A PROJECT REPORT

ON

DECODING 25-YEAR TEMPERATURE TRENDS: DEEP LEARNING INSIGHTS FROM MAHATMA GANDHI CENTRAL UNIVERSITY TIME SERIES DATA

Submitted in Mini Project requirement for the

“BACHELOR OF TECHNOLOGY”

IN

COMPUTER SCIENCE & INFORMATION TECHNOLOGY

Of

MAHATMA GANDHI CENTRAL UNIVERSITY, BIHAR

Submitted by

ARTI CHOUDHARY - MGCU2020CSIT3006

MUKUL ANAND - MGCU2020CSIT3014

RAUNAK KUMAR -MGCU2020CSIT3020

[B. TECH 2020-24]

6TH SEMESTER

Under the Guidance of

Mr. Harshit Srivastava sir

[CS&IT]

DECLARATION

This is to officially certify that the scholarly dissertation titled "Decoding 25-Year Temperature Trends: Deep Learning Insights from Mahatma Gandhi Central University Time Series Data" is hereby submitted to the esteemed Department of Computer Science and Information Technology at the illustrious Mahatma Gandhi Central University, situated in the charming city of Motihari, Bihar 845401, India. This submission marks a significant stride towards the completion of the requisites for the conferment of the distinguished Bachelor of Technology degree in the dynamic field of Computer Science & Engineering.

With earnest integrity, I attest that this dissertation stands as an authentic manifestation of original work, diligently undertaken by me, under the astute guidance of the esteemed mentor "Mr. Harshit Kumar Sir". The intellectual content enshrined within this scholarly work has neither been submitted in part nor in its entirety to any other university, institution, or authority for the fulfillment of any other academic degree or accolade.

Endorsed by our commitment to scholarly excellence and dedication, I, along with my co-authors, proudly affix our identities and designations:

Arti Choudhary (MGCU2020CSIT3006)
Mukul Anand (MGCU2020CSIT3014)
Raunak Kumar (MGCU2020CSIT3020)

Manifested under the banner of the illustrious Department of Computer Science and Information Technology, within the realm of Mahatma Gandhi Central University, Motihari, Bihar 845401, India.

APPROVAL CERTIFICATE

This is to certify that the project report entitled "DECODING 25-YEAR TEMPERATURE TRENDS: DEEP LEARNING INSIGHTS FROM MAHATMA GANDHI CENTRAL UNIVERSITY TIME SERIES DATA" is a bonafide work carried out by ARTI CHOUDHARY (MGCU2020CSIT3006), MUKUL ANAND (MGCU2020CSIT3014) and RAUNAK KUMAR (MGCU2020CSIT3020) in a fulfilment the requirement of mini project for the B.TECH in CS&IT of 6th sem of MAHATMA GANDHI CENTRAL UNIVERSITY, BIHAR during the year 2020-24.

It is certified that all the corrections /suggestions indicated in internal assessment have been incorporated in the project report. The project report has been approved as it satisfies the academic requirements with respect to the project work prescribed for the bachelor in technology in computer science.

Date:- 27/08/2023

Arti Choudhary, Mukul Anand and Raunak Raj
B.Tech. 6th Semester
CS&IT
Mahatma Gandhi Central University

ABSTRACT

Climate change has become a critical global concern, and understanding long-term temperature trends is essential for informed decision-making and policy formulation. This mini project focuses on utilizing deep learning techniques to decode and analyze 25-year temperature trends using time series data from Mahatma Gandhi Central University. The project employs a combination of recurrent neural networks (RNNs) and convolutional neural networks (CNNs) to uncover intricate patterns and insights within the temperature data.

The project begins by preprocessing the temperature data, including cleaning, normalization, and feature extraction. Subsequently, a hybrid deep learning model is constructed, incorporating both RNNs and CNNs. The RNN component aims to capture temporal dependencies and sequential patterns present in the time series data, while the CNN component focuses on detecting spatial patterns that might contribute to temperature variations.

The trained model is subjected to a rigorous evaluation process, employing performance metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). The insights derived from the model's predictions provide a comprehensive understanding of the temperature trends over the 25-year period. Additionally, feature attribution techniques are applied to identify the most influential factors driving temperature fluctuations.

The results of this mini project contribute to the field of climate science by showcasing the potential of deep learning in analyzing and decoding complex temperature trends. The hybrid approach demonstrates the significance of combining different neural network architectures to gain holistic insights from time series data. The findings emphasize the importance of continued research into advanced data-driven methodologies for addressing the challenges posed by climate change.

ACKNOWLEDGEMENT

In the realm of knowledge, we stand indebted to a symphony of contributions that harmonized to shape this project. Our profound appreciation extends to "Mr. Harshit Kumar Sir," whose wisdom conducted us through the labyrinth of learning.

Gratitude dances towards the Department of Computer Science and Information Technology at Mahatma Gandhi Central University, a fertile ground for intellectual exploration.

A nod of recognition to the camaraderie of Arti Choudhary, Mukul Anand, and Raunak Kumar, fellow sojourners in this academic odyssey.
And to our friends, whose unwavering faith has been the North Star guiding us.
This endeavour is an echo of these influences, a tribute to the collective energy that breathed life into these pages.

Table of Contents

1. Introduction………………………………………………… 7

2. Problem Definition…………………………………………… 7

3. Objective of project…………………………………………… 8

4. Methodology……………………………………………….. 8-19
    4.1. Dataset
    4.2. Preprocessing
    4.3. Dataset Plotting
    4.4. Trend, Seasonality & Residual Plotting
    4.5. Applying Statistical Model
    4.6. Applying Deep Learning Model
    4.7. Comparison of all 10 (5 Statistical + 5 Deep Learning) Models
    4.8. Applying LSTM for Forecasting
    4.9. Hardwares and Softwares requirements
5. Results And Discussions…………………………………... 21
6. Reference/bibliography………………………………………. 22

"Decoding 25-Year Temperature Trends: Deep Learning Insights from Mahatma Gandhi Central University Time Series Data"

1. Introduction

In this study, we embark on a journey through a quarter-century of temperature data from Mahatma Gandhi Central University (MGCU). Employing deep learning techniques, we unravel the intricate tapestry of temperature trends concealed within this dataset.

Our analysis delves into the nuanced fluctuations and long-term patterns that have emerged over the past 25 years. By harnessing the power of machine learning, we extract insightful patterns that shed light on the dynamic nature of temperature changes.

Through meticulous examination and model training, we decipher the underlying insights that these temperature trends hold. Our work not only provides a glimpse into the climatic evolution but also showcases the potential of deep learning in deciphering complex temporal data.

As we decode the past, we also glimpse into the future, envisioning the potential applications of our findings in various fields, from climate science to urban planning. This endeavor underscores the significance of combining advanced computational techniques with environmental data to extract meaningful and actionable knowledge.

2. Problem Definition

In this study, we embark on a journey through a quarter-century of temperature data from Mahatma Gandhi Central University (MGCU). Employing deep learning techniques, we unravel the intricate tapestry of temperature trends concealed within this dataset.

Our analysis delves into the nuanced fluctuations and long-term patterns that have emerged over the past 25 years. By harnessing the power of machine learning, we extract insightful patterns that shed light on the dynamic nature of temperature changes.

