Rainfall Forecasting Using Time Series Analysis

A Project Report

Submitted in partial fulfillment of the requirement for the award of the degree of

Bachelor of Technology
in
Electronics and Communication Engineering

by

THARUN V. P

14BEC0664

Under the guidance of

Prof. S. Renuga Devi

School of Electronics Engineering

Vellore Institute of Technology, Vellore-632014

VIT

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

April 2018

DECLARATION

I hereby declare that the project work entitled "**Rainfall Forecasting Using Time Series Analysis**" submitted by me, for the award of the degree of *Bachelor of Technology in Electronics and Communication Engineering* to Vellore Institute of Technology is a record of bonafide work carried out by me under the supervision of **Prof. S. Renuga Devi.**

I further declare that the work reported in this report has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Signature of the Candidate

Place : Vellore

Date : 22/4/18

CERTIFICATE

This is to certify that the project work entitled "Rainfall Forecasting Using Time Series Analysis" submitted by **Tharun V.P**, School of Electronics Engineering, Vellore Institute of Technology, for the award of the degree of *Bachelor of Technology in Electronics and Communication Engineering*, is a record of bonafide work carried out by him under my supervision, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The report fulfills the requirements and regulations of the institute and in my opinion meets the necessary standards for submission.

Place : Vellore
Date : 22.04.2018

Signature of the Guide

The project work is satisfactory / unsatisfactory

Internal Examiner **External Examiner**

Approved by

Head of the Department
Department of Communication Engineering
School of Electronics Engineering

ACKNOWLEDGEMENT

I would like to express my gratitude towards beloved **Chancellor Dr. G. Viswanathan**, for providing necessary facilities to carry out and finish the project successfully. I am grateful to **Vice Presidents: Mr. G V Selvam, Dr. Sekar Viswanathan and Mr. Sankar Viswanathan** for their support and encouragement. I owe my sincere thanks to the **Vice Chancellor Dr. Anand A. Samuel** and **Dean SENSE, Prof. Elizabeth Rufus** for their continuous support.

I take this opportunity to express my profound gratitude and deep regards to my guide **Prof. S. Renuga Devi** for her enthusiasm and support. I thank her for her able guidance, untiring attention, and constant encouragement throughout the project work.

I am deeply indebted to all the faculty who with their expertise, experience and knowledge guided me in my project and encouraged me to work upon the weak links. A special mention is to be made for Prof. Sankar Ganesh. In addition, I would like to acknowledge the Project Coordinators for their continuous updates regarding the procedures for the reviews and guidelines for presentations and thesis preparation.

I also acknowledge my project partner Ramya Prakash and all my friends who gave me support and timely suggestions, imperative for my work.

Above all and most importantly, I thank God almighty and my parents for their blessings throughout the implementation of this project.

Tharun V.P

Executive Summary

Indian Meteorological Department (IMD) has progressively expanded its infrastructure for meteorological observations, communications, forecasting and weather services and it has concurrently contributed to scientific growth. Meteorological data mining is a form of data mining concerned with finding hidden patterns inside largely available meteorological data, so that the information retrieved can be transformed into usable knowledge. Weather is one of the meteorological data that is rich by important knowledge.

Rainfall data follow a time series pattern which can be daily, monthly or yearly, defined as the outcome of interaction between several complex atmospheric processes. The benefits of the rainfall if properly cultivated could enhance the production rate and lead to improved management of disasters. One way to make it possible is to devise methods to interpret the general yearly or monthly rainfall pattern much in advance. The above can be achieved with an efficiently developed prediction model that could forecast the rainfall pattern either by studying the previously available rainfall data or by studying the atmospheric factors on which the rainfall intensity values are closely related to or both. A large uncertainty involved in determining the contribution of these atmospheric factors is one of the biggest challenges that has to be faced while developing rainfall prediction models. The rainfall forecasts have become an essential information to support crops, for water and flood management in the modern-day world. It also has a vital role to play in disaster management on occurrences of landslides caused due to heavy downpours. Preventive methods can be adopted by Governmental organizations with respect to the predictions made by the forecasting models to mitigate the damages that could possibly be caused by an unexpected downpour.

