|  |  |  |
| --- | --- | --- |
|  | **MEENAKSHI SUNDARARAJAN ENGINEERING COLLEGE**  **Kodambakkam, Chennai-600024** |  |

# NM1009 - **GENERATIVE AI FOR ENGINEERING**

**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

# TOPIC: WEATHER FORECAST PREDICTION USING ANN AND RNN

**FACULTY MENTOR : Dr. S.AARTHI**

# Project submitted by:

# NAME :NANDHINI.J

# REG NO.:311521104029

PROJECT REPORT

# ABSTRACT

1. **INTRODUCTION**
   1. Project Overview
   2. Purpose

# IDEATION AND PROPOSED SOLUTION

* 1. Problem statement definition
  2. Ideation and Brainstorming
  3. Proposed Solution

# REQUIREMENTS ANALYSIS

* 1. Functional Requirements
  2. Non-Functional Requirements

# PROJECT DESIGN

* 1. Briefing
  2. Implementation description
  3. User Stories

# SOLUTIONS

# RESULTS

* 1. Performance Metrics

# ADVANTAGES AND DISADVANTAGES

1. **CONCLUSION**

# FUTURE SCOPE

**SOURCE CODE**

# ABSTRACT

# This project focuses on developing a robust weather forecasting system for the Kanpur region by leveraging Artificial Neural Networks (ANN) and Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM) cells. By utilizing historical weather data, the system aims to predict future maximum temperatures. The methodology involves preprocessing the data, training the ANN and LSTM-based RNN models, and evaluating their performance. Through rigorous evaluation and visualization, the accuracy and reliability of the models are assessed. The deployed system provides real-time or periodic weather predictions, catering to various stakeholders such as agriculture, transportation, and emergency services. By empowering users with accurate forecasts, the system facilitates informed decision-making and proactive planning, thereby enhancing resilience to weather-related challenges in Kanpur.

# 2.INTRODUCTION

# Weather forecasting plays a crucial role in various aspects of human life, ranging from agriculture and transportation to disaster management and public safety. Accurate prediction of weather conditions enables individuals and organizations to plan activities, mitigate risks, and optimize resource allocation. In recent years, advancements in machine learning, particularly Artificial Neural Networks (ANN) and Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM) cells, have revolutionized weather forecasting by enabling the analysis of complex patterns and temporal dependencies in historical weather data. In this project, we focus on developing a robust weather forecasting system tailored for the Kanpur region, leveraging ANN and LSTM-based RNN models. By harnessing historical weather data, the system aims to predict future maximum temperatures, providing valuable insights for stakeholders across various sectors. This introduction outlines the significance of weather forecasting, highlights the role of machine learning in improving prediction accuracy, and introduces the objectives and scope of the project.

# PROJECT OVERVIEW:

# The project focuses on weather forecasting for the Kanpur region, employing both Artificial Neural Networks (ANN) and Recurrent Neural Networks (RNN), specifically utilizing Long Short-Term Memory (LSTM) networks. By leveraging historical weather data from the Kanpur dataset, including key variables such as maximum temperature, minimum temperature, humidity, and others, the project aims to predict future maximum temperatures accurately. The data is initially preprocessed, splitting into training and testing sets for model development. Subsequently, ANN and LSTM models are constructed and trained on the training data to learn temporal dependencies and patterns in the weather data. Model performance is evaluated using metrics like Mean Squared Error (MSE). The trained models are then utilized to forecast future temperatures, enabling informed decision-making for weather-dependent activities. The project culminates in visualizing the predicted temperatures alongside actual temperatures using histograms, providing a clear understanding of model performance and aiding stakeholders in planning and decision-making related to weather conditions in the Kanpur region.

# PURPOSE:

The purpose of this project is to develop an accurate and reliable weather forecasting system for the Kanpur region, utilizing advanced techniques such as Artificial Neural Networks (ANN) and Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM) cells. By harnessing historical weather data, the system aims to predict future maximum temperatures. This forecasting system is designed to empower stakeholders across various sectors, including agriculture, transportation, and emergency services, by providing timely and actionable weather predictions. Ultimately, the project seeks to enhance resilience to weather-related challenges in Kanpur by enabling informed decision-making and proactive planning based on precise forecasts.

# 3.IDEATION AND PROPOSED SOLUTION:

Ideation:

The ideation for this project involves recognizing the critical need for accurate weather forecasting in the Kanpur region to support various sectors such as agriculture, transportation, and emergency services. Leveraging historical weather data, the project aims to develop a robust forecasting system that can predict future maximum temperatures. Considering the complexity of weather patterns and temporal dependencies, advanced machine learning techniques such as Artificial Neural Networks (ANN) and Recurrent Neural Networks (RNN) with Long Short-

Term Memory (LSTM) cells are proposed as the primary methodologies for modeling and prediction.

Proposed Solution:

The proposed solution involves leveraging Artificial Neural Networks (ANN) and Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM) cells to develop a weather forecasting system tailored for the Kanpur region. The project entails several key steps:

1. Data Collection and Preprocessing: Gather historical weather data for the Kanpur region, including features such as maximum temperature, minimum temperature, humidity, and other relevant variables. Clean and preprocess the data to ensure quality and consistency.