Through meticulous examination and model training, we decipher the underlying insights that these temperature trends hold. Our work not only provides a glimpse into the climatic evolution but also showcases the potential of deep learning in deciphering complex temporal data.

As we decode the past, we also glimpse into the future, envisioning the potential applications of our findings in various fields, from climate science to urban planning. This endeavor underscores the significance of combining advanced computational techniques with environmental data to extract meaningful and actionable knowledge.

3. Objective of project

3.1. To create a strong and precise model for predicting temperature values over a defined period.

3.2. To grasp intricate temporal patterns in temperature data, enhancing prediction accuracy beyond traditional methods by utilizing deep learning methods.

4. Methodology:

4.1. **Dataset:** Temperature Data of MGCUB of last 25 years
https://drive.google.com/file/d/1NKoygABRLL_oa-SrBQJmzMcwj4ueXf1v/view?usp=drive_link

	DATE	TEMPERATURE
0	10-08-1998	30.46
1	11-08-1998	29.19
2	12-08-1998	30.83
3	13-08-1998	28.33
4	14-08-1998	27.62
...	...	...
9126	05-08-2023	31.28
9127	06-08-2023	31.93
9128	07-08-2023	28.81
9129	08-08-2023	29.76
9130	09-08-2023	29.72

9131 rows × 2 columns

	TEMPERATURE
count	9131.000000
mean	26.093223
std	7.242553
min	7.190000
25%	20.030000
50%	27.610000
75%	31.165000
max	43.660000

After Indexing

	TEMPERATURE
DATE	
1998-10-08	30.46
1998-11-08	29.19
1998-12-08	30.83
1998-08-13	28.33
1998-08-14	27.62
...	...
2023-05-08	31.28
2023-06-08	31.93
2023-07-08	28.81
2023-08-08	29.76
2023-09-08	29.72

9131 rows × 1 columns

Figure 1:Dataset Description

4.2. Preprocessing:

In essence, our code takes a time series dataset of temperature readings, prepares the data by cleaning and formatting it, visualizes the data and its components, explores potential patterns through autocorrelation, and potentially assesses different machine learning models for forecasting. The visualizations and preprocessing aim to extract meaningful insights from the temperature data.

Here's a simplified breakdown of the major steps:

4.2.1. *Importing Libraries:* The code begins by importing necessary libraries like pandas for data handling, matplotlib for plotting, and others.

4.2.2. *Loading Data:* The temperature data from a CSV file is loaded into a pandas Data Frame.

4.2.3. Data Cleaning and Formatting:

4.2.3.1. The 'DATE' column is converted to a datetime format.

4.2.3.2. The 'DATE' column is set as the index of the Data Frame.

```
#convert the 'Date column to a datatime type
data['DATE'] = pd.to_datetime(data['DATE'])

#Set the 'Date column as the index
data.set_index('DATE', inplace=True)
data
```

DATE	TEMPERATURE
1998-10-08	30.46
1998-11-08	29.19
1998-12-08	30.83
1998-08-13	28.33
1998-08-14	27.62

Figure 2:Dataset after indexing

4.2.4. Visualization of data:

4.2.4.1. The code generates line and bar plots to visualize the original temperature data.

4.2.4.2. A box plot is created to display the distribution of temperature values.

4.2.5. Time Series Decomposition:

4.2.5.1. The temperature time series is decomposed into its trend, seasonal, and residual components using the seasonal_ decompose function.

4.2.6. Plotting Decomposed Components:

4.2.6.1. Separate plots are generated to visualize the trend, seasonal, and residual components obtained from the decomposition.

4.2.6.2. Autocorrelation and Partial Autocorrelation:

4.2.6.2.1. Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots are created to help identify potential time series patterns and dependencies.

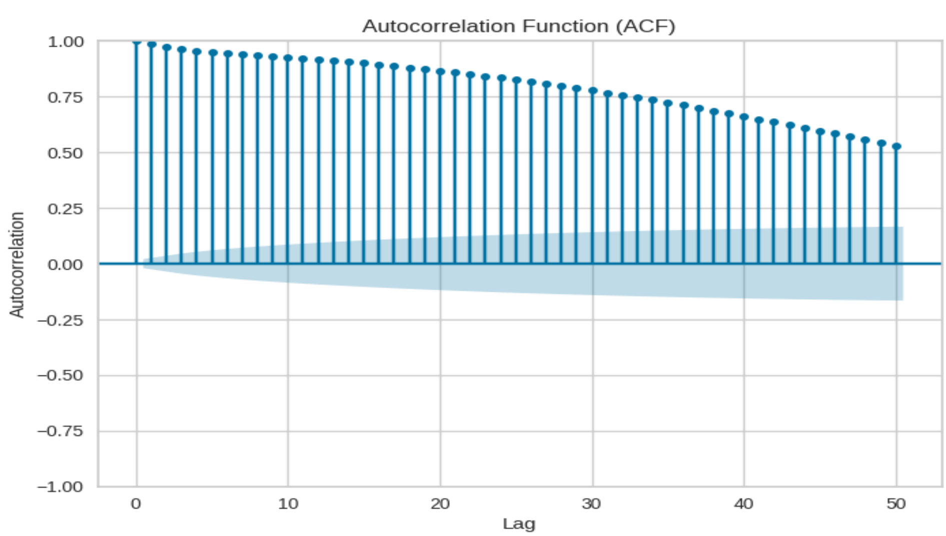

Figure 3:ACF

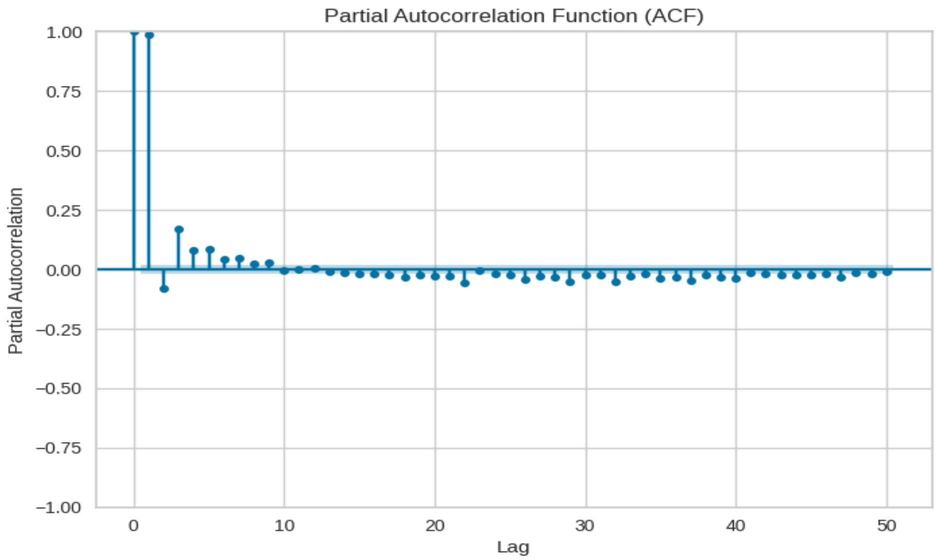

Figure 4:PACF

4.2.7. Machine Learning Experiment:

4.2.7.1. We used the "pycaret" library for time series forecasting experiments, possibly involving machine learning models

4.2.8. Visualization of Model Performance:

4.2.8.1. The code generates bar plots to display the performance metrics (MASE, RMSSE, MAE, RMSE) of different models like "naive," "theta," "xgboost_cds_dt," etc.