The research and analysis undertaken for this work is based on the daily rainfall data of Coonoor in Nilgiris district of Tamil Nadu. The various prediction models developed throughout this project have been trained with the same daily rainfall data obtained from the Indian Meteorological Department and the Public Works Department of the Nilgiris district. The dataset includes daily rainfall data spanning over 9 years from 2004 to 2013 along with the details of the atmospheric factors like Temperature, Humidity, Wind speed and Wind direction which plays a significant role in determining the rainfall pattern in an area. The rainfall intensity details of three nearby regions of Coonoor like the Coonoor railway, Runnymedu and

the Coonoor center were further added to the dataset in order train a more efficient and robust prediction model.

Considering the very fact that the target dataset includes parameters with widely varying ranges and units, a thorough analysis is done to understand the extend of correlation between these parameters and the rainfall intensity values. The dataset is further pre-processed with at most care with highly efficient data processing techniques to ensure that the information contained within each parameter is secured. In the further stage, the pre-processed data was passed on to advanced prediction models for making coherent forecasts. Finally, the models are compared based on their performance in terms of prediction accuracy which is assessed using the Root Mean Square Error (RMSE) values obtained from the comparison of the predicted results with the actual rainfall intensity values in the dataset.

The prediction models worked upon in this project are widely classified under two categories which are Univariate and Multivariate Time series analysis. The Univariate analysis includes Statistical modelling techniques like Baseline forecast, Seasonal persistence, Auto regression and ARIMA along with Recurrent Neural Network based models like Non-Auto Regressive(NAR) neural network and Long Short Term Memory(LSTM) Recurrent Neural Network. Similarly, Multivariate analysis includes Statistical regression techniques like Support Vector Regression, Machine Learning based techniques like Decision tree, Random Forest and Xgboost along with Recurrent Neural Network based models like Non-Auto Regressive with Exogenous inputs(NARX) and multivariate LSTM RNN.

Upon the selection of the best algorithm from each of the categories mentioned above, Sensitivity analysis is performed separately on each of them to obtain the best hyper-parameters using Grid search algorithm and Trial and Error technique. For the reason that there are a large number of input parameters involved in the training process, a lot of time is taken for the completion. Such problems associated with the large number of input parameters is solved using advanced Feature extraction techniques like Principle Component Analysis(PCA) and Kernal-PCA(KPCA).These techniques helps in the dimensionality reduction of the input parameters with minimal information loss. Further in order to address the problems associated with high Variance and high Bias of the models, techniques like K-fold cross validation and Dropout regularisation were implemented during the model training stage.

Amongst the models that were implemented and tested based on the prediction accuracy, Random forest outperforms all other machine learning and statistical modelling techniques. Xgboost being a highly advanced boosted tree algorithm takes extremely less time for execution. It also gives commendable performance in comparison with most other algorithms. Among the RNN models, NARX shows hopeful results while LSTM comes out to be disappointing considering the fact that it's an highly advanced Neural Network. Artificial Neural Network implemented using multivariate predictors provides the most stable and accurate results in comparison with all the other models.

List of Figures

List of Tables

List of Terms and Abbreviations

SVM	Support Vector Machine
DT	Decision Tree
KNN	K-Nearest Neigbor
XGBoost	Extreme Gradient Boost
NAR	Non-Linear Auto Regressive Neural Network
ANN	Artificial Neural Network
MSE	Mean Square Error
NARX	Non-Linear Auto Regressive with Exogenous Inputs
RNN	Recurrent Neural Network
RMSE	Root Mean Square
LSTM	Long Short Term Memory
PCA	Principal Component Analysis
k-PCA	Kernel Principal Component Analysis
ARIMA	Auto Regressive Integrated Moving Average

CHAPTER 1

INTRODUCTION

Rainfall is a natural phenomenon defined as the outcome of interaction between several complex atmospheric processes. A large uncertainty involved in determining the contribution of the atmospheric processes is one of the biggest challenges to face in developing rainfall forecasting models. Forecasting of rainfall is very difficult to model since the atmospheric processes involved follow a rather complex nonlinear pattern. Rainfall prediction is a major concern for meteorological department as it is closely associated with the economy and sustenance of human life. The temperature, relative humidity, wind speed, wind direction, cloud coverage etc. are some of the factors that critically affect the occurrence of rainfall. The rainfall forecast is essential information to support crop, water and flood management. It also has a vital role to play in disaster management on occurrence of landslides caused due to heavy downpour. The occurrence of landslides is normally unpredicted and causes a lot of death and damages to infrastructure such as buildings, highways, roads and bridges. Rainfall forecast models can help in saving the lives and properties of the people, which indirectly support the economy of the country. To provide a good and accurate prediction, forecasting models have been developed and implemented using statistical modelling and regression techniques.

