A

Mini Project

On

RAINFALL PREDICTION USING NEURAL NETWORK ANALYSIS

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

VOBILISHETTY SURAJ KUMAR (207R1A05B8)

Under the Guidance of

K.SHILPA

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi) Recognized Under Section 2(f) & 12(B) of the UGCAct.1956, Kandlakoya (V), Medchal Road, Hyderabad-501401.

2020-2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled **"RAINFALL PREDICTION USING NEURAL NETWORK ANALYSIS"** being submitted by **VOBILISHETTY SURAJ KUMAR (207R1A05B8)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2023-24.

The results embodied in this project have not been submitted to any other University or Institute for the award of any degree or diploma.

K.SHILPA	**Dr. A. Raji Reddy**
(Assistant Professor)	DIRECTOR
INTERNAL GUIDE	

Dr. K. Srujan Raju	**EXTERNAL EXAMINER**
HOD	

Submitted for viva voce Examination held on ____________________

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. I take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

I take this opportunity to express my profound gratitude and deep regard tomy guide **K.SHILPA**, Assistant Professor for his exemplary guidance, monitoring andconstant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

I also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **G. Vinesh Shanker, Dr. J. Narasimharao, Ms. K. Shilpa,& Dr. K. Maheswari** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

I am also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

I am obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. I also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. I am grateful for their constant support and help.

Finally, I would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. I sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

VOBILISHETTY SURAJ KUMAR (207R1A05B8)

ABSTRACT

Rainfall prediction is an important and challenging task in meteorology. Rainfall is predicted using different models with their combination, observation, trends of knowledge and patterns. Rainfall can be predicted using various machine learning techniques. Artificial neural networks (ANN) are the valuable and attractive soft computing method for prediction. ANN is based on self-adaptive mechanism in which the model learns from historical data capture functional relationships between data and make predictions on current data. ARIMA(Auto Regressive Integrated moving Average) is used in predicting the rainfall. The accurate prediction of rainfall is a major criterion for managing the water resources. The prediction accuracy error is measured using MAPE and RMSE. The prediction accuracy is measured using confusion matrix and RMSE. The results show that the prediction model based on ANN indicates acceptable accuracy.

LIST OF FIGURES/TABLES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

TABLE OF CONTENTS

1. INTRODUCTION

1.INTRODUCTION

PROJECT SCOPE

This project is titled “Rainfall Prediction Using Neural Network Analysis”.

The project scope encompasses the development of a rainfall prediction system using neural network analysis, focusing on improving accuracy, adaptability, and usability. It acknowledges the potential for future expansion and aims to address the needs of various stakeholders who rely on accurate rainfall forecasts for various applications.

PROJECT PURPOSE

The purpose of the project is to harness machine learning techniques to improve rainfall prediction accuracy, which has far-reaching implications for agriculture, water resource management, disaster preparedness, urban planning, and scientific research. It contributes to informed decision-making and helps address the challenges posed by weather variability and climate change.

PROJECT FEATURES

The "Rainfall Prediction Using Neural Network Analysis" project incorporates several features that define its functionality and capabilities. These features are designed to achieve the project's objectives and improve rainfall prediction accuracy. Here are the key project features: Geographical Adaptability, Accuracy Assessment etc.These features collectively contribute to the project's ability to provide accurate and reliable rainfall predictions, making it a valuable tool for various sectors and stakeholders that rely on rainfall forecasts for planning and decision-making.

2.SYSTEM ANALYSIS

2.SYSTEM ANALYSIS

SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done.

PROBLEM DEFINITION

The "Rainfall Prediction Using Neural Network Analysis" project aims to overcome these challenges by leveraging advanced machine learning techniques, particularly neural network analysis, to improve the accuracy and adaptability of rainfall predictions. It seeks to provide a reliable tool for various stakeholders tó make data-driven decisions, adapt to changing climate conditions, and enhance resilience in the face of weather-related challenges.

EXISTING SYSTEM

The existing system for rainfall prediction typically involves the use of statistical models, such as time series analysis, regression analysis, and autoregressive integrated moving average models..These models rely on historical rainfall data and other relevant weather factors, such as temperature, humidity, and wind speed, to predict future rainfall patterns. These models may have limited accuracy and may not be able to capture complex nonlinear relationships between different weather factors

DISADVANTAGES OF EXISTING SYSTEM

Following are the disadvantages of existing system:

- Dependency on historical data
- Limited accuracy
- Limited interpretability
- Difficulty in determining optimal network architecture
- Computational resources

LITERATURE SURVEY

Hu (1964) initiated the implementation of ANN, an important soft computing methodology in weather forecasting. Since the last few decades, ANN a voluminous development in the application field of ANN has opened up new avenues to the forecasting task involving environment related phenomenon (Gardener and Dorling, 1998; Hsiesh and Tang, 1998). Michaelides et al (1995) compared the performance of ANN with multiple linear regressions in estimating missing rainfall data over Cyprus. Kalogirou et al (1997) implemented ANN to reconstruct the rainfall over the time series over Cyprus. Lee et al(1998) applied ANN in rainfall prediction by splitting the available data into homogenous subpopulations. Wong et al (1999) constructed fuzzy rules bases with the aid of SOM and back-propagation neural networks and then with the help of the rule base developed predictive model for rainfall over Switzerland using spatial interpolation. Toth et al. (2000) compared short time rainfall prediction models for real-time flood forecasting. Different structures of auto-regressive moving average (ARMA) models, ANN and nearest- neighbours approaches were applied for forecasting storm rainfall occurring in the Sieve River basin, Italy, in the period 1992-1996 with lead times varying from 1 to 6 h. The ANN adaptive calibration application proved to be stable for lead times longer than 3 hours, but inadequate for reproducing low rainfall events. Koizumi (1999) employed an ANN model using radar, satellite and weather-station data together with numerical products generated by the Japan Meteorological Agency (JMA) Asian Spectral Model and the model was trained using 1-year data. It was found that the ANN skills were better than the persistence forecast (after 3 h), the linear regression forecasts, and the numerical model precipitation prediction. As the ANN model was trained with only 1 year data, the results were limited. The author believed that the performance of the neural network would be improved when more training data became available. It is still unclear to what extent each predictor contributed to the forecast and to what extent recent observations might improve the forecast.