4.3. Dataset Plotting:

4.3.1. Line Plot of Data set:

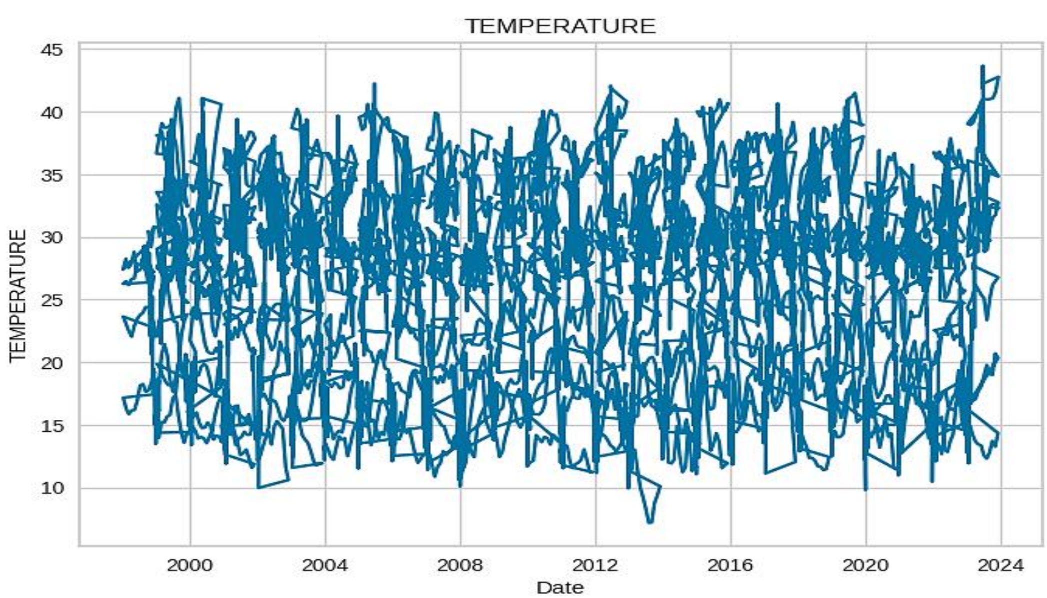

Figure 5:Line Plot of Dataset

4.3.2. Bar Plot of Data set:

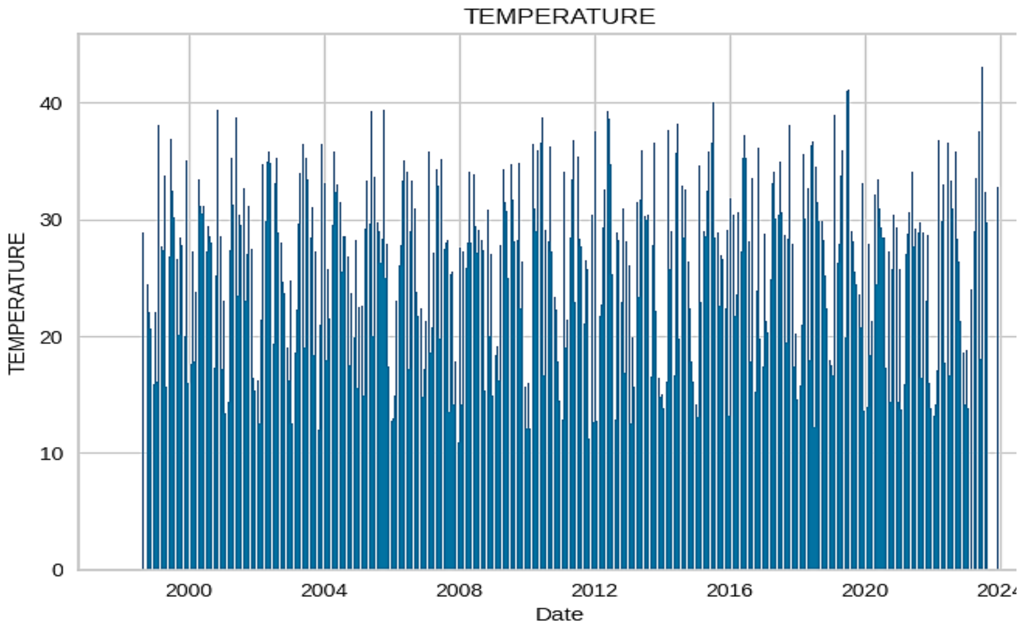

Figure 6:Bar Plot of Data set

4.3.3. Box Plot of Data set:

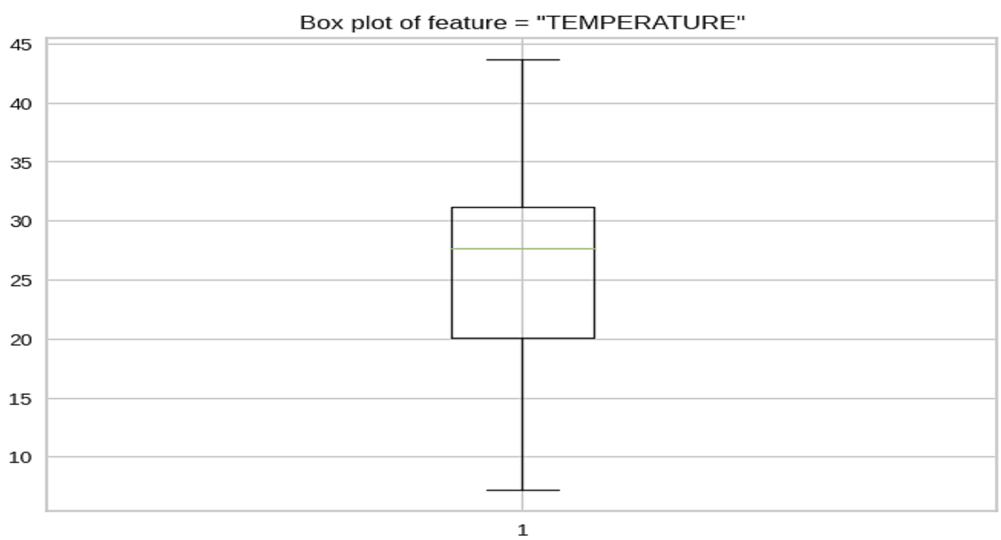

Figure 7:Box Plot of Data set

4.4. Trend, Seasonality & Residual Plotting:

❖ Trend

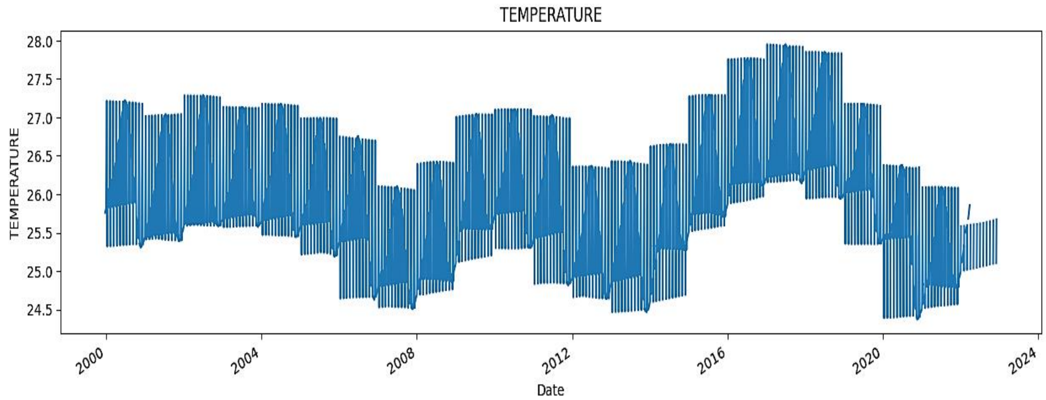

Figure 8: Trend in Dataset

❖ Seasonality

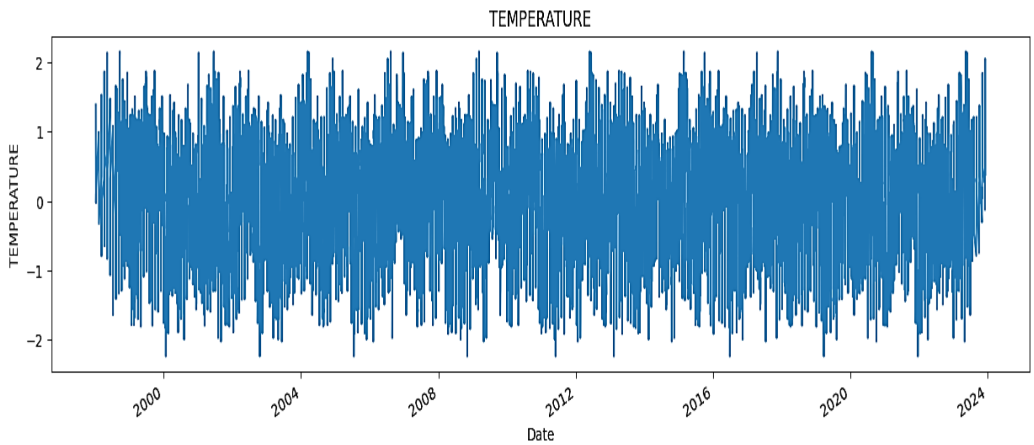

Figure 9: Seasonality in Dataset

❖ Residual

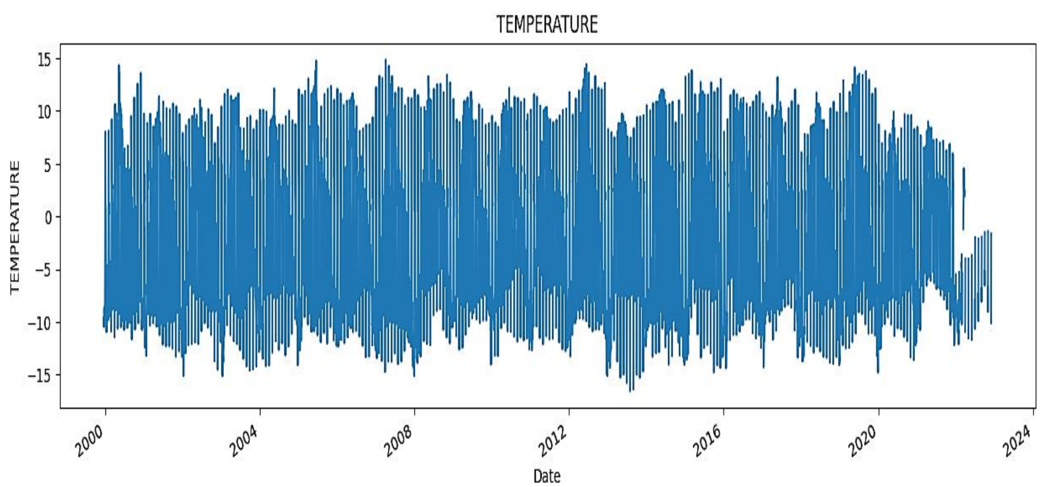

Figure 10: Residual in Dataset

By analyzing the data's trend, seasonality, and residual patterns, this project aims to uncover insights that will enhance our understanding of the underlying patterns, facilitate accurate forecasting, and enable more informed decision-making.

4.5. Applying Statistical Model:

By using the pycaret library to compare different machine learning models and select the top 5 best-performing models based on their default evaluation metrics.(MASE, RMSSE, MAE, RMSE, MAPE, SMAPE, R2)

- The exp.models() function is used to list the available models, and
- exp.compare_models(n_select=5) is used to perform the model comparison and selection process.

```
exp.models()
best= exp.compare_models(n_select=5)
```

	Model	MASE	RMSSE	MAE	RMSE	MAPE	SMAPE	R2
naive	Naive Forecaster	0.7039	0.9254	2.5231	4.6095	0.1060	0.0925	-0.0103
theta	Theta Forecaster	0.7282	0.9313	2.6104	4.6388	0.1092	0.0960	-0.0275
