704
dropout (Dropout)	(None, 336, 64)	0
gru (GRU)	(None, 16)	3936
dense (Dense)	(None, 512)	8704
dense_1 (Dense)	(None, 1)	513

Total params: 21857 (85.38 KB)
Trainable params: 21857 (85.38 KB)
Non-trainable params: 0 (0.00 Byte)

Figure 6: Bi-LSTM GRU

A variety of deep learning architectures are used to generate models for exploration. These models take advantage of the data's temporal structure, taking into account the sequential elements and patterns that evolve over time. They are intended to capture the data's linkages and dependencies that are critical for successful weather forecasting.

Layer (type)	Output Shape	Param #
gru (GRU)	(None, 336, 128)	50304
dropout (Dropout)	(None, 336, 128)	0
gru_1 (GRU)	(None, 128)	99072
dense (Dense)	(None, 512)	66048
dense_1 (Dense)	(None, 1)	513

Total params: 215937 (843.50 KB)
Trainable params: 215937 (843.50 KB)
Non-trainable params: 0 (0.00 Byte)

Figure 7: Stacked GRU

These deep learning models' complexity and flexibility allow them to adapt to the dynamic nature of meteorological data, which frequently exhibits non-linear and nuanced patterns impacted by elements such as seasons, climate events, and geographical variables. The algorithms are prepared to detect these temporal variations and make predictions based on the vast historical context of the input data.

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 480, 1)]	0	[]
dropout (Dropout)	(None, 480, 1)	0	['input_1[0][0]']
multi_head_attention (MultiHeadAttention)	(None, 480, 1)	2017	['dropout[0][0]', 'dropout[0][0]', 'dropout[0][0]']
flatten (Flatten)	(None, 480)	0	['multi_head_attention[0][0]']
dense (Dense)	(None, 1)	481	['flatten[0][0]']

Figure 8: Attention

Researchers hope to improve the precision of weather forecasts by using deep learning techniques, particularly in the context of the Indian weather system, which is notorious for its fluctuation and unpredictability. These models provide a forward-thinking approach to analyzing and forecasting meteorological conditions, providing valuable insights for a variety of applications such as agriculture, disaster planning, and energy management.

VI. EXPERIMENTAL RESULTS

CatBoost has the lowest RMSE of 1.49 among the models, suggesting its outstanding accuracy in temperature forecasts. With an RMSE of 1.51, XGBoost comes in second, demonstrating the efficiency of gradient boosting algorithms in capturing the complicated patterns of time-series meteorological data. The RMSE of Stacked Gated Recurrent Units (GRU) is 1.55, demonstrating competitive performance among deep learning models. Meanwhile, with an RMSE of 1.60, BiLSTM GRU combines two powerful deep learning components. These findings highlight the utility of gradient boosting techniques in weather prediction.

Deep learning models, such as the Attention Transformer and Bi-Attention, have slightly higher RMSE values, 1.71 and 1.66, indicating that they are less accurate in temperature prediction. With an RMSE of 1.66, AdaBoost, a conventional ensemble learning approach, performs comparably to Bi-Attention. Specific criteria and the trade-off between accuracy and computing complexity should dictate model selection.

Table 1 Analysis of different temporal ML and DL models

ML/ DL MODEL	RMSE
CatBoost	1.49
XGBoost	1.51
Stacked GRU	1.55
BiLSTM GRU	1.60
Attention transformer	1.71
Bi-Attention	1.66
AdaBoost	1.66

Table 2 examines various Machine Learning(ML) and Deep Learning (DL) models that use rolling variance and standard deviation as training set characteristics. The Root Mean Squared Error (RMSE) measure is used to assess the predictive accuracy of various models.

Notably, the addition of rolling variance and standard deviation as features improved model performance. CatBoost, a gradient boosting technique, showed an amazing RMSE of 1.44, suggesting improved temperature forecast accuracy. XGBoost also performed well, with an RMSE of 1.42, demonstrating the efficiency of gradient boosting algorithms when given this additional data.