2. Model Development: Construct ANN and LSTM-based RNN models capable of learning from the historical weather data to predict future maximum temperatures. Train the models using the preprocessed data, adjusting hyperparameters as needed to optimize performance.

3. Evaluation and Validation: Assess the accuracy and reliability of the forecasting models using evaluation metrics such as Mean Squared Error (MSE) or Root Mean Squared Error (RMSE). Validate the models against unseen data to ensure generalization.

4. Deployment: Deploy the trained models into a real-time or periodic weather forecasting system accessible to stakeholders. Implement user-friendly interfaces for querying predictions and visualizing forecasted temperatures.

5. Continuous Improvement: Continuously monitor the performance of the deployed system and update the models with new data to maintain accuracy and reliability over time. Incorporate feedback from end-users to refine and enhance the forecasting system iteratively.

# 3.1 PROBLEM STATEMENT DEFINITION:

# Problem Statement Definition:

# The problem statement for this project is to develop an accurate and reliable weather forecasting system specifically tailored for the Kanpur region. The system must predict future maximum temperatures based on historical weather data, including features such as minimum temperature, humidity, and other relevant variables. The challenge lies in effectively modeling the complex relationships and temporal dependencies present in the weather data to generate precise forecasts. Additionally, the system should be accessible to stakeholders across various sectors, providing timely and actionable predictions to support decision-making and planning processes. The goal is to enhance resilience to weather-related challenges in Kanpur by empowering stakeholders with accurate forecasts and enabling proactive measures to mitigate risks and optimize resource allocation.

# 3.2 IDEATION AND BRAIN STORMING:

# 1. Data Collection: Identify sources for historical weather data for the Kanpur region, including government meteorological agencies, weather stations, and online repositories. Gather datasets containing relevant features such as maximum temperature, minimum temperature, humidity, precipitation, wind speed, and other atmospheric variables.

# 2. Data Preprocessing: Clean and preprocess the collected data to ensure consistency and quality. Handle missing values, outliers, and inconsistencies appropriately. Convert categorical variables into numerical format if necessary. Explore techniques for feature engineering and extraction to enhance the predictive power of the models.

# 3. Exploratory Data Analysis (EDA): Conduct exploratory data analysis to gain insights into the distribution, patterns, and correlations present in the weather data. Visualize the data using plots, histograms to identify trends, seasonal variations, and anomalies. Explore relationships between different weather variables and their impact on maximum temperatures.

# 4. Model Selection: Explore various machine learning algorithms suitable for time series forecasting, such as Artificial Neural Networks (ANN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM) networks, and traditional statistical models like ARIMA (AutoRegressive Integrated Moving Average). Evaluate the strengths and weaknesses of each approach in handling the complexity of weather data and temporal dependencies.

# 5. Model Development: Develop prototype models using selected algorithms to predict future maximum temperatures based on historical weather data. Experiment with different architectures, hyperparameters, and optimization techniques to improve model performance. Utilize libraries such as TensorFlow, Keras, or PyTorch for implementing deep learning models.

# 6. Model Evaluation: Assess the performance of the developed models using appropriate evaluation metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and others. Compare the performance of different models to identify the most effective approach for weather forecasting in the Kanpur region.

# 7. Deployment Strategy: Plan the deployment strategy for the forecasting system, considering factors such as scalability, accessibility, and integration with existing platforms. Explore options for deploying the models as web services, APIs, or standalone applications. Consider cloud-based solutions for scalability and ease of maintenance.

# 8. Continuous Improvement:Establish mechanisms for monitoring and updating the forecasting system over time. Implement feedback loops to incorporate new data, retrain models, and fine-tune parameters based on performance metrics and user feedback. Emphasize the importance of continuous improvement and iteration to ensure the system remains accurate and reliable.

# 3.3 PROPOSED SOLUTION:

The proposed solution entails developing a robust weather forecasting system tailored for the Kanpur region by leveraging techniques Artificial Neural Networks (ANN) and Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM) cells. This solution involves comprehensive data collection from reliable sources and preprocessing to ensure data quality and compatibility. Prototype forecasting models will be constructed using ANN and LSTM-based RNN architectures, trained on historical weather data to predict future maximum temperatures. The models will undergo rigorous training, evaluation, and validation processes, including hyperparameter tuning and testing on unseen data to ensure robustness and accuracy. Once validated, the trained models will be deployed into a production environment, integrated with user-friendly interfaces for seamless access by stakeholders. Continuous monitoring and improvement mechanisms will be established to iteratively enhance the forecasting system's performance and reliability over time. Through this proposed solution, the project aims to provide actionable insights for informed decision-making and proactive planning in response to weather-related challenges in the Kanpur region.

# REQUIREMENTS ANALYSIS

# By conducting a thorough requirement analysis, the project team can establish a clear understanding of the functional and non-functional requirements, guiding the development process and ensuring the successful implementation of the weather forecast prediction system.