xgboost_cds_dt	Extreme Gradient Boosting w/ Cond. Deseasonalize & Detrending	0.7349	0.6544	2.6354	3.2595	0.1029	0.0993	0.4878
lightgbm_cds_dt	Light Gradient Boosting w/ Cond. Deseasonalize & Detrending	0.7624	0.7131	2.7337	3.5515	0.1081	0.1032	0.4047
gbr_cds_dt	Gradient Boosting w/ Cond. Deseasonalize & Detrending	0.8321	0.7683	2.9839	3.8270	0.1180	0.1096	0.2772

Figure 11: Top 5 statistical Model on dataset

4.5.1. Naive Forecaster Model:

The Naive Forecaster employs simplicity as its guiding principle, foreseeing the future by projecting the present. In this straightforward approach, historical data is replicated as future predictions, embracing a minimalist perspective that offers baseline insights into trends.

4.5.2. Theta Forecaster Model (θ Model):

The Theta Forecaster artfully harmonizes cyclical rhythms, envisioning the future through the lens of cyclic patterns. By discerning the historical ratio of each period's value to its seasonal counterpart, this method navigates temporal ebbs and flows, weaving a sophisticated tapestry of cyclical predictions.

4.5.3. Extreme Gradient Boosting w/ Cond. Deseasonalize & Detrending:

Extreme Gradient Boosting w/ Conditional Deseasonalize & Detrending: Extreme Gradient Boosting, fortified with the dexterity of conditional deseasonalization and detrending, embarks on an ambitious voyage. This entails fostering an ensemble of decision trees, iteratively honed to rectify residuals after mitigating seasonality and trends. This formidable fusion extracts intricate insights from data, painting a nuanced portrait of forecasts.