Deep learning models, including Stacked Gated Recurrent Units (GRU) and the Attention Transformer, also demonstrated enhanced accuracy.

Stacked GRU achieved an RMSE of 1.40, demonstrating its ability to use rolling variance and standard deviation data to make accurate predictions. Similarly, the Attention Transformer profited from similar qualities, albeit having a somewhat higher RMSE of 1.46. The RMSE values for BiLSTM GRU, Bi-Attention, and AdaBoost were 1.60, which, while not as low as some of the boosting algorithms, nonetheless suggest good predictive performance. In summary, including rolling variance and standard deviation data in training sets improved model accuracy, particularly for gradient boosting methods and specific deep learning architectures.

Table 2 Analysis of models when rolling variance and standard deviation is used in training sets

ML/ DL MODEL	RMSE
CatBoost	1.44
XGBoost	1.42

Stacked GRU	1.40
BiLSTM GRU	1.60
Attention transformer	1.46
Bi-Attention	1.59
AdaBoost	1.60

VII. CONCLUSION

Both traditional machine learning models and deep learning models have proven to be effective in a variety of applications. Deep learning models, on the other hand, have a distinct edge in terms of adaptability and parameter adjustment, allowing them to better cater to individual limitations and requirements.

Deep learning models are very good at learning intricate patterns and representations in data. They are capable of adapting to complex relationships and non-linearities, making them ideal for jobs such as weather forecasting. These models can also be fine-tuned and tailored to fit specific requirements. This versatility enables meteorologists and academics to optimize the model's performance, ensuring it matches the intricacies of weather data.

While deep learning models have impressive prediction capabilities, it is important to note that there is still an approximate 2 degree Celsius difference between the actual and anticipated temperatures. This difference is relevant in the context of weather forecasting since modest temperature fluctuations can have considerable effects on daily activities, agriculture, and disaster management. It provides vital insights into temperature trends, which can help with informed decision-making when planning outdoor events, preparing for temperature-sensitive activities, or assessing potential weather-related concerns. Furthermore, the accuracy of weather forecasts is influenced by a variety of factors, such as the quality and quantity of available data, model complexity, and the unpredictability of specific weather occurrences.

VIII. FUTURE SCOPE

Weather forecasting is an important part of our daily life, affecting everything from agriculture to transportation to disaster preparedness. Because of the inherent complexities of weather patterns, relying only on tabular data, such as temperature, humidity, and pressure readings, sometimes falls short of delivering reliable forecasts. Integration of satellite pictures and Generative Adversarial Networks (GANs) has emerged as a viable way to alleviate this constraint and increase the accuracy of weather predictions. This combination has the ability not only to improve forecasting accuracy but also to play a critical role in the early detection of dangerous natural disasters. Weather is governed by atmospheric and climatic variables that are diverse and frequently exhibit nonlinear and dynamic behavior. Understanding these complicated phenomena necessitates the use of numerous data sources, like satellite imagery.

GANs, or Generative Adversarial Networks, have proven to be effective picture generating and translation tools. GANs are made up of two neural networks, a generator and a discriminator, that compete against each other to improve the generator's capacity to generate realistic images. This method can be applied to weather forecasting by utilizing GANs to synthesize future satellite photos. A GAN may learn to generate synthetic satellite photos that capture the expected circumstances at a specific time in the future by training it on historical weather data and associated satellite images.

The use of GAN-generated satellite imagery into weather prediction models has the potential to greatly increase forecast accuracy. These simulated images can assist modelers in accounting for a variety of meteorological phenomena such as the formation and movement of weather fronts, the development of thunderstorms, and the advancement of tropical cyclones. As a result, more precision in simulating future weather conditions can lead to more precise predictions, benefiting a wide range of sectors and applications. Furthermore, the utility of GANs in weather prediction goes beyond simply improving accuracy. Early identification of natural catastrophes is critical in areas prone to disasters such as hurricanes, tornadoes, and floods. GANs can be used to translate satellite photos and compare current conditions to forecasts from weather models. Any differences between actual and simulated visuals can serve as early warning indications of potentially hazardous events. This proactive approach can save lives by initiating timely evacuation and readiness measures far in advance.