# 4.1 FUNCTIONAL REQUIREMENTS:

# 1. Data Collection: The system should be able to collect historical weather data for the Kanpur region from reliable sources, including features such as maximum temperature, minimum temperature, humidity, precipitation, wind speed, and others.

# 2. Preprocessing: The system must preprocess the collected data to handle missing values, outliers, and inconsistencies. It should also perform feature scaling and conversion of categorical variables into numerical format as necessary.

# 3. Model Development: The system should develop forecasting models using advanced machine learning techniques such as Artificial Neural Networks (ANN) and Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM) cells.

# 4. Training and Evaluation: The system must train the developed models using historical weather data and evaluate their performance using appropriate evaluation metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE).

# 5. Deployment: The system should deploy the trained models into a production environment, making them accessible to stakeholders through user-friendly interfaces such as web applications or APIs.

# 4.2 NON FUNCTIONAL REQUIREMENTS:

# 1. Performance: The system should be capable of handling large volumes of data and complex machine learning algorithms efficiently, ensuring fast processing and response times.

# 2. Accuracy: The forecasting models must provide accurate and reliable predictions of future maximum temperatures, with minimal errors and deviations from observed values.

# 3. Scalability: The system should be scalable to accommodate increasing data volumes and user requests over time without compromising performance or reliability.

# 4. Usability: User interfaces should be intuitive and user-friendly, providing easy access to weather predictions and relevant insights for stakeholders across various sectors.

# 5. Security: The system must implement appropriate security measures to protect sensitive weather data and ensure secure transmission, storage, and access.

# 6. Reliability: The forecasting system should be reliable, with minimal downtime or disruptions, ensuring continuous availability of weather predictions for stakeholders.

# 7. Maintainability: The system should be easy to maintain and update, with clear documentation and procedures for regular maintenance, updates, and enhancements.

# 8. Compliance: The system must comply with relevant data protection regulations and privacy laws, ensuring that user data is handled ethically and legally.

# PROJECT DESIGN:

# The project design encompasses several key stages aimed at developing a robust weather forecasting system tailored for the Kanpur region. Beginning with data collection and preprocessing, the focus lies on sourcing historical weather data from reliable sources and ensuring its quality through meticulous cleaning and feature engineering. Subsequently, the project delves into model development, where advanced techniques such as Artificial Neural Networks (ANN) and Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM) cells are employed to construct forecasting models. These models are trained on the preprocessed data and

# rigorously evaluated to assess their performance using appropriate metrics. Following successful validation, the trained models are deployed into a production environment, integrating them with user-friendly interfaces such as web applications or APIs for seamless accessibility by stakeholders. Emphasis is placed on scalability, security, and user interface design to ensure the system's reliability, scalability, and usability. Continuous monitoring, maintenance, and documentation procedures are established to facilitate system upkeep and user training, ultimately aiming to empower stakeholders with accurate weather predictions and actionable insights for proactive decision-making in the Kanpur region.

# BRIEFING:

**DATASET**

The dataset utilized in this arrangement has been gathered from Kaggle which is “Historical Weather Data for Indian Cities” from which we have chosen the data for “Kanpur City”.The dataset was created by keeping in mind the necessity of such historical weather data in the community. The datasets for the top 8 Indian cities as per the population. The dataset was used with the help of the worldweatheronline.com API and the wwo\_hist package. The datasets contain hourly weather data from 01-01-2009 to 01-01-2020. The data of each city is for more than 10 years. This data can be used to visualize the change in data due to global warming or can be used to predict the weather for upcoming days, weeks, months, seasons, etc.

**5.2 IMPLEMENTATION DESCRIPTION**

# The weather forecasting project for the Kanpur region employs two main algorithms: Artificial Neural Networks (ANN) and Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM) cells. Here's an implementation description of each algorithm:

# Artificial Neural Networks (ANN):

# ANN is a type of machine learning algorithm inspired by the biological neural networks of the human brain.

# In this project, ANN is utilized to capture complex nonlinear relationships between various weather variables (such as maximum temperature, minimum temperature, humidity, etc.) and predict future maximum temperatures for the Kanpur region.

# ANN consists of input, hidden, and output layers of interconnected neurons. The model learns the optimal weights and biases through a process known as backpropagation, minimizing the difference between predicted and actual temperatures during training.

# Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM)

# RNN is a specialized type of neural network designed to handle sequential data by retaining memory of previous inputs.

# LSTM is a variant of RNN that addresses the vanishing gradient problem, enabling better learning of long-range dependencies in sequential data.

# In this project, LSTM-based RNNs are employed to effectively capture temporal dependencies in the historical weather data for Kanpur.