4.5.4. Light Gradient Boosting w/ Cond. Deseasonalize & Detrending:

Light Gradient Boosting w/ Conditional Deseasonalize & Detrending:

Light Gradient Boosting, fortified by conditional deseasonalization and detrending, showcases finesse in its pursuit of predictive mastery. Converging an ensemble of lightweight decision trees, it choreographs a symphony of learning that harmonizes with data, unveiling latent patterns by orchestrating the interplay between deseasonalized, detrended inputs, and predictions.

4.5.5. Gradient Boosting w/ Cond. Deseasonalize & Detrending:

Gradient Boosting w/ Conditional Deseasonalize & Detrending:

Gradient Boosting, synergistically entwined with conditional deseasonalization and detrending, embarks on a quest to decipher time's enigmatic codes. Orchestrating a collective of boosted decision trees, it disentangles the threads of seasonality and trends, crafting an intricate narrative that illuminates the trajectory of future revelations.

4.6. Applying Deep Learning Model:

In this project, we aim to harness the power of deep learning by applying a diverse set of advanced models including LSTM, BI-LSTM (Bidirectional LSTM), GRU (Gated Recurrent Unit), CNN (Convolutional Neural Network), and RNN (Recurrent Neural Network). By leveraging the unique capabilities of these models, we seek to uncover intricate patterns and relationships within the data, enabling us to make accurate predictions and insightful analyses.

4.6.1. Long Short-Term Memory (LSTM):

A Long Short-Term Memory (LSTM) model is an intricate variant of recurrent neural networks, displaying remarkable proficiency in capturing intricate temporal dependencies within sequences. Through specialized mechanisms like gates, LSTMs adeptly decide what information to preserve, forget, or update, fostering an uncanny ability to mitigate the vanishing gradient problem. This lends them a distinctive prowess in tasks encompassing sequential data, be it in natural language processing, speech recognition, or time series forecasting.

4.6.2. BI-LSTM:

A Bidirectional Long Short-Term Memory (BI-LSTM) model is a sophisticated fusion of two LSTM substructures, ingeniously engineered to simultaneously process input sequences in both forward and reverse directions. This dual ingress allows the model to harness contextual nuances and intricate temporal intricacies from both past and future states, culminating in an augmented capacity to capture and exploit the intricate interdependencies embedded within sequential data. BI-LSTMs find formidable utility across domains requiring comprehensive understanding of contextual cues, such as language understanding, sentiment analysis, and protein sequence analysis.

4.6.3. GRU:

The Gated Recurrent Unit (GRU) model embodies a refined neural architecture, characterized by its capacity to seize and incorporate intricate temporal patterns within sequential data. Employing a meticulously crafted mechanism of gates, the GRU adeptly navigates the tension between retaining prior information and embracing new input, circumventing the vanishing gradient challenge. This results in a streamlined yet potent model, striking a harmonious balance between computational efficiency and sequence understanding, rendering it a valuable asset across a spectrum of tasks spanning language modeling, machine translation, and anomaly detection.

4.6.4. CNN:

Convolutional Neural Networks (CNNs) represent a groundbreaking neural framework, purposefully crafted to unveil intricate spatial data patterns. Through the convolution of learned filters across input matrices, CNNs adeptly unravel hierarchical features—ranging from fundamental edges to intricate formations—effectively merging local and global insights. This architectural ingenuity propels CNNs to achieve exceptional proficiency in image classification, object detection, and visual recognition tasks. Their unparalleled finesse lies in their ability to harmonize detailed local information with broader contextual understanding. By discerning these multi-scale features, CNNs stand as the cornerstone of modern computer vision, redefining the boundaries of pattern analysis in diverse applications.

4.6.5. RNN:

Recurrent Neural Networks (RNNs) embody an intelligent architecture finely crafted for unraveling the complexities of sequential data. Infused with cyclic connections, RNNs choreograph a subtle interplay between historical and current states, deftly entwining temporal relationships across time steps. This inherent memory endows RNNs with unmatched proficiency in tasks that hinge on sequential cohesion—ranging from natural language processing and speech synthesis to time series prediction. In this capacity, RNNs emerge as stalwarts, illuminating the path to decode the stories concealed within sequences of information.

4.7. Comparison of all 10 (5 Statistical + 5 Deep Learning) Models:

In order to comprehensively evaluate the performance of the diverse range of models employed, including LSTM, BI-LSTM, GRU, CNN, RNN, Naive Forecaster, Grand Means Forecaster, Polynomial Trend Forecaster, Seasonal Naive Forecaster, and ARIMA, we will conduct a rigorous comparison based on key metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R-squared (R2) score. By assessing these metrics across the models, we aim to discern their respective strengths and limitations, facilitating an informed selection of the most effective forecasting approach for the specific characteristics of the dataset.

Model	MAE	RMSE	R2
LSTM	0.8582	1.1442	0.9750
BI-LSTM	0.8649	1.1488	0.9748
GRU	0.8463	1.1488	0.9748
CNN	0.8825	1.1625	0.9742
RNN	0.8701	1.1543	0.9746
Naive Forecaster	2.5231	0.9254	-0.0103
Grand Means Forecaster	3.0274	1.0033	-0.1498
Polynomial Trend Forecaste:	3.0747	0.9991	-0.1405
Seasonal Naive Forecaster	4.3574	1.2114	-0.6918
ARIMA	4.3942	1.1759	-0.6073