This work is a comparative study of primitive techniques such as Baseline / Naïve Forecast, Seasonal Persistence Algorithm, AutoRegression and ARIMA, time series analysis networks such as NAR and NARX, regression techniques such as Support Vector Regression, Random Forest, Decision Tree and XGBoost; Artificial Neural Network and Recurrent Neural Network on the basis of accuracy of prediction. Primitive rainfall forecast models fails to provide good accuracy for rainfall prediction as compared to regression techniques due to dynamic nature of atmospheric composition. Later, the rainfall forecasting models were developed using NAR, NARX, Artificial Neural Networks and Recurrent Neural Networks.

The work includes various data pre-processing techniques such as Imputation, Categorical Encoding, Feature Scaling and splitting of dataset into training and testing data. The objective is to predict rainfall intensity of Coonoor in Nilgiris district, Tamil Nadu using various techniques. The Nilgiris is prone to moderate to high risk of landslides due to intense rainfall as a triggering cause. Hence, the rainfall forecast models reduces most of the effects of intense rainfall wherein people are informed of the possibility prior to the occurrence of the same.

The use of Artificial Neural Network provides better solutions when compared to the primitive, statistical and regression techniques in terms of the evaluation parameters such as accuracy of prediction. The analysis was carried out on Python and MATLAB platforms.

1.1 MOTIVATION FOR RESEARCH

A number of incidents which took place in the past witnesses a huge loss of property, lives, infrastructure and economy of the country. There are a large number of meteorological and public works department who predict the rainfall based on the geological surveys. Traditional methods which uses geological surveys are time consuming. The aspects that provided motivation to develop forecast models were:

- Disaster management such as landslides due to intense rainfall.
- Loss in agricultural economy due to the heavy rainfall which could have been avoided if prior information of rainfall was provided.
- The necessity to analyse the risk to improvise the flood and water management.
- Establishing statistical, time series networks, regression, artificial and recurrent neural network tools for rainfall forecasting.

1.2 PROBLEM STATEMENT

Nilgiris is one of the major tourist destinations of India. Nilgiris attract lakhs of tourists every year. Nilgiris has to consider the use of unsafe and unstable slopes to construction sites in order to fill growing urbanism needs. Nilgiris being at an elevation of 2,637 m coupled with the fact that it receives heavy rainfall of over 1700 mm every year makes it highly susceptible to rainfall induced landslides every year. Coonoor is a Taluk and a municipality in Nilgiris district. It is located at an altitude of 1850 m above sea level and is the second largest hill station in the Nilgiri hills after Ooty. There are 14 rain gauge stations in and around Coonoor, which keep a record of the amount of rainfall received. This phenomenon causes increased interest in the forecasting of rainfall. Traditional methods require geologists to do surveys which is time consuming. Therefore, faster and efficient methods have to be developed and implemented in order to forecast the rainfall.

1.3 OBJECTIVES OF THE PROJECT

The main aim of this project is to develop and implement efficient models to forecast rainfall accurately. The objectives of this project are as follows:

- Pre-process the data using Imputation, Categorical encoding, Feature scaling and splitting the dataset.
- To implement Baseline / Naïve Forecast, Seasonal Persistence, Auto Regression, Auto Regressive Integrated Moving Average (ARIMA), NAR, NARX, ANN and Recurrent Neural Network to forecast the rainfall using time series analysis.
- Comparison of Univariate and Multivariate time series analysis for various forecasting methods to predict the rainfall.
- Comparative study of different regression techniques such as Support Vector Regression, Decision Tree, Random Forest and XGBoost based on R-square, Adjusted R-square and mean square error.
- To determine the best Artificial Neural Network Model which can forecast rainfall with nominal error.
- To implement Feature extraction (Dimensionality reduction) techniques to reduce the computational complexity of neural network models.
- To implement the forecasting model using Recurrent Neural Network with multiple input parameters (time series analysis).
- To compare and analyze various Regression techniques, Artificial and Recurrent neural network to decide the most efficient model based on the evaluation criterion.
- To perform sensitivity analysis and identify the best hyper-parameters for optimum performance of the models.