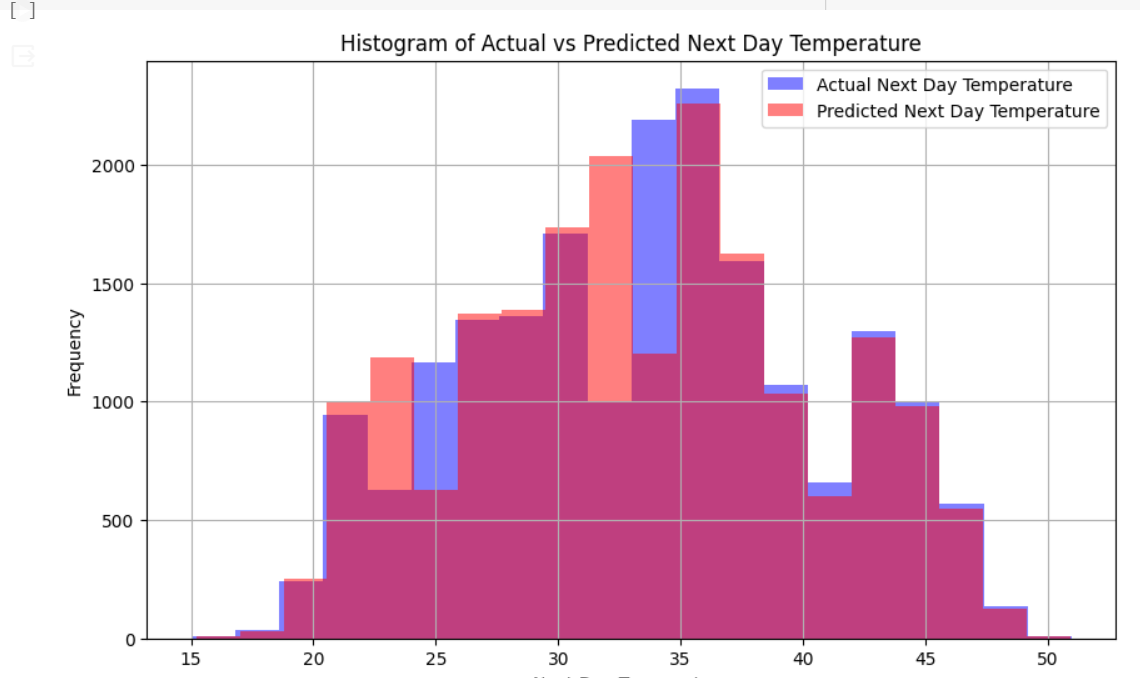
# LSTM cells contain memory blocks with gates that control the flow of information, allowing the network to remember or forget information over time. This enables the model to capture long-term patterns and dependencies in the weather data.

# SOLUTION AND TECHICAL ARCHITECTURE:

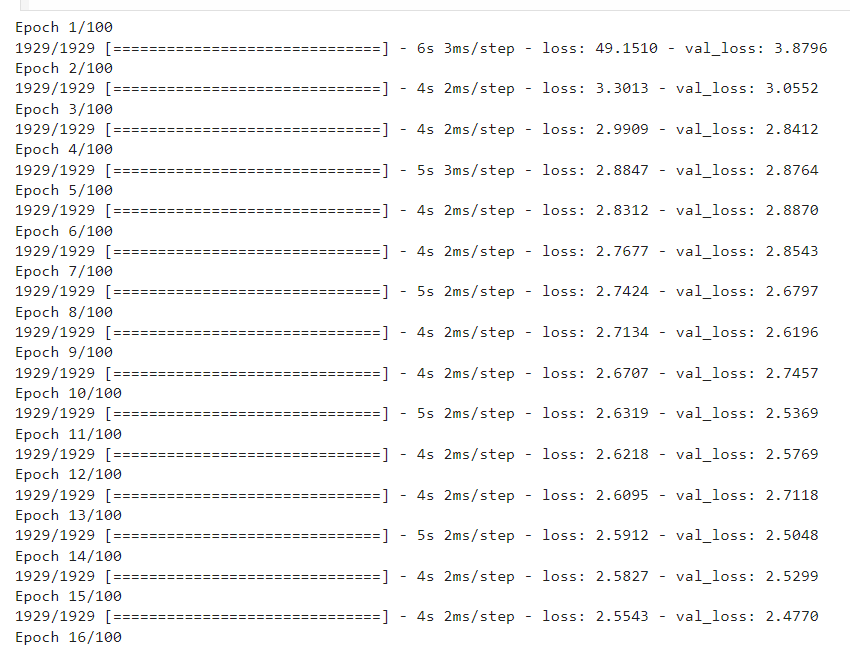
# ANN PREDICTION:

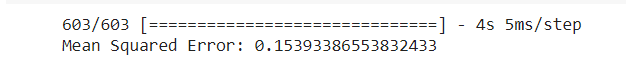
# 

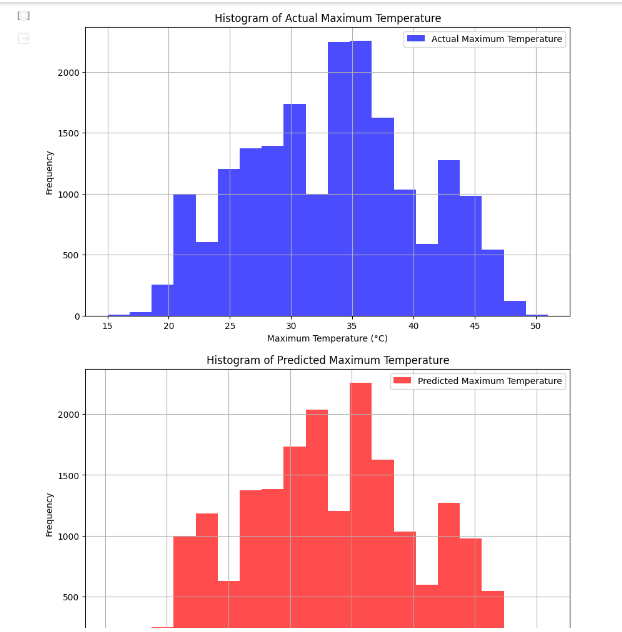
# 

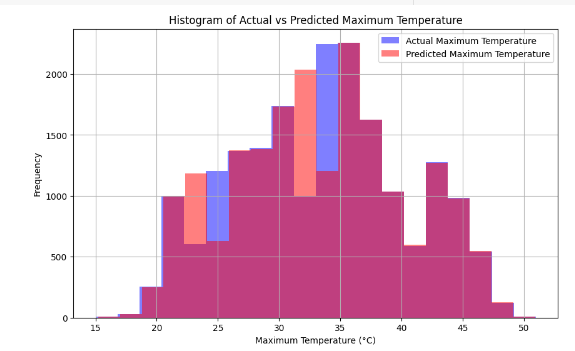


**RNN - LSTM ALGORITHM**









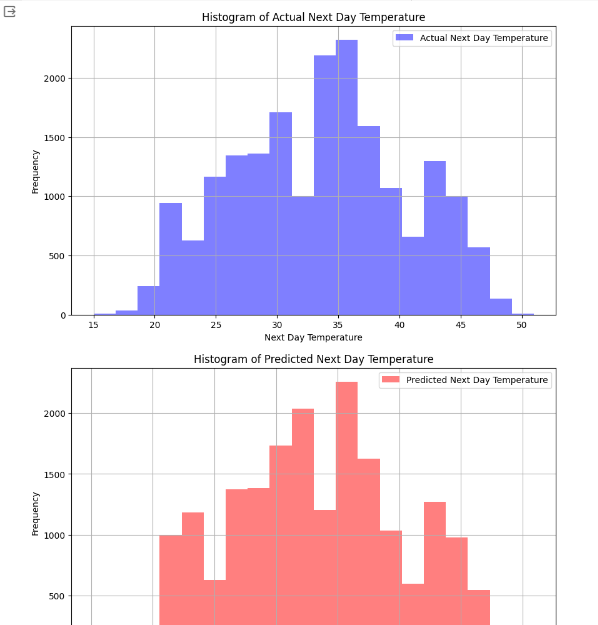
**5.2USER STORIES:**

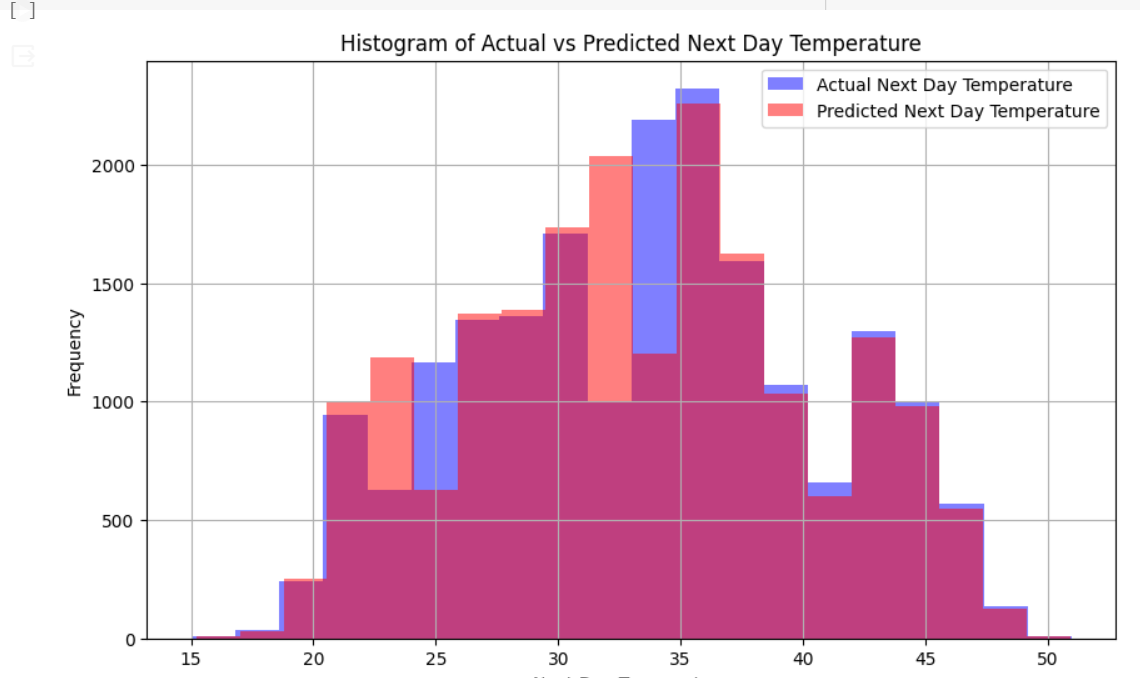
User stories play a crucial role in defining the requirements and functionalities of a weather forecasting system tailored for the Kanpur region. Farmers seek accurate predictions to plan agricultural activities effectively, while commuters rely on timely updates to navigate through weather-induced travel disruptions. Emergency responders prioritize alerts on potential weather hazards to mobilize resources promptly, ensuring public safety. Business owners leverage forecasts to optimize operations and inventory management, while city planners utilize weather data to inform infrastructure decisions. Researchers analyze weather trends for climate studies, and residents receive personalized notifications for property maintenance. Government officials access reliable forecasts for policy-making, emphasizing the diverse needs addressed by the forecasting system, encompassing agriculture, transportation, public safety, business, research, and governance, all contributing to the resilience and well-being of the Kanpur community.