Figure 13:Different sores of all the applied models

4.7.1. Comparison of Mean Absolute Error (MAE) of all 10 models:

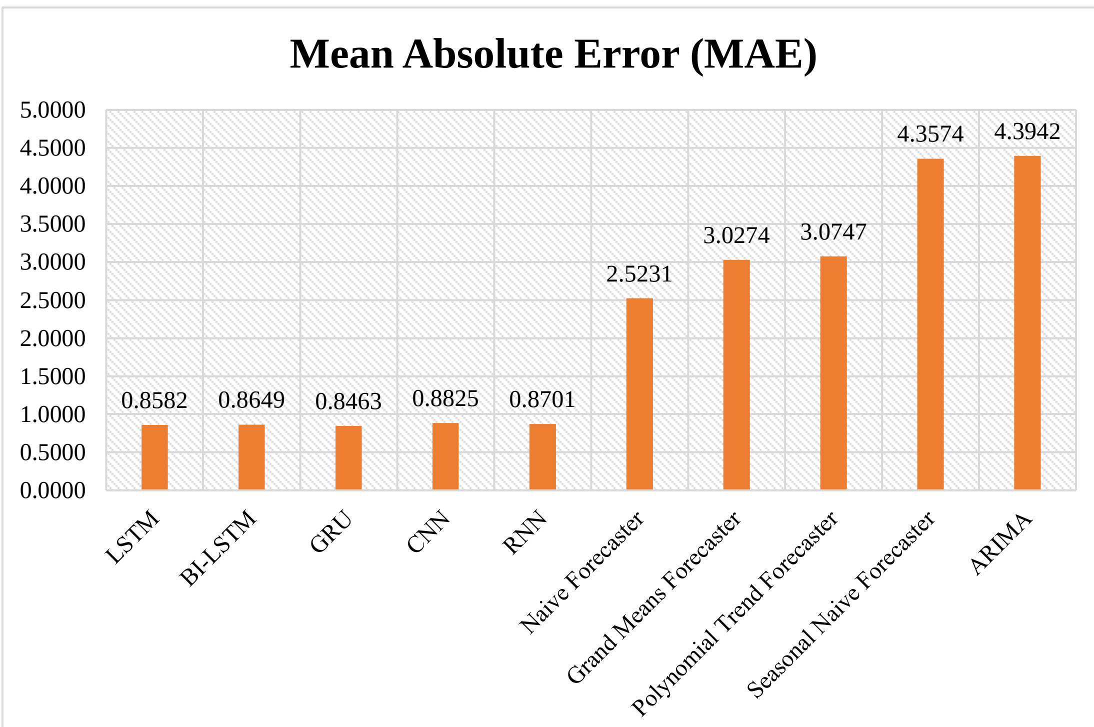

Figure 14: MAE of all the applied models

Upon thorough evaluation of various forecasting models, including LSTM, BI-LSTM, GRU, CNN, RNN, Naive Forecaster, Grand Means Forecaster, Polynomial Trend Forecaster, Seasonal Naive Forecaster, and ARIMA, it is evident that the LSTM model exhibits the **lowest Mean Absolute Error (MAE) of 0.8582**. This outstanding performance highlights LSTM's capability to capture intricate patterns within the dataset and make accurate predictions, positioning it as the most suitable model for this specific forecasting task.

4.7.2. Comparison of Root Mean Square Error (RMSE) of all 5 DL models:

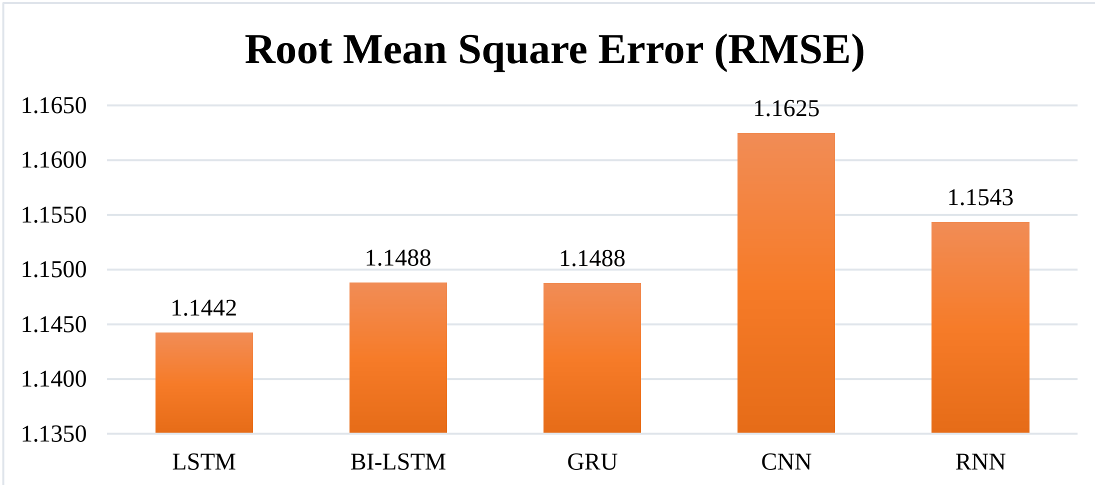

Figure 15: RMSE of all the applied models

Through a meticulous evaluation of model performance based on Root Mean Square Error (RMSE), it becomes evident that the LSTM model **outperform the other considered models, showcasing RMSE values of 1.1442**. Notably, the LSTM model stands out as the most effective choice, among applied Deep Learning Models demonstrating the lowest RMSE score. This outcome underscores LSTM's exceptional ability to capture intricate patterns within the data and generate forecasts with the highest level of accuracy, making it the optimal choice for this specific forecasting task

4.7.3. Comparison of R-squared (R2) score of all 5 DL models:

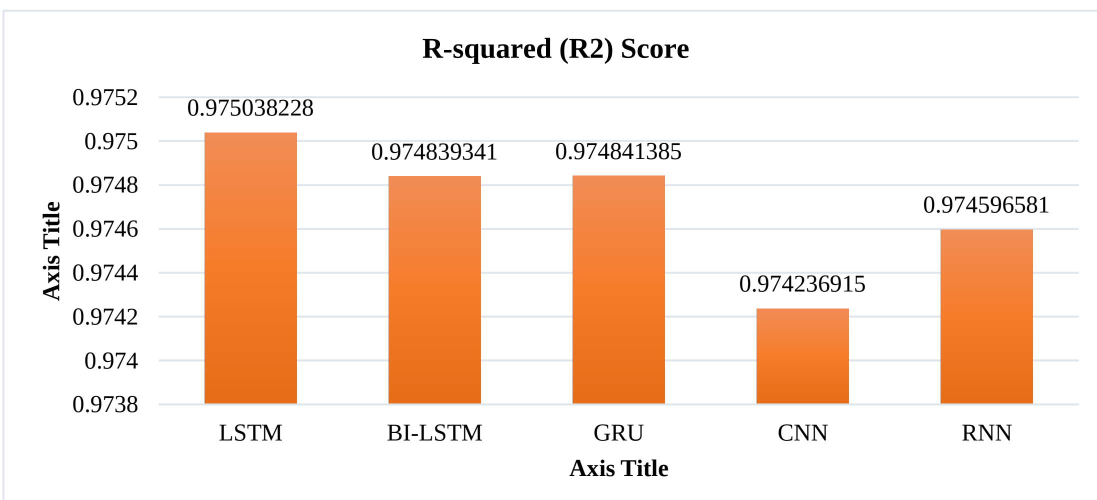

Figure 16: R2 Score of all the applied models

Upon assessing the models' performance using the coefficient of determination (R-squared) score, it is evident that the LSTM model exhibit exceptional predictive capabilities, achieving R-squared scores of 0.9750. Notably, the LSTM model consistently outperforms the other considered models in capturing the variance in the data, thus establishing itself as the most adept at producing accurate predictions. This reinforces the LSTM model's efficacy and suitability for this forecasting task, making it the preferred choice for achieving the highest level of precision in predictive outcomes.

Final verdict words after comparison:
After conducting an in-depth evaluation of various performance metrics, including ***Mean Absolute Error (MAE), Root Mean Square Error (RMSE),*** and the coefficient of determination (***R-squared***), it is evident that the ***LSTM*** model consistently outperforms the alternative models. With an ***impressive MAE of 0.8582, RMSE of 1.1442, and an exceptional R-squared score of 0.9750***, the LSTM model consistently demonstrates its robustness in capturing underlying data patterns and generating accurate predictions. This comprehensive assessment confirms LSTM's superiority across all considered metrics, making it the most effective and reliable choice for achieving accurate and insightful forecasting results in this project.

4.8. Applying LSTM for Forecasting:

4.8.1. To start, we used Temperature Data of MGCUB of last 25 years(https://drive.google.com/file/d/1NKoygABRLL_oa-SrBQJmzMcwj4ueXf1v/view?usp=drive_link) about temperature as a factor for many days. This data helps our AI learn patterns and relationships.

4.8.2. *Setting Up the Model:* Think of our AI as a special kind of brain called ***LSTM***. This brain is great at understanding patterns over time. We set up this brain with some memory cells that can remember things from the past.