1.4 SIGNIFICANCE OF THE RESEARCH

Rainfall is instrumental for the survival of living forms across the globe. The rainfall holds even more importance if viewed in the perspective of a country like India where the agriculture sector contributes largely to the economic growth and stability. Agriculture is the predominant occupation in India and most of the parts of the country depend on the rainfall for the agricultural needs. Hence, this research focuses to develop, implement and evaluate the various techniques in order to forecast

rainfall more accurately and in a faster manner. With increased demand for safer hill roads and safer construction sites due to increase in tourist attractions in Coonoor, a better understanding of when disasters could occur is very important. It focuses on all the variables that act as a triggering cause for the intense rainfall such as temperature, humidity, wind speed, wind direction, cloud coverage, etc.

1.5 PROPOSED APPROACH

The idea of the project is to design and evaluate an effective and robust model using different variants of ANN, RNN, regression and statistical modelling for forecasting the daily rainfall for Coonoor in Nilgiris district. The methodology followed for the project is as follows:

- **Step 1→Data Collection**: This stage deals with collection and consolidation of the rainfall data along with other physical parameters.
- **Step 2 → Pre-processing**: This stage scales down the rainfall data collected along with other parameters using imputation, categorical encoding, and feature scaling and splitting of dataset. It scales down the data within 0 and 1 or -1 and 1.
- **Step 3 → Development and Implementation of Forecasting Model**: In this stage, forecasting models are designed as follows:
    1. Univariate TSA: Forecast rainfall using only one input parameter i.e. rainfall
        a. Implementation of Persistence / Naïve forecast, Seasonal Persistence, Auto Regression, ARIMA, NAR, RNN.
    2. Multivariate TSA: Forecast rainfall using multiple input parameters e.g. temperature, humidity, cloud coverage, wind speed, etc.
        a. Implementation of Regression, ANN, RNN and NARX for Multivariate inputs.
- **Step 4→ Performance Evaluation**: This stage evaluates the models developed in Step 3 on different performance measures and performs a comparative analysis.

CHAPTER 2

BACKGROUND THEORY

2.1 RAINFALL FORECASTING-THE NEED OF THE HOUR

The rainfall is not only essential for the existence and equilibrium of the nature but also is vital for economic stability of the country. Since the agricultural sector, the major economic pillar of the country, depends largely on the rainfall, the rainfall forecasting models hold an unprecedented significance. Rainfall forecasting models provide an insight about the measure and probability of the occurrence of precipitation, which enlightens the system to formulate its policies in light of the forecasted information. The significance of the rainfall forecasting models are as follows:

- Knowing an advance estimate of the precipitation occurring in future, the authorities could make suitable arrangements, or policies to map the required and available water demand for crop plantation.
- With the advance information on rainfall occurrence, the system could compute the probability of floods occurrence and seek proper methods to prevent them,
- The system can even predict landslides based on forecasted rainfall values and search proper ways to fulfil the requirements of deficit areas.
- By employing proper means, the system can save the lives and properties of people indirectly adding to the economic benefit of the country.

2.2 STUDY AREAS- NILGIRIS AND COONOOR

2.2.1 NILGIRIS

Nilgiris is a district in Tamil Nadu, India and has an area of 2,452.50 km^2. The district is a hilly region, situated at an elevation of 2000 to 2600 meters above the mean sea level. Almost the entire district lies in the Western Ghats. Its latitudinal and longitudinal dimensions being 130 km. The Nilgiris district is bounded by Mysore district of Karnataka and Wayanad district of Kerala in the North, Malappuram and Palakkad districts of Kerala in the West, Coimbatore district of Tamil Nadu in the South and Erode district of Tamil Nadu and Chamarajnagar district of Karnataka in the East.

The district usually receives rain during South West Monsoon and North East Monsoon. There are 16 rainfall-registering stations in the district. The average annual rainfall of the district is 1335 mm.

2.2.2 COONOOR

Coonoor is a Taluk and a municipality in the Nilgiris district. It is known for its production of Nilgiri Tea. Coonoor is located at an altitude of 1850 m above sea level, and is the second largest hill station in the Nilgiri hills after Ooty. It is an ideal base for a number of trekking expeditions leading into the Nilgiris. Coonoor is located at 11.35°N 76.82°E. There are around 14 rain gauge stations near Coonoor, which keep a record of received precipitation amount.