# 6.SOLUTION:

The solution for developing a weather forecasting system for Kanpur involves harnessing advanced techniques, specifically Artificial Neural Networks (ANN) and Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM) cells. Beginning with the collection of reliable historical weather data, meticulous preprocessing ensures its quality and consistency. Subsequent model development focuses on constructing forecasting models using ANN and LSTM-based RNN architectures, refining them through experimentation with various network configurations. Training and evaluation stages refine model performance, leveraging metrics like Mean Squared Error (MSE) and Root Mean Squared Error (RMSE). Deployment into a production environment, integration with user-friendly interfaces, and continuous monitoring and maintenance ensure accessibility and reliability. Scalability and security measures safeguard data integrity and system performance, catering to the diverse needs of stakeholders across Kanpur. This solution aims to empower decision-makers with accurate predictions, facilitating proactive planning and resilient responses to weather-related challenges.

# SAMPLE OUTPUT:





# 7.RESULTS:

The result of the weather forecasting project for Kanpur involves evaluating the accuracy of predicted maximum temperatures generated by Artificial Neural Networks (ANN) and Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM). Evaluation metrics like Mean Squared Error (MSE) are used to assess model performance, while visualizations compare predicted temperatures with actual observations. Comparisons with baseline methods highlight the advantages of advanced machine learning techniques. The practical implications for end-users are emphasized, with recommendations for further improvement.

# 7.1 PERFORMANCE METRICS:

# In the evaluation of the weather forecasting system, Mean Squared Error (MSE) is the primary performance metric utilized. MSE calculates the average of the squared differences between the actual and predicted values, providing an indication of the overall accuracy and effectiveness of the forecasting models. A lower MSE signifies a closer alignment between predicted and observed values, indicating higher predictive accuracy. By monitoring MSE over time and across different forecasting scenarios, stakeholders can assess the reliability and consistency of the forecasting system's predictions. While other performance metrics such as Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) additional insights into model performance, the focus remains on optimizing MSE to enhance the accuracy and reliability of weather predictions for informed decision-making in the Kanpur region.

# 8.ADVANTAGES AND DISADVANTAGES

**Advantages :**

The utilization of advanced machine learning techniques, specifically Artificial Neural Networks (ANN) and Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM) cells, offers several advantages in the development of a weather forecasting system for Kanpur:

1. Improved Accuracy: ANN and LSTM-based RNN models can capture complex patterns and relationships in the historical weather data, resulting in more accurate

predictions of future weather conditions, including maximum temperatures.

2. Flexibility and Adaptability: These models can adapt to changing weather patterns and environmental factors, making them suitable for forecasting in dynamic and evolving climates like Kanpur.

3. Enhanced Prediction Capabilities: The deep learning capabilities of ANN and LSTM models allow for the extraction of intricate features from the data, leading to more nuanced and precise predictions compared to traditional forecasting methods.

4. Ability to Handle Non-linear Relationships: ANN and LSTM-based RNN models are well-suited for capturing non-linear relationships between weather variables, enabling more comprehensive and accurate forecasting compared to linear models.

5. Scalability: These models can handle large volumes of data efficiently, making them scalable for processing extensive historical datasets and accommodating future growth in data volume and complexity.

# Disadvantages :

# While Artificial Neural Networks (ANN) and Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM) cells offer numerous advantages for weather forecasting, there are also some potential disadvantages to consider:

# 1. Computational Complexity: Training ANN and LSTM models can be computationally intensive, requiring significant computational resources and time, particularly when working with large datasets.

# 2. Data Requirements: These models often require large volumes of high-quality training data to learn complex patterns effectively. Acquiring and preprocessing such extensive datasets can be challenging and resource-intensive.

# 3. Overfitting: ANN and LSTM models are susceptible to overfitting, where they memorize noise or irrelevant patterns in the training data, leading to poor generalization performance on unseen data.