Abraham et al. (2001) used an ANN with scaled conjugate gradient algorithm (ANN-SCGA) and Evolving Fuzzy neural network (EFUNN) for predicting the rainfall time series. In the study, monthly rainfall was used as input data for training model. The authors analyzed 87 years of rainfall data in Kerala, a state in the southern part of the Indian Peninsula. The empirical results showed that neuro-fuzzy systems were efficient in terms of having better performance time and lower error rates 5 compared to the pure neural network approach. Nevertheless, rainfall is one of the 20 most complex and difficult elements of the hydrology cycle to understand and to model due to the tremendous range of variation over a wide range of scales both in space and time (French et al., 1992).

PROPOSED SYSTEM

The proposed system for rainfall prediction using neural network analysis involves the use of machine learning algorithms, specifically artificial neural networks (ANNs).ANNs are capable of learning complex nonlinear relationships between input and output variables, making them ideal for predicting complex weather patterns.ANNs can also handle large datasets and can continuously learn and adapt to new data, improving their prediction accuracy over time.

ADVANTAGES OF THE PROPOSED SYSTEM

The following are advantages of existing system:

- Improved decision-making
- Faster processing
- Improved accuracy
- Flexibility
- Automated analysis

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and a business proposalis put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This isto ensure that the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis:

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on a project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminaryinvestigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication that the system is economically possible for development.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical require ements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

HARDWARE & SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENTS:

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- Processor : Intel I3 or above
- Hard disk : 500GB
- RAM : 4GB and above
- Input devices : Keyboard, mouse.

SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

- Operating system : Microsoft Windows , Linux or Mac
- Languages : Python(Jupyter Notebook)
- Tools : Anaconda – Jupyter.

3.ARCHITECTURE

3.ARCHITECTURE

PROJECT ARCHITECTURE

This project architecture shows the procedure followed for classification,starting from input to final prediction.

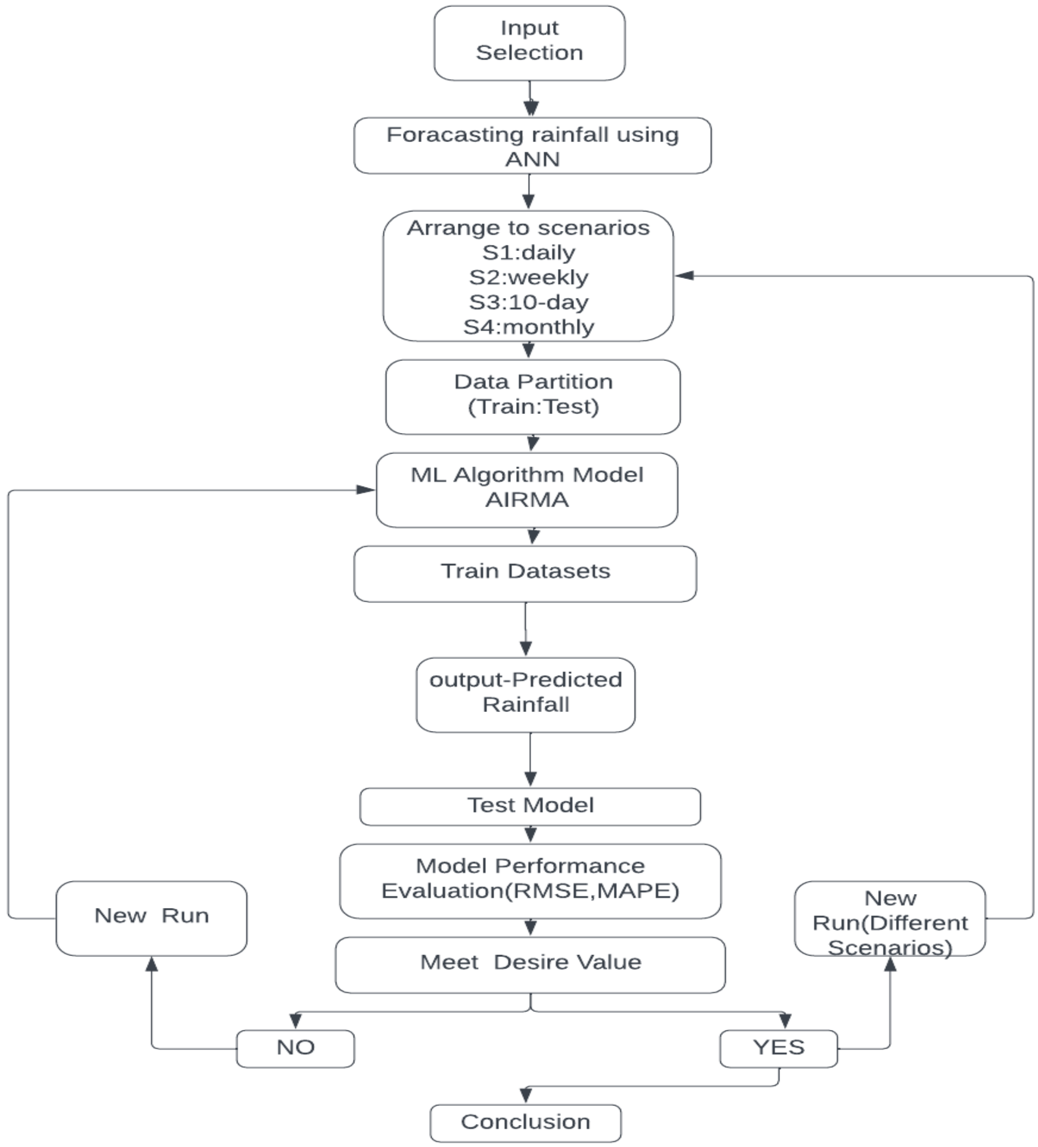


Figure 3.1: Project Architecture of Rainfall Prediction Using Neural Network Analysis.

DESCRIPTION

The project focuses on the important and challenging task of rainfall prediction in meteorology. It employs various machine learning techniques, including Artificial Neural Networks (ANN) and Auto Regressive Integrated Moving Average (ARIMA) models, to predict rainfall patterns. The project aims to improve the accuracy of rainfall prediction, a critical factor in managing water resources ..The model takes sequence of daily rainfall intensities and geographical parameters. After initial pre-processing, input goes to a deep network, which is a ANN (ARTIFICAL NEURAL NETWORK) and a wide network consists of convolutions. The model is trained using joint training approach, considering outputs from deep and wide networks simultaneously.

USE CASE DIAGRAM

In the use case diagram, we have basically one actor who is the user in the trained model.

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of usersthe system has. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures.