As a result, weather prediction is a difficult and critical endeavor, and traditional methods frequently fall short of the precision required for particular purposes. The use of Generative Adversarial Networks to integrate satellite pictures with weather prediction models has the potential to revolutionize this industry. We can increase the accuracy of weather forecasts, boost disaster preparedness, and ultimately save lives by harnessing the power of GANs to generate realistic future satellite imagery and detect early indicators of natural calamities. This novel technique represents a substantial advancement in our ability to comprehend and predict the complex and ever-changing world of weather.

IX. REFERENCES

[1] L. Yang, L. Shang and L. Yang, "Improved Prediction of Markov Chain Algorithm for Indoor Temperature," 2016 International Symposium on Computer, Consumer and Control (IS3C), Xi'an, China, 2016, pp. 809-812, doi: 10.1109/IS3C.2016.206.

[2] Y. Jianwen, Y. Qiang, L. Honglin, W. Haixu and Z. Zhuoyu, "Anomalies analysis and prediction method research of converter transformer winding temperature in HVDC project based on BP neural network," 2022 IEEE 4th International Conference on Power, Intelligent Computing and Systems (ICPICS), Shenyang, China, 2022, pp. 932-935, doi: 10.1109/ICPICS55264.2022.9873587.

[3] Kreuzer, D., Munz, M., & Schlüter, S. (2020). Short-term temperature forecasts using a convolutional neural network — An application to different weather stations in Germany. *Machine Learning With Applications*, *2*, 100007. https://doi.org/10.1016/j.mlwa.2020.100007

[4] Alawadi, Sadi & Mera, David & Fernández-Delgado, Manuel & Alkhabbas, Fahed & Olsson, Carl & Davidsson, Paul. (2022). A comparison of machine learning algorithms for forecasting indoor temperature in smart buildings. Energy Systems. 13. 10.1007/s12667-020-00376-x.

[5] Luong, Minh-Thang & Pham, Hieu & Manning, Christopher. (2015). Effective Approaches to Attention-based Neural Machine Translation. 10.18653/v1/D15-1166.

[6] B. Bechtel, S. Wiesner and K. Zakšek, "Estimation of Dense Time Series of Urban Air Temperatures from Multitemporal Geostationary Satellite Data," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 7, no. 10, pp. 4129-4137, Oct. 2014, doi: 10.1109/JSTARS.2014.2322449.

[7] Holmstrom, Mark A. and Dylan Liu. “Machine Learning Applied to Weather Forecasting.” (2016).

[8] L. Wei, L. Guan and L. Qu, "Prediction of Sea Surface Temperature in the South China Sea by Artificial Neural Networks," in IEEE Geoscience and Remote Sensing Letters, vol. 17, no. 4, pp. 558-562, April 2020, doi: 10.1109/LGRS.2019.2926992.

[9] Hewage, Pradeep & Trovati, Marcello & Pereira, Ella & Behera, Ardhendu. (2021). Deep learning-based effective fine-grained weather forecasting model. Pattern Analysis and Applications. 24. 10.1007/s10044-020-00898-1.

[10] L. Shi, N. Liang, X. Xu, T. Li and Z. Zhang, "SA-JSTN: Self-Attention Joint Spatiotemporal Network for Temperature Forecasting," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 9475-9485, 2021, doi: 10.1109/JSTARS.2021.3112131.

[11] Awad, Mariette & Khanna, Rahul. (2015). Support Vector Machines for Classification. 10.1007/978-1-4302-5990-9_3.

[12] C. Jiang, H. Zhang, Y. Ren, Z. Han, K. -C. Chen and L. Hanzo, "Machine Learning Paradigms for Next-

Generation Wireless Networks," in IEEE Wireless Communications, vol. 24, no. 2, pp. 98-105, April 2017, doi: 10.1109/MWC.2016.1500356WC.
[13] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. *ArXiv*. /abs/1706.03762