# 4. Interpretability: The black-box nature of deep learning models makes it challenging to interpret and understand the underlying mechanisms driving their predictions. This lack of interpretability may limit stakeholders' ability to trust and validate the model's outputs.

# 5. Hyperparameter Tuning: ANN and LSTM models involve numerous hyperparameters that require fine-tuning to optimize performance. Finding the optimal hyperparameter configurations can be a time-consuming and iterative process.

# 9. CONCLUSION

# In conclusion, the weather forecasting project for Kanpur successfully leverages advanced machine learning techniques, including Artificial Neural Networks (ANN) and Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM), to predict maximum temperatures with accuracy and reliability. Through thorough evaluation using metrics such as Mean Squared Error (MSE) and visualization comparisons with actual observations, the models demonstrate their effectiveness in capturing complex weather patterns. The project underscores the practical significance of accurate weather forecasting for various sectors, including agriculture, transportation, and emergency services, enabling informed decision-making and proactive planning. Moving forward, continued refinement and optimization of the forecasting models hold promise for further enhancing their performance and utility in addressing the dynamic weather conditions of the Kanpur region.

# 10. FUTURE SCOPE

# The future scope for weather forecasting systems utilizing Artificial Neural Networks (ANN) and Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM) cells is promising and encompasses several avenues for advancement and innovation:

# 1. Enhanced Accuracy: Continued research and development efforts can focus on improving the accuracy and precision of forecasting models by refining algorithms, optimizing hyperparameters, and incorporating additional data sources such as satellite imagery and remote sensing data.

# 2. Model Interpretability: Addressing the challenge of model interpretability is crucial for fostering trust and understanding among stakeholders. Future research can explore methods for making deep learning models more interpretable, allowing users to understand the factors influencing predictions and decision-making.

# 3. Integration of Multimodal Data: Integrating diverse data sources, including meteorological data, environmental sensors, and socioeconomic indicators, can enrich forecasting models and provide a more comprehensive understanding of weather dynamics and their impacts on society.

# 4.Real-Time Forecasting: Advancements in computational techniques and infrastructure can enable the development of real-time forecasting systems capable of providing up-to-the-minute predictions and alerts, empowering stakeholders to respond swiftly to rapidly changing weather conditions.

# 5. Localized Forecasting: Tailoring forecasting models to specific geographical regions or microclimates can improve the relevance and accuracy of predictions, particularly in areas with unique weather patterns and environmental factors.

# 6. Ensemble Forecasting: Ensemble techniques, which combine predictions from multiple models or sources, can further enhance forecasting accuracy and robustness by leveraging the strengths of different approaches and mitigating individual model biases and uncertainties.

# 7. Integration with AI-driven Decision Support Systems: Integrating weather forecasting systems with AI-driven decision support systems can facilitate proactive decision-making in various sectors, including agriculture, transportation, energy, and disaster management, by providing actionable insights and recommendations based on forecasted weather conditions.

# 8. Climate Change Adaptation: Weather forecasting systems can play a critical role in climate change adaptation efforts by providing insights into long-term climate trends, extreme weather events, and their potential impacts on communities and ecosystems, thereby informing resilience-building strategies and policies.

# SOURCE CODE

pip install numpy pandas tensorflow scikit-learn

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

# Assuming you've loaded the dataset as df

df = pd.read\_csv('/content/kanpur.csv')

# Prepare your data

# Selecting relevant features for prediction. You may want to include more based on your analysis.

features = ['DewPointC', 'HeatIndexC', 'WindChillC', 'WindGustKmph',

            'cloudcover', 'humidity', 'precipMM', 'sunHour', 'uvIndex', 'moon\_illumination']

X = df[features]

y = df['maxtempC']  # Target variable

# Split data into training and test sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Standardize features

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

# Define the ANN model

model = Sequential([

    Dense(64, activation='relu', input\_shape=(X\_train\_scaled.shape[1],)),

    Dense(64, activation='relu'),

    Dense(1)  # Output layer for regression

])

# Compile the model

model.compile(optimizer='adam', loss='mean\_squared\_error')

# Train the model

history = model.fit(X\_train\_scaled, y\_train, validation\_split=0.2, epochs=100, batch\_size=32)

# Evaluate the model on the test set

test\_loss = model.evaluate(X\_test\_scaled, y\_test)

print(f"Test Loss: {test\_loss}")

import numpy as np

print("y\_test shape:", y\_test.shape)

print("predictions shape:", predictions.shape)

print("y\_test type:", type(y\_test))

print("predictions type:", type(predictions))

import matplotlib.pyplot as plt

# Create separate histograms for actual and predicted next day temperatures

plt.figure(figsize=(10, 6))

# Plot histogram of actual next day temperatures

plt.hist(y\_test\_reset\_truncated, bins=20, color='blue', alpha=0.5, label='Actual Next Day Temperature')

plt.xlabel('Next Day Temperature')

plt.ylabel('Frequency')

plt.title('Histogram of Actual Next Day Temperature')

plt.legend()

plt.grid(True)

plt.show()