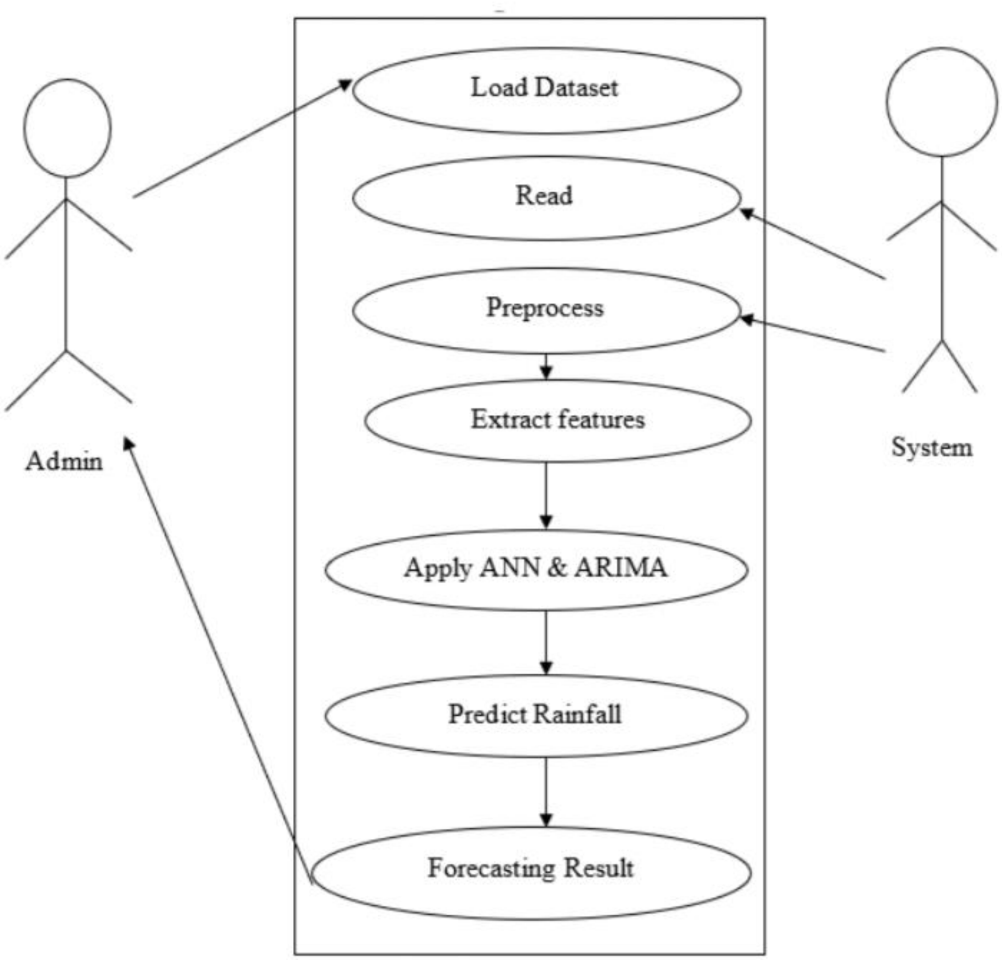


Figure 3.2: Use case diagram for Rainfall prediction using neural network

CLASS DIAGRAM

Class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations(or methods), and the relationships among objects.

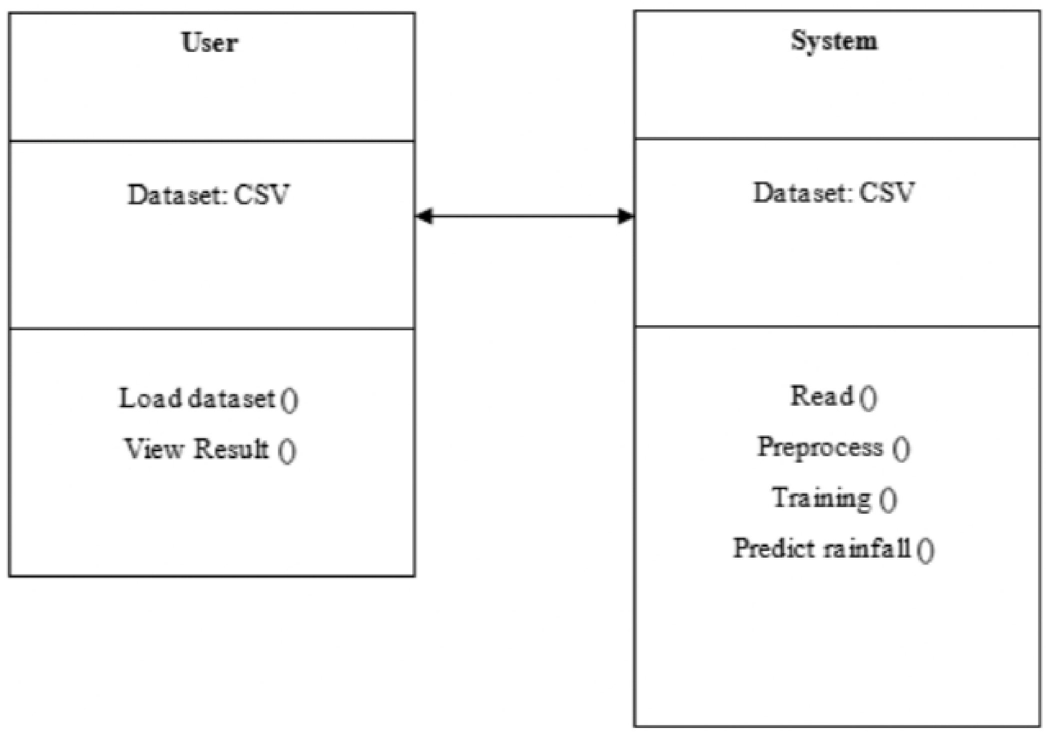


Figure 3.3: Class Diagram for Rainfall Prediction Using Neural Network Analysis

SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development.