2.3 TIME SERIES ANALYSIS

Time Series problems are different to traditional prediction problems. The addition of time adds an order to observations that both must be preserved and can provide additional information for learning algorithms. It is an important area of machine learning that is often neglected as the involvement of time component makes time series problems more difficult to handle. Time series analysis involves developing models that best capture or describe an observed time series in order to understand the underlying causes. Time series analysis provides the decomposition of a time series into 4 constituent parts:

1. **Level**: The baseline value for the series if it were a straight line.
2. **Trend**: The optional and often linear increasing or decreasing behaviour of the series over time.
3. **Seasonality**: The optional repeating patterns or cycles of behaviour over time.
4. **Noise**: The optional variability in the observations that cannot be explained by the model.

The main features of many time series are trends and seasonal variations and most of time series observations close together in time tend to be correlated.

The project includes univariate and Multivariate time series analysis. The univariate time series analysis, as the name suggests, includes only one parameter for forecasting the rainfall whereas Multivariate time series analysis includes many parameters for forecasting. In general, the following methods are used in this project to forecast the rainfall in Coonoor in Nilgiris District, Tamil Nadu

1. Persistence algorithm / Naïve Forecast
2. Seasonal Persistence Algorithm

3. AutoRegression time series analysis
4. AutoRegressive Integrated Moving Average (ARIMA)
5. Non-Linear AutoRegressive Neural Network (NAR)
6. Non-Linear Autoaregressive with Exogenous inputs (NARX)
7. Regression Techniques (SVR, DT, RF and XGBoost)
8. Artificial Neural Network
9. Recurrent Neural Network

2.3.1 STATISTICAL MODELLING

2.3.1.1 Baseline / Naïve Forecast

Baseline algorithm or the Naïve forecast is the most common baseline method for supervised machine learning. This algorithm uses the value of the previous time step (t-1) to predict the outcome expected at the next time step (t+1). For example, table 1 shows a sample of 5 rows. The t-1 column is the input variable and the t+1 is the output variable starting with index 0. Persistence problem can be defined as a function that returns the value provided as input. For example, if the t-1 value of 266.0 was given and returned as prediction whereas the actual value happens to be 150. Once predictions are made for all time steps they are compared to actual values and MSE is calculated. This forecasting technique is so easy to understand and to implement and evaluate because the model is 1-step behind reality. Figure 1 shows a persistence model prediction which shows a rising trend and month-to month noise in the figure highlighting the limitations of naïve forecast.

Table 1. Sample dataset for Persistence / Naïve forecast

Index	t-1	t+1
0	NaN	266.0
1	266.0	145.9
2	145.9	183.1
3	183.1	119.3
4	119.3	180.3

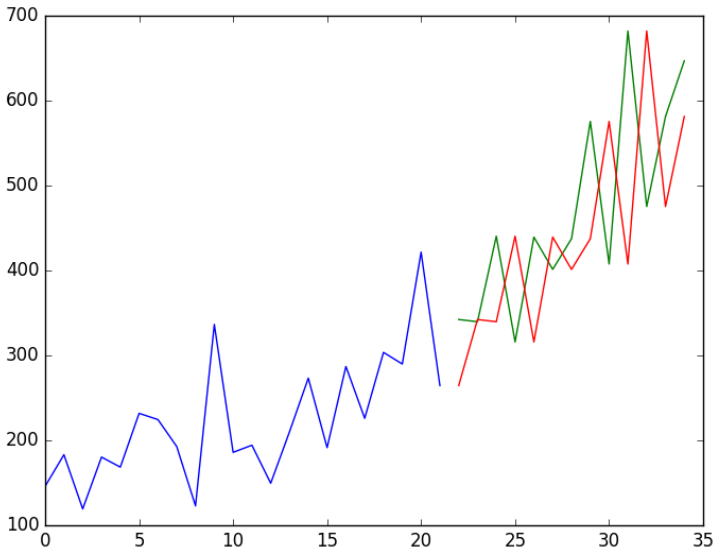


Figure 1. Graph of Persistence Model Algorithm

2.3.1.2 Seasonal Persistence Algorithm

A better forecast on time series data with a seasonal component is to persist the observation for the same time in the previous season is called seasonal persistence. It results in an effective predictive model as it uses the time series data of previous season of the same time. It is a simple model because it uses a simple function of the last few observations at the same time in previous seasonal cycles, for example, the mean of the observations. It uses the concept of sliding window to make forecasts. Within a sliding window, observations at the same in the previous season will be collected and the mean of the observations will be used to forecast the rainfall.