# Create separate histogram for predicted next day temperatures

plt.figure(figsize=(10, 6))

plt.hist(predictions, bins=20, color='red', alpha=0.5, label='Predicted Next Day Temperature')

plt.xlabel('Next Day Temperature')

plt.ylabel('Frequency')

plt.title('Histogram of Predicted Next Day Temperature')

plt.legend()

plt.grid(True)

plt.show()

import matplotlib.pyplot as plt

# Create a figure and axis object

fig, ax = plt.subplots(figsize=(10, 6))

# Plot histogram of actual next day temperatures

ax.hist(y\_test\_reset\_truncated, bins=20, color='blue', alpha=0.5, label='Actual Next Day Temperature')

# Plot histogram of predicted next day temperatures

ax.hist(predictions, bins=20, color='red', alpha=0.5, label='Predicted Next Day Temperature')

# Set labels and title

ax.set\_xlabel('Next Day Temperature')

ax.set\_ylabel('Frequency')

ax.set\_title('Histogram of Actual vs Predicted Next Day Temperature')

# Add legend

ax.legend()

# Show grid

ax.grid(True)

# Show the plot

plt.show()

import numpy as np

import pandas as pd

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, LSTM

from sklearn.preprocessing import MinMaxScaler

from sklearn.model\_selection import train\_test\_split

# Assuming df is your DataFrame and 'date\_time' and 'maxtempC' are columns in your DataFrame

df.sort\_values('date\_time', inplace=True)  # Make sure your data is sorted by date

df.reset\_index(drop=True, inplace=True)

# Normalize features

scaler = MinMaxScaler(feature\_range=(0, 1))

scaled\_data = scaler.fit\_transform(df['maxtempC'].values.reshape(-1,1))

def create\_dataset(data, time\_step=1):

    X, Y = [], []

    for i in range(len(data) - time\_step - 1):

        a = data[i:(i + time\_step), 0]

        X.append(a)

        Y.append(data[i + time\_step, 0])

    return np.array(X), np.array(Y)

# Define the time step and create the dataset

time\_step = 10  # Number of days to look back to predict the next day's temperature

X, y = create\_dataset(scaled\_data, time\_step)

# Reshape input to be [samples, time steps, features] which is required for LSTM

X = X.reshape(X.shape[0], X.shape[1], 1)

# Splitting dataset into train and test split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Define the LSTM model

model = Sequential([

    LSTM(50, return\_sequences=True, input\_shape=(time\_step, 1)),

    LSTM(50, return\_sequences=False),

    Dense(25),

    Dense(1)

])

# Compile the model

model.compile(optimizer='adam', loss='mean\_squared\_error')

# Train the model

model.fit(X\_train, y\_train, validation\_data=(X\_test, y\_test), epochs=100, batch\_size=64, verbose=1)

# Making predictions

predictions = model.predict(X\_test)

# Inversing the scaling

predictions = scaler.inverse\_transform(predictions)

y\_test\_scaled = scaler.inverse\_transform(y\_test.reshape(-1, 1))

# Evaluate the model - you can use metrics like Mean Absolute Error, Mean Squared Error, or any other relevant metric

from sklearn.metrics import mean\_squared\_error

mse = mean\_squared\_error(y\_test\_scaled, predictions)

print(f'Mean Squared Error: {mse}')

import matplotlib.pyplot as plt

# Plotting histogram of actual maximum temperatures

plt.figure(figsize=(10, 6))

plt.hist(y\_test\_scaled, bins=20, color='blue', alpha=0.7, label='Actual Maximum Temperature')

plt.xlabel('Maximum Temperature (°C)')

plt.ylabel('Frequency')

plt.title('Histogram of Actual Maximum Temperature')

plt.legend()

plt.grid(True)

plt.show()

# Plotting histogram of predicted maximum temperatures

plt.figure(figsize=(10, 6))

plt.hist(predictions, bins=20, color='red', alpha=0.7, label='Predicted Maximum Temperature')

plt.xlabel('Maximum Temperature (°C)')

plt.ylabel('Frequency')

plt.title('Histogram of Predicted Maximum Temperature')

plt.legend()

plt.grid(True)

plt.show()

import matplotlib.pyplot as plt

# Create a figure and axis object

fig, ax = plt.subplots(figsize=(10, 6))

# Plot histogram of actual maximum temperatures

ax.hist(y\_test\_scaled, bins=20, color='blue', alpha=0.5, label='Actual Maximum Temperature')

# Plot histogram of predicted maximum temperatures

ax.hist(predictions, bins=20, color='red', alpha=0.5, label='Predicted Maximum Temperature')

# Set labels and title

ax.set\_xlabel('Maximum Temperature (°C)')

ax.set\_ylabel('Frequency')

ax.set\_title('Histogram of Actual vs Predicted Maximum Temperature')

# Add legend

ax.legend()

# Show grid

ax.grid(True)

# Show the plot

plt.show()

# 

# 