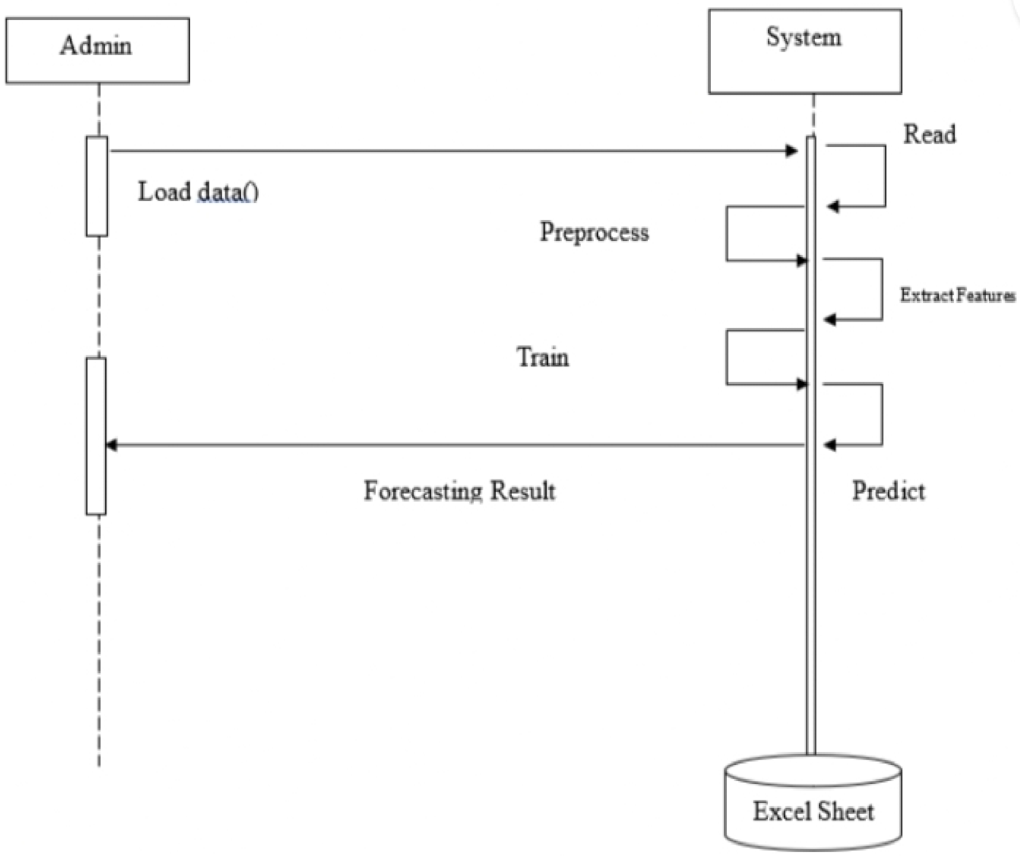


Figure 3.4: Sequence Diagram for Rainfall Prediction Using Neural Network Analysis

ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. They can alsoinclude elements showing the flow of data between activities through one or more datastores.

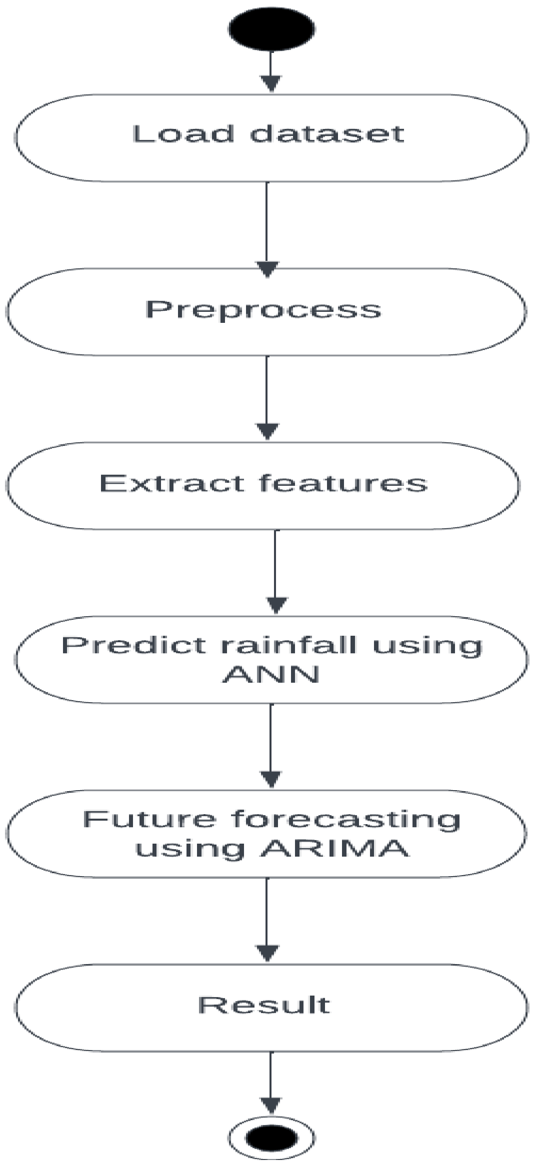


Figure 3.5: Activity Diagram Rainfall Prediction Using Neural Network Analysis.

4.IMPLEMENTATION

4. SAMPLE CODE

```
conda install python-graphviz
*****************IMPORT LIBRARIES***********************
import warnings
warnings.filterwarnings('ignore')
import os
import shutil
import numpy as np
import pandas as pd
import seaborn as sns
import plotly.graph_objs as go
import plotly.offline as py
import matplotlib
import matplotlib.pyplot as plt
#from mpl_toolkits.basemap import Basemap
import matplotlib.pyplot as plt
from matplotlib.patches import Polygon
import matplotlib.patches as mpatches
from matplotlib.collections import PatchCollection
import plotly.figure_factory as ff
from IPython.display import HTML, display
from IPython.core import display as ICD
from plotly.offline import init_notebook_mode, iplot
init_notebook_mode(connected=True)
import Artificial_Neural_Networks as ANN
import ARIMA
import math
from itertools import groupby
%matplotlib inline
from keras.models import Sequential
```

```
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Flatten
from keras.layers.convolutional import Conv1D
from keras.layers.convolutional import MaxPooling1D
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
from IPython.display import SVG
from keras.utils.vis_utils import model_to_dot
from importlib import reload
import itertools
```