For example, if the data is monthly and the month to be predicted is January, then with the window size of 2 would include observations of last two Januaries average to forecast.

2.3.1.3 Autoregression Time Series Analysis

It is a time series analysis model that uses previous observations as inputs to a regression equation to predict the value at the next time step. It is a simple model that results in accurate forecasts. An autoregression model makes an assumption that the observations at previous time steps are useful to predict the value at the next time step. It works in the following manner:

$$\mathbf{Y=b0 + b1*X1}$$

Where Y is the prediction, b0 and b1 are coefficients found by optimizing the model on training data and X is an input value. If the prediction is to be done for the next time step (t+1) given the observations of last two time steps (t-1 and t-2).

$$\mathbf{X(t+1) = b0 + b1*X(t-1) + b2*X(t-2)}$$

X(t-1) and X(t-2) are lag variables. The stronger the correlation between the output variable and a specific lagged variable, the more weight that autoregression model can put on that variable when modelling.
As the regression model uses data from the same input variable at previous time steps, it is referred to as regression of self-i.e. autoregression.

2.3.1.4 Autoregressive Integrated Moving Average (Arima)

ARIMA is an acronym for AutoRegressive Integrated Moving Average. It is a popular and widely used statistical model for time series forecasting. This model captures the key aspects briefed as below:

AR: AutoRegression- a model that uses the dependent relationship between an observation and any number of lagged observations.

I: Integrated- the use of subtracting an observation from an observation at the previous time step in order to make the time series stationary.

MA: Moving Average- a model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.

The parameters of the ARIMA model are described as follows:

- **p**: the number of lag observations included in the model called the lag order.
- **d**: the number of times that the raw observations are differenced, also called the degree of differencing.
- **q**: the size of the moving average window called the order of moving average.

A linear regression model is constructed including the specified number and type of terms, and the data is prepared by a degree of differencing in order to make it stationary, i.e to remove trend and seasonal structures that negatively affect the regression model. A value of 0 can be used for a

parameter which indicates to not use that element of the model. This way the ARIMA model can be configured to perform the function of an ARMA model and even a simple AR, I or MA model.

2.3.1.5 Non-Linear Autoregressive Neural Network (NAR)

Prediction is a kind of dynamic filtering, in which past values of one or more time series are used to predict future values. Dynamic neural networks, which include tapped delay lines are used for nonlinear filtering and prediction. NAR is the acronym for Nonlinear Auto Regressive which uses the past values of y(t) to predict the next y(t) series. NAR is a type of neural network that can be trained to predict a time series from 'd' past value of the series.

Figure 2 shows that the network has one input, 2 time delay and 10 hidden neurons. In closed loop mode, this input is joined to the output. NAR takes the arguments as shown below:

narnet (feedbackDelays, hiddenSizes, trainFcn)

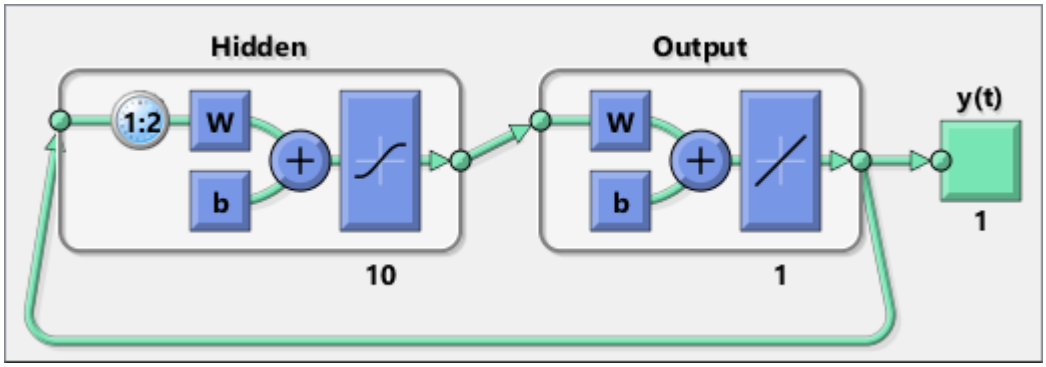


Figure 2. Non-linear AutoRegressive Neural Nerwork Architecture

2.3.1.6 Non-Linear Autoregressive With Exogenous Inputs (NARX)

The NARX is the acronym for Nonlinear AutoRegressive network with Exogenous inputs which is a recurrent dynamic network, with feedback connections enclosing several layers of the network. It is based on the linear ARX model which is commonly used for time series modelling. The defining equation for the NARX model is