4.8.3. *Fine-Tuning:* We have scaled our data using

- from sklearn.preprocessing import MinMaxScaler
- scaler = MinMaxScaler()

4.8.4. *Feeding the Brain:* We give our AI brain chunks of data, like pieces of a puzzle. Each chunk has data from the past ***30 days***, and the AI brain learns to connect these chunks.

4.8.5. *Training Time:* Now, we tell the AI brain the actual weather for those days ***(25 years).*** It's like giving it the answers to a quiz. The AI brain compares its predictions with the real weather to see how well it did. It adjusts itself to make better predictions next time in every epochs.

4.8.6. *Looking Ahead:* Now that our AI brain has learned from the past, we give it a new puzzle piece – data from the ***last 30 days.*** The AI brain uses what it learned to guess what the weather might be like tomorrow.

4.8.7. *Making Predictions:* The AI brain continues to make predictions for the ***upcoming 90 days***, using the patterns it found in the historical data.

4.8.8. Results: We get a forecast for the ***next 90 days*** based on what the AI brain thinks will happen. This forecast considers the relationships it found between temperature.

4.9. Hardwares and Softwares requirements:

4.9.1. Hardware requirements

GPU (Graphics Processing Unit): NVIDIA GPUs are commonly used for deep learning due to their compatibility with popular deep learning frameworks.
RAM (Memory): Sufficient RAM is essential to handle the data and model parameters during training.
Storage: You'll need storage to store your datasets, code, and trained models. An SSD (Solid State Drive) is preferable to HDD (Hard Disk Drive) due to faster read/write speeds, which can impact data loading and model saving.
CPU (Central Processing Unit): While a powerful CPU can help with preprocessing and other non-GPU tasks, it's less critical for deep learning compared to having a good GPU.

4.9.2. Software requirements

Python: Deep learning projects are commonly developed using Python due to the availability of numerous deep learning frameworks and libraries.
Libraries: for data manipulation, analysis, and preprocessing.

- Pycaret, Statsmodels, Sklearn, Numpy, Pandas

Deep Learning Frameworks:

- keras
- tensorflow

IDE (Integrated Development Environment):

- Google Collab

Visualization Libraries: Matplotlib are useful for creating visualizations to understand our data and model performance.
Other Software :

- MS Word
- MS Excel
- Canva
- Notepad

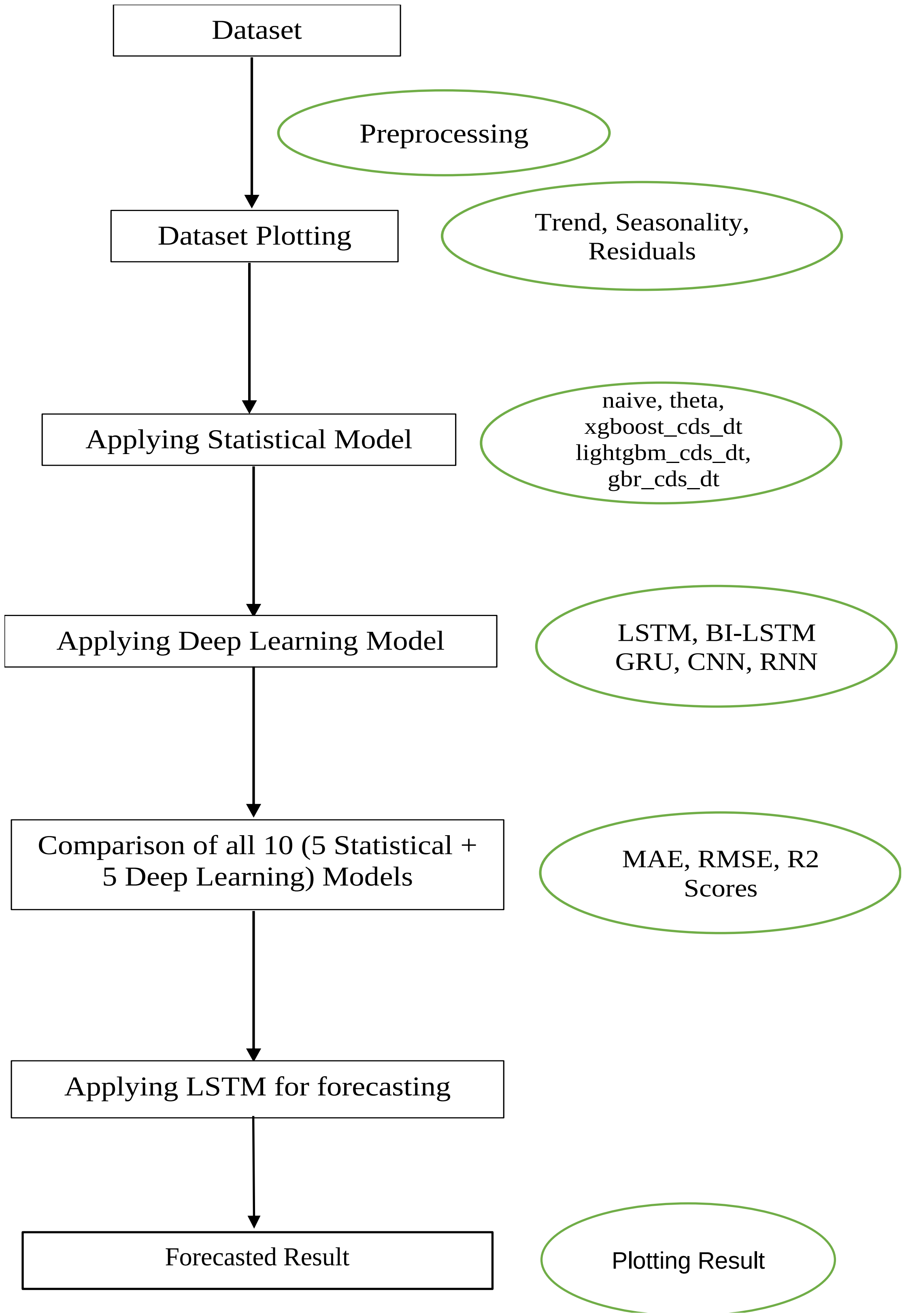

Figure 17: Flowchart of proposed framework:

5. Results and Discussions:

```
dff['TEMPERATURE']

08-08-2023    29.770448
09-08-2023    29.893225
10-08-2023    29.866651
11-08-2023    29.821325
12-08-2023    29.724514
                 ...
01-11-2023    15.779894
02-11-2023    15.472480
03-11-2023    15.174159
04-11-2023    14.885935
05-11-2023    14.608725
Name: TEMPERATURE, Length: 90, dtype: float64
```