************************USEFUL FUNCTIONS************************

```
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
def mean_absolute_percentage_error(y_true, y_pred):
    y_true, y_pred = np.array(y_true), np.array(y_pred)
    return np.mean(np.abs((y_true - y_pred) / y_true)) * 100
def root_mean_squared_error(y_true, y_pred):
    mse = mean_squared_error(y_true, y_pred)
    rmse = np.sqrt(mse)
    return rmse
def calculate_performance(y_true, y_pred):
    mse = mean_squared_error(y_true, y_pred)
    mae = mean_absolute_error(y_true, y_pred)
    mape = mean_absolute_percentage_error(y_true, y_pred)
    rmse = root_mean_squared_error(y_true, y_pred)
    return round(mse, 3), round(mae, 3), round(mape, 3), round(rmse, 3)
```

```
*******************DATASET*****************
PATH = 'Dataset/rainfall_data_1901_to_2002.xlsx'
data = pd.read_excel(PATH)


*********************PREPROCESS DATA************
data = data.drop(columns='vlookup')
data = data[data['Year'].notnull()]
data['Year'] = data.Year.astype('int')
data.index = range(len(data))
m_data = data[data['State'] == 'Maharashtra']
m_data = m_data.drop(columns='State')
districts = m_data.District.unique()
years = list(range(1901, 2003))
months = data.columns[3:]
year_month = [str(year) + '_' + month for year in years for month in months]
dates = pd.date_range(start='1901-01', freq='MS', periods=len(years)*12)
maharashtra_data = pd.DataFrame({'Year_Month': year_month})
maharashtra_data['Date'] = dates
maharashtra_data[['Year', 'Month']] = maharashtra_data['Year_Month'].str.split('_', n=1,
expand=True)
maharashtra_data = maharashtra_data.drop(columns=['Year_Month'])
for district in districts:
    df = m_data[m_data.District == district].drop(columns=['District', 'Year'])
    df = df.to_numpy().reshape((len(years) * len(months), 1))[:,0]
    maharashtra_data[district] = df
maharashtra_data.head()
m_data = maharashtra_data.copy()
districts_of_interest = ['Kolhapur', 'Latur']
months_of_interest = ['Jun', 'Jul', 'Aug', 'Sep']
rainfall_season_data = m_data[m_data.Month.isin(months_of_interest)]
```

```
rainfall_season_data = rainfall_season_data[['Date', 'Year', 'Month'] +
districts_of_interest]
rainfall_season_data.head()

****************PARAMETERS*********************
future_steps = 10
STORAGE_FOLDER = 'State_predictions/'
if not os.path.exists(STORAGE_FOLDER):
    os.makedirs(STORAGE_FOLDER)

parameters_WNN = [[3,6,8,10,12], [4,5,6], [1], [300], [20], [future_steps]]
#parameters_WNN = [[12], [4], [1], [50], [20], [future_steps]]
# seasonal_period, hidden_nodes, epochs, batch_size, future_steps
parameters_WAANN = [[12], [3,4,5,6], [300], [20], [future_steps]]
#parameters_WAANN = [[12], [3], [50], [20], [future_steps]]

******************RUN,COMPARE AND SAVE OUTPUT******************
for district in districts_of_interest:
    temp_data = rainfall_season_data[['Date', 'Year', 'Month', district]]
    for month in months_of_interest:
        df = temp_data[temp_data.Month == month]
        df.index = range(len(df))
        df = df[['Date', district]]
        dates = df.Date
        rainfall_data = pd.DataFrame({'Precipitation': df[district][:-1*future_steps]})
        rainfall_data.index = dates[:-1*future_steps]
        test_rainfall_data = pd.DataFrame({'Precipitation': df[district][-1*future_steps:]})
        test_rainfall_data.index = dates[-1*future_steps:]
        scaler = MinMaxScaler(feature_range=(0, 1))
        scaler.fit(rainfall_data)
        output_folder = STORAGE_FOLDER + district + '_' + month + '/'
```

```
if not os.path.exists(output_folder):
    os.makedirs(output_folder)
ANN = reload(ANN)
        RMSE_info = ANN.compare_ANN_methods(rainfall_data,
    test_rainfall_data, scaler,
        parameters_WNN,parameters_WNN, parameters_WAANN,
    parameters_LSTM, future_steps, output_folder)
ANN = reload(ANN)
ANN.save_RMSE_info(output_folder, RMSE_info)
```

Since we used 300 epochs but by default we need to use 1000epochs for Wavelet Neural Network for better accuracy, due to lack of Memory

```
    for district in districts_of_interest:
temp_data = rainfall_season_data[['Date', 'Year', 'Month', district]]
for month in months_of_interest:
    df = temp_data[temp_data.Month == month]
    df.index = range(len(df))
    df = df[['Date', district]]
    dates = df.Date
    rainfall_data = pd.DataFrame({'Precipitation': df[district][:-1*future_steps]})
    rainfall_data.index = dates[:-1*future_steps]
    test_rainfall_data = pd.DataFrame({'Precipitation': df[district][-1*future_steps:]})
    test_rainfall_data.index = dates[-1*future_steps:]

    scaler = MinMaxScaler(feature_range=(0, 1))
    scaler.fit(rainfall_data)

    output_folder = STORAGE_FOLDER + district + '_' + month + '/'
    print(output_folder)
```

```
if not os.path.exists(output_folder):
    os.makedirs(output_folder)

ARIMA = reload(ARIMA)
forecasted_values = ARIMA.ARIMA(rainfall_data, output_folder, future_steps)

errors = test_rainfall_data.Precipitation - forecasted_values
        ARIMA_actual_forecasted = pd.DataFrame({'Actual':
    test_rainfall_data.Precipitation,
                                'Forecasted': forecasted_values,
                                'Errors': errors})
        ARIMA_actual_forecasted.to_csv(output_folder + '/' +
    'ARIMA_actual_and_forecasted.csv')

plt.figure(figsize=(10,5))
        plt.plot(ARIMA_actual_forecasted.drop(columns=['Actual', 'Forecasted']),
    color='blue', label='Error: Actual - Forecasted')
plt.xlabel('Year')
plt.ylabel('Error')
plt.legend(loc='best')
plt.title('ARIMA - Error: Actual - Forecasted')
plt.savefig(output_folder + 'ARIMA_error_plot'  + '.png')

y_true = test_rainfall_data.ix[:int(future_steps)]
plt.figure(figsize=(10,5))
plt.plot(y_true, color='green', label='Actual values')
plt.plot(forecasted_values, color='red', label='Forecasted values')
plt.xlabel('Year')
plt.ylabel('Monthly mean Precipitation')
plt.legend(loc='best')
plt.title('ARIMA - Comaprison: Actual vs Forecasted')
```

```
plt.savefig(output_folder + 'ARIMA_best_forecast'  + '.png')

mse, mae, mape, rmse = calculate_performance(y_true, forecasted_values)
RMSE_score = pd.read_csv(output_folder + 'RMSE_score.csv')
best_method = RMSE_score.iloc[RMSE_score.RMSE.argmin]['Unnamed: 0']
        RMSE_score = RMSE_score.append({'Unnamed: 0': 'ARIMA', 'RMSE':
   str(rmse)}, ignore_index=True)
RMSE_score.RMSE = RMSE_score.RMSE.astype('float')

current_best_method = RMSE_score.iloc[RMSE_score.RMSE.argmin]
   ['Unnamed: 0']

RMSE_score = RMSE_score.set_index('Unnamed: 0')
RMSE_score.to_csv(output_folder + 'RMSE_score.csv')

axis = RMSE_score.plot(kind='bar', figsize=(10,5), rot=0, title='RMSE scores')
axis.set_xlabel('Method name')
for p in axis.patches:
   axis.annotate(np.round(p.get_height(),decimals=2),
           (p.get_x()+p.get_width()/2., p.get_height()),
           ha='center', va='center', xytext=(0, 10),
           textcoords='offset points', fontsize=14, color='black')

fig = axis.get_figure()
fig.savefig(output_folder + 'RMSE.png')

if current_best_method == 'ARIMA':
   y_true = test_rainfall_data.ix[:int(future_steps)]
   plt.figure(figsize=(10,5))
   plt.plot(y_true, color='green', label='Actual values')
   plt.plot(forecasted_values, color='red', label='Forecasted values')
```

```
plt.xlabel('Year')
plt.ylabel('Monthly mean Precipitation')
plt.legend(loc='best')
plt.title('Best of all: ARIMA - Comaprison: Actual vs Forecasted')
plt.savefig(output_folder + 'BEST_FORECAST_ARIMA'  + '.png')
os.remove(output_folder + 'BEST_FORECAST_' + best_method + '.png')
```