$$y(t) = f(y(t-1), y(t-2), \ldots\ldots, y(t-n), u(t-1), u(t-2), \ldots\ldots, u(t-n))$$

where the next value of the dependent output signal y(t) is regresses on previous values of the output signal and previous values of an independent (exogenous) input signal. NARX model can be

implemented using a feedforward neural network to approximate the function f. Figure 3 shows a network where a two-layer feedforward network is used for the approximation.

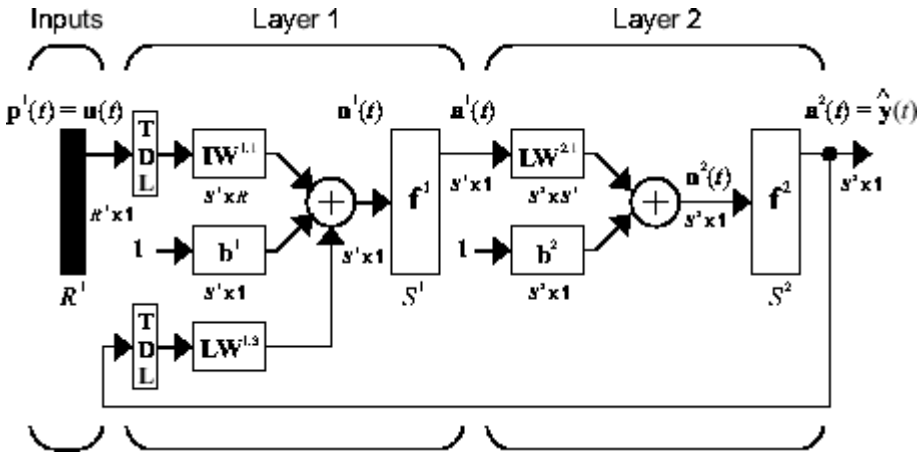


Figure 3. NARX two-layer feedforward Architecture

There are many applications for the NARX network. It can be used as a predictor, to predict the next value of the input signal. It can also be used for modeling nonlinear dynamic systems.
The output of the NARX network to be an estimate of the output of some nonlinear dynamic system. The output is fed back to the input of the feedforward neural network as a part of the NARX architecture. Figure 4 shows a NARX architecture with 10 hidden neurons and two time delays for both input and output.

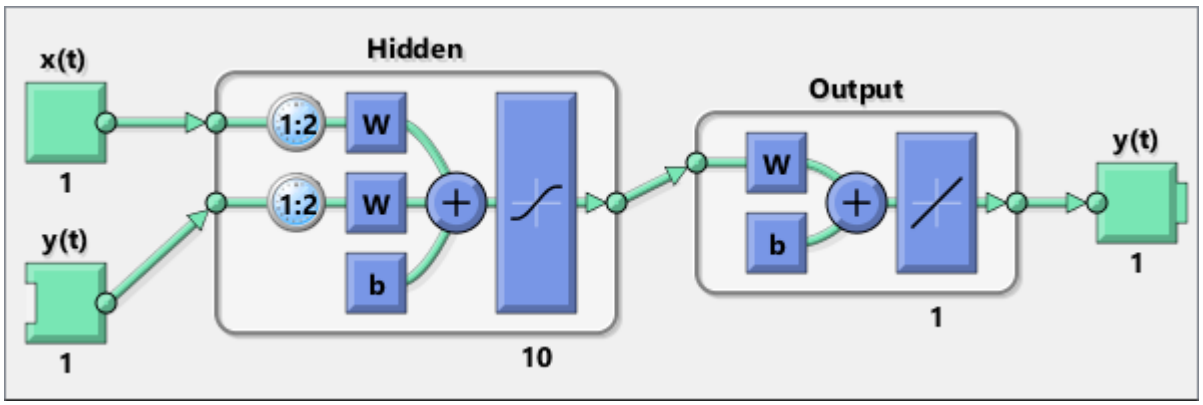


Figure 4. NARX open loop Architecture