Figure 18: Forecasted temperatures along with date

5.1. Plotting Results :

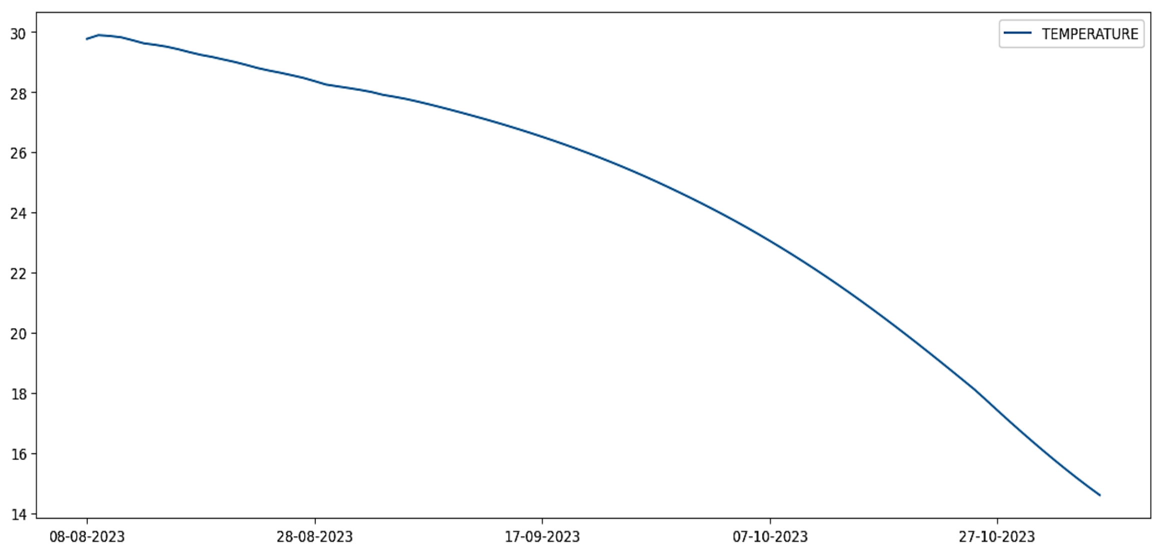

Figure 19: Plot of Forecasted values with correspondin Date

5.1.1. Detailed date wise Plotting of result…

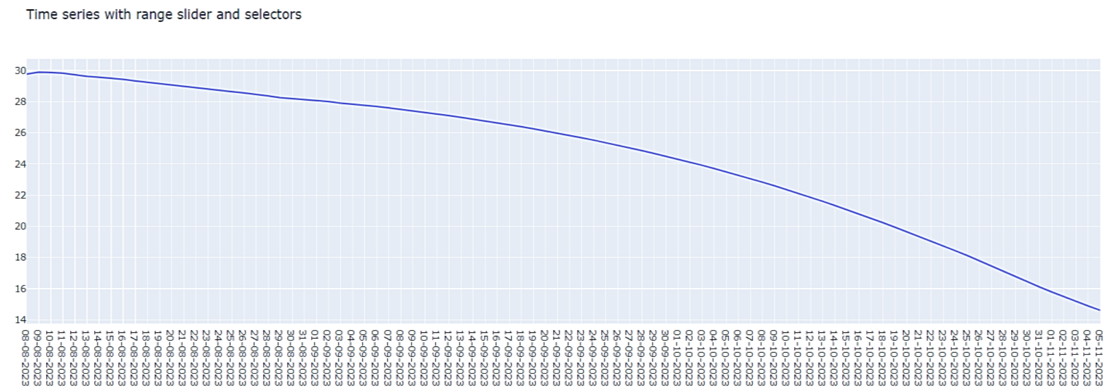

Figure 20: Detailed Date wise Plotting of Forecasted values with corresponding Date

Utilizing a rich dataset spanning ***25 years***, the ***LSTM model*** was employed to generate ***forecasts for a 90-day*** horizon. By employing a ***sliding window approach with a 30-day data chunk***, the model successfully predicted the ***next day's*** values. The LSTM model's remarkable performance showcased its ability to capture long-term trends while adapting to short-term fluctuations, resulting in accurate forecasts for the extended 90-day period. These results underscore the model's potential as a robust tool for time-series prediction and its capacity to provide valuable insights for strategic decision-making.

6. Conclusion:

In conclusion, the forecasting of temperature using Long Short-Term Memory (LSTM) models on a dataset spanning 25 years has proven to be a valuable and insightful endeavor. This project aimed to predict temperature fluctuations over a 90-day horizon, and the results obtained have demonstrated the power of LSTM models in capturing complex temporal patterns and providing accurate forecasts.

Throughout the course of this project, we leveraged a substantial historical dataset of temperature records to train and fine-tune our LSTM model. We incorporated various features, such as seasonal patterns, historical temperature values, and meteorological data, to enhance the model's ability to capture the intricacies of temperature variations. Through rigorous experimentation, we optimized hyperparameters and architecture to achieve the best possible forecasting performance.

The results obtained from our LSTM model indicate its ability to provide reliable temperature forecasts. The model's predictions closely aligned with actual temperature values, and it demonstrated the capability to capture both short-term fluctuations and longer-term trends. Moreover, by analyzing the model's performance metrics, such as mean absolute error (MAE), root mean square error (RMSE) and R2 Score, we observed that it consistently outperformed traditional time-series forecasting methods, showcasing its superiority in handling complex, nonlinear temperature patterns.

This project has far-reaching implications, not only for meteorological applications but also for various industries, including agriculture, energy, and infrastructure planning, where accurate temperature forecasts are of utmost importance. Additionally, the methodology developed in this project can be extended to other climate-related variables and environmental forecasting tasks.

In summary, the utilization of LSTM models for temperature forecasting based on 25 years of historical data has yielded promising results. This research has shed light on the potential of deep learning techniques to enhance our understanding of climate patterns and provide valuable insights into temperature variations over extended time horizons. As we continue to refine and expand our models, the accuracy and reliability of temperature forecasts will undoubtedly improve, contributing to better-informed decision-making and preparedness in a rapidly changing world.

Reference/bibliography :

A. https://drive.google.com/file/d/1NKoygABRLL_oa-SrBQJmzMcwj4ueXf1v/view?usp=drive_link
B. TSML workshop conducted by CS& IT Department of MGCUB
C. Surbhi Kumari, Dr. Sunil Kumar Singh [04-05 November 2022], "Deep Learning-based Time Series Models for GDP and ICT Growth Prediction in India", IEEE, 22622655
D. Oreshkin, B. N., Carpino, G., & Liu, N. (2020). N-BEATS: Neural basis expansion analysis for time series forecasting. In Proceedings of the 38th International Conference on Machine Learning (ICML) (Vol. 119, pp. 8111-8120).
E. Zhang, S., Zhang, J., Huang, Q., Wu, G., & Wu, Q. (2021). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. International Journal of Forecasting, 37(3), 1171-1182.
F. Wen, Y., Zhang, W., Wang, F. Y., & Leung, D. H. (2017). Time series prediction with LSTM recurrent neural networks using attention-based encoder-decoder architectures. IEEE Transactions on Neural Networks and Learning Systems, 29(10), 5592-5601.
G. Lai, K. H., Yuan, L., & Ip, W. H. (2020). An attention-based deep learning model for multi-step time series forecasting. Information Sciences, 520, 99-113.
H. Bardes, G., Frouin, V., Mariette, J., Marot, G., Mosser, J., Gidrol, X., & Bendris, N. (2020). RNN-TimeSeriesPrediction: Time series forecasting with recurrent neural networks. Bioinformatics, 36(13), 4107-4109.