5.RESULTS

Out[4]:

	Date	Year	Month	Ahmadnagar	Akola	Amravati	Aurangabad	Bhandara	Bid	Buldana	...	Nashik	Osmanabad	Parbhani	Pune	Sangli	Satara
0	1901-01-01	1901	Jan	2.510	34.202	35.651	10.922	23.397	16.647	31.455	...	5.063	15.500	33.207	0.922	0.138	0.197
1	1901-02-01	1901	Feb	11.489	1.099	6.822	4.362	63.844	1.916	0.823	...	1.609	2.784	4.997	7.195	0.537	0.525
2	1901-03-01	1901	Mar	11.325	30.002	36.103	25.161	33.563	27.287	28.448	...	11.196	11.333	31.625	5.105	13.090	9.566
3	1901-04-01	1901	Apr	33.931	10.248	10.636	12.714	61.560	33.211	13.902	...	7.838	34.814	41.941	35.949	50.077	30.110
4	1901-05-01	1901	May	30.401	2.891	4.173	34.244	13.665	59.027	9.397	...	7.475	52.792	31.794	36.650	78.994	65.226

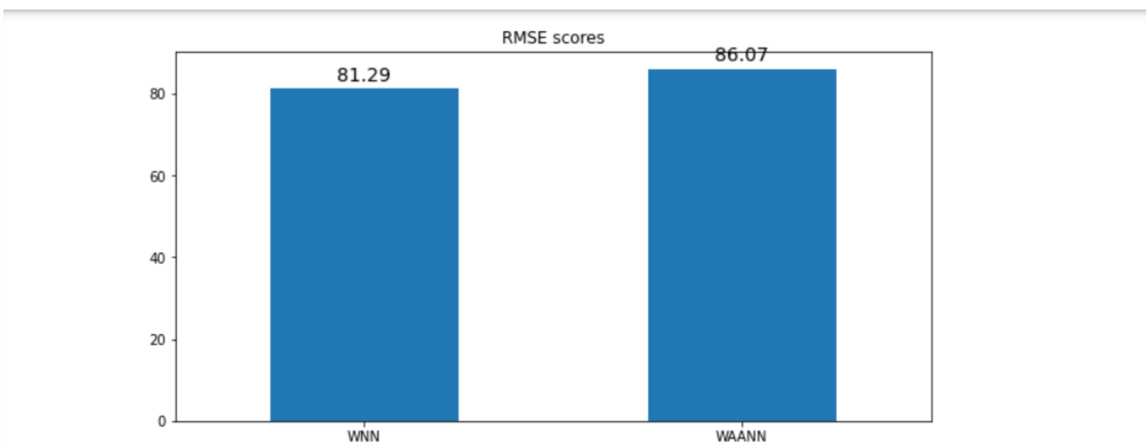
5 rows × 32 columns

Screenshot 5.1: Preprocess Data1

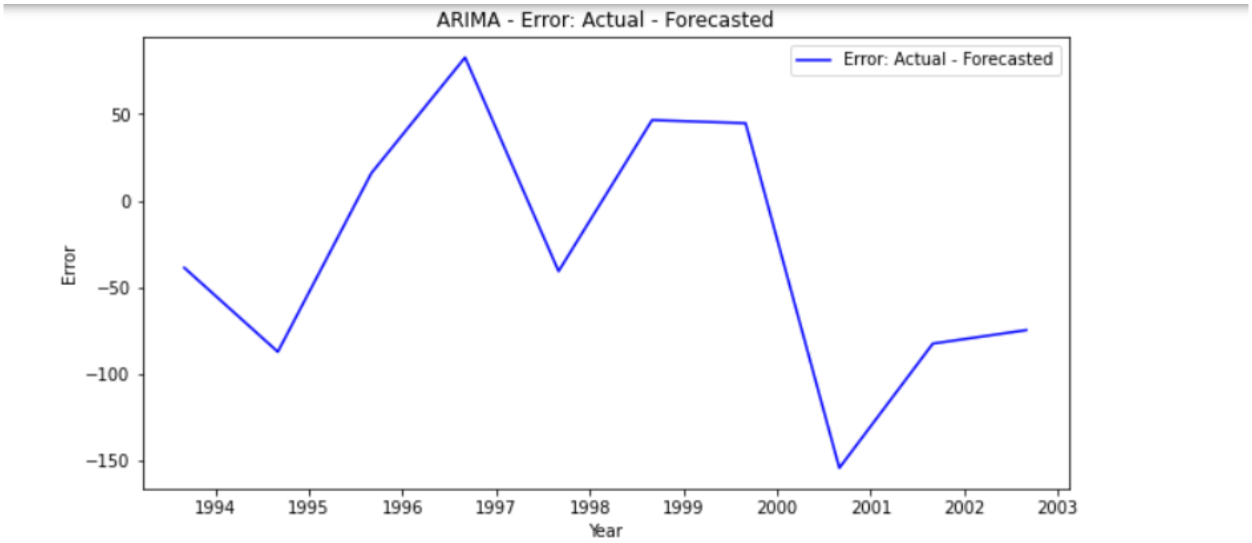
Out[8]:

	Date	Year	Month	Kolhapur	Latur
5	1901-06-01	1901	Jun	554.047	188.878
6	1901-07-01	1901	Jul	496.636	175.092
7	1901-08-01	1901	Aug	507.657	138.084
8	1901-09-01	1901	Sep	221.539	102.949
17	1902-06-01	1902	Jun	631.349	46.720

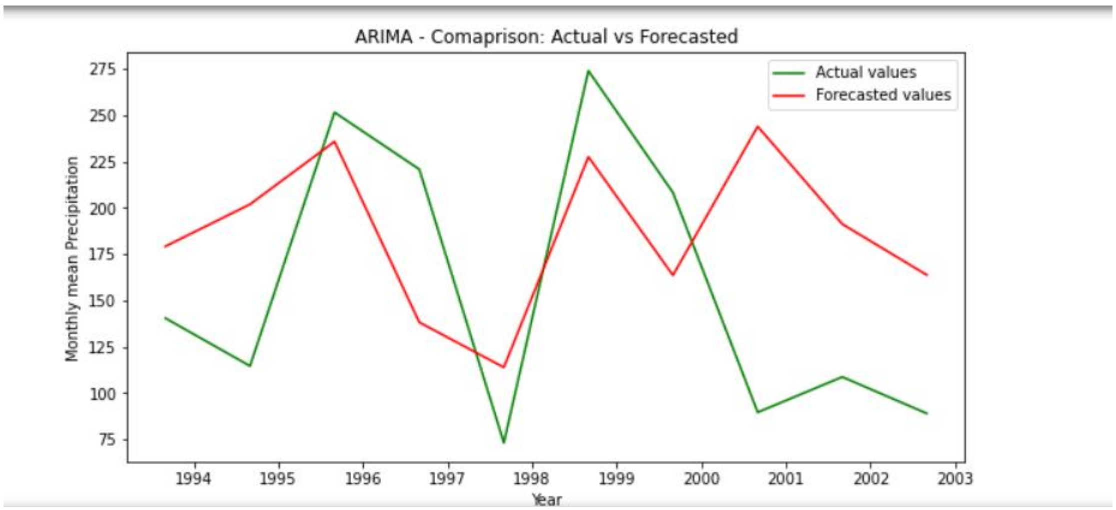
Screenshot 5.2: Preprocess Data2



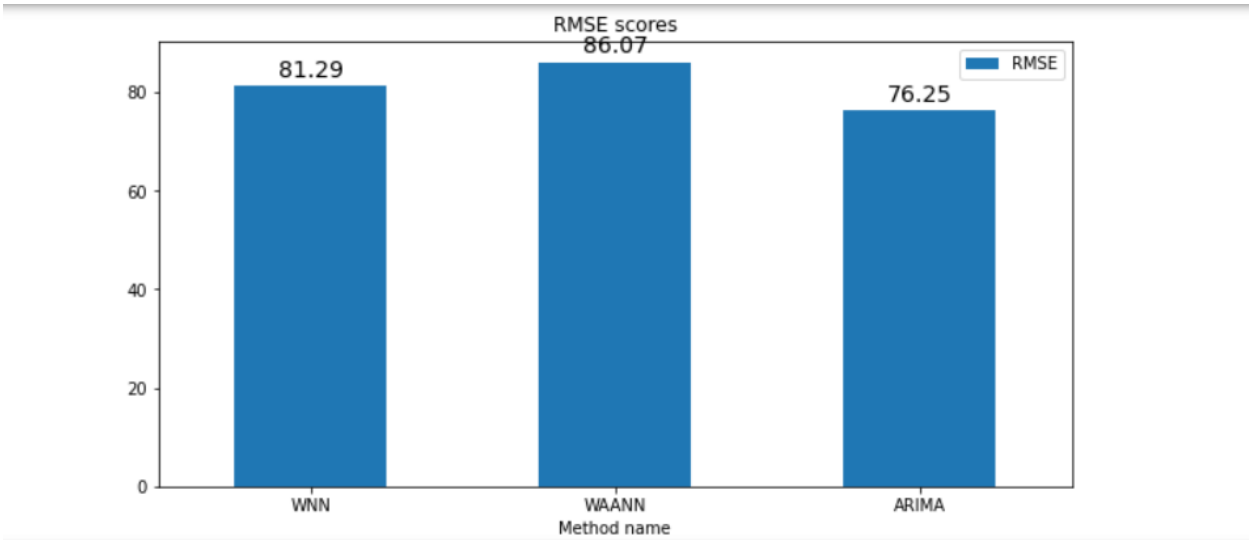
Screenshot 5.3: Comparison Bar graph between WNN and WAANN



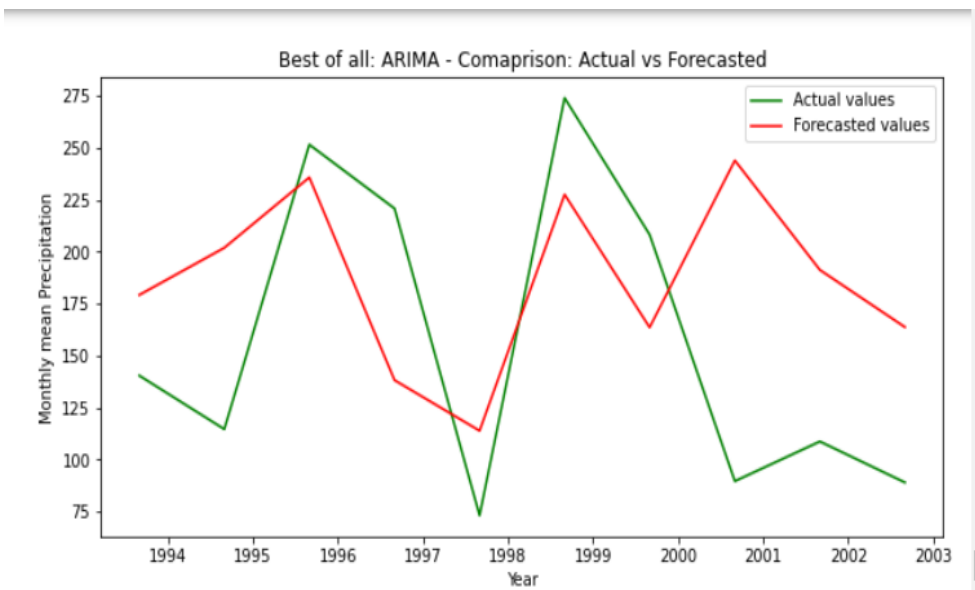
Screenshot 5.4: ARIMA error Actual values



Screenshot 5.5: ARIMA comparison between Actual vs Forecasted values



Screenshot 5.6: Bar graph comparison of RMSE scores between WNN ,WAANN ,ARIMA



Screenshot 5.7: Best of all ARIMA comparison

6.TESTING

6.TESTING

INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

TYPES OF TESTING

UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .It is done after the completion of an individual unit before integration. This is a structural testing that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application and/or system configuration. Unit testsensure that each unique path of a business process performs accurately to the documented specifications and

INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases.

TEST CASES

CLASSIFICATION

Test case ID	Test case name	Purpose	Input	Output
1	Data Preparation	To verify that the input data.	Historical rainfall data.	Preprocessed data with the appropriate features
2	Model Training	To ensure that the ANN model is trained properly.	Training dataset Historical rainfall data.	Trained ANN model with optimized weights.
3	Model Evaluation	To assess the accuracy of the ANN model.	Testing data set with rainfall data.	Model evaluation metrics such as RMSE.
4	Hyper parameter Tuning	To optimize Hyper Parameters of ANN	Various Hyper Parameter Settings To be tested	Best performing set of Hyper parameters based on validation data.
5	Real time prediction	To verify the ANN model to make real time rainfall prediction	Current weather data	Rainfall Prediction for a specific time horizon.
6	Extreme Weather Events	To test the model ability to predict heavy rainfall	Historical data with instances	Verify that the model can accurately predict
7	Data Anomalies	To ensure the model handles data anomalies	Test data with missing values outliers	It provides responsible predictions
8	Model robustness	To evaluate the robustness against input data	Test model data from different geographical regions	The predictions generalize well to different scenarios
9	Interpretability	To assess interpretability and explain ability	Trained ANN model	Determined features for predictions
10	Deployment	To verify deployment of ANN model	Deployed model integrated into a real time	The model provides accurate and timely rainfall predictions

7.CONCLUSION

CONCLUSION & FUTURESCOPE

CONCLUSION

Rainfall prediction using neural network analysis is a powerful tool that offers numerous advantages over traditional methods. By leveraging the power of machine learning, we can make more accurate predictions and better prepare for the impacts of extreme weather events. The model was able to provide better results than other algorithms in the literature and can potentially be used to provide accurate predictions for planning in agriculture and other industries.

FUTURE SCOPE

Improved accuracy and precision : Advances in deep learning can enhance rainfall prediction accuracy.

High-Resolution Spatial Prediction: ANNs enable localized forecasts ,vital for flood prediction and urban planning.

Climate Change Adaptation: ANNs model climate change's impact on rainfall patterns.

User-friendly Interfaces: Accessible software for diverse users.

Customized Applications: Tailored ANN model for specific uses.

8.BIBLIOGRAPHY

8. BIBLIOGRAPHY

REFERENCES

1. Rainfall dataset5of our country is5taken from kaggle website https://www.kaggle.com/rajanand/rainfall-in-india
2. Mosavi, A., Ozturk, P., & Chau, K. W. (2018). Flood prediction using5machine learning5models: Literature review. Water (Switzerland), 10(11). https://doi.org/10.3390/w10111536
3. Janani, B; Sebastian, P. (2014). Analysis on the weather forecasting5and techniques. International Journal5of Advanced Research in Computer Engineering & Technology, 3(1), 59–61.
4. Chaudhari, M. S., & Choudhari, N. K. (2017). Open Access Study of5Various Rainfall Estimation & Prediction Techniques5Using Data Mining. American Journal of Engineering Research (AJER), 7, 137– 139. http://www.ajer.org/papers/v6(07)/Q060713713 9.pdf
5. Shi, X., Chen, Z., & Wang, H. (2015). Convolutional LSTM Network. Nips, 2–3. https://doi.org/[]
6. Wahyuni, E. G.Fauzan, L. M.F.Abriyani, F.Muchlis, N. F., & Ulfa, M. (2018). Rainfall prediction5with backpropagation method.5Journal of Physics: ConferenceSeries,983(1).
7. M.J.C., Hu, Application of ADALINE system to weather forecasting, Technical Report, Stanford Electron, 1964
8. Kalogirou, S. A., Neocleous, C., Constantinos, C. N., Michaelides, S. C. & Schizas, C. N.,"A time series construction of precipitation records using artificial neural networks. In: Proceedings of EUFIT '97 Conference, 8–11 September, Aachen, Germany. pp 2409–2413 1997.
9. Lee, S., Cho, S.& Wong, P.M.,"Rainfall prediction using artificial neural network.",J. Geog. Inf. Decision Anal. 2, 233–242 1998.
10. Wong, K. W., Wong, P. M., Gedeon, T. D. & Fung, C. C. , "Rainfall Prediction Using Neural Fuzzy Technique." 1999.

11. E.Toth, A.Brath, A.Montanari," Comparison of short-term rainfall prediction models for real-time flood forecasting", Journal of Hydrology 239 (2000) 132–147
12. Koizumi, K.: "An objective method to modify numerical model forecasts with newly given weather data using an artificial neural network", Weather Forecast., 14, 109–118, 1999.
13. N. Q. Hung, M. S. Babel, S. Weesakul, and N. K. Tripathi "An Artificial Neural network Model for rainfall Forercastingin Bangkok,Thailand", Hydrol. Earth Syst. Sci., 13, 1413–1425, 2009
14. 0011Kyaw Kyaw Htike and Othman O. Khalifa, "Research paper on ANN model using focused time delay learning", International Conference on Computer and Communication Engineering (ICCCE 2010), 11-13 May 2010, Kuala Lumpur.
15. Ajith Abraham, Dan Steinberg and Ninan Sajeeth Philip,"Rainfall Forecasting Using Soft Computing Models and Multivariate Adaptive Regression Splines", 2001.
16. Paras, Sanjay Mathur, Avinash Kumar, and Mahesh Chandra, "A feature based on weather prediction using ANN" World Academy of Science, Engineering and Technology 34 2007.

GITHUB LINK

https://github.com/Suraj5566/Rainfall-prediction-using-neural-network-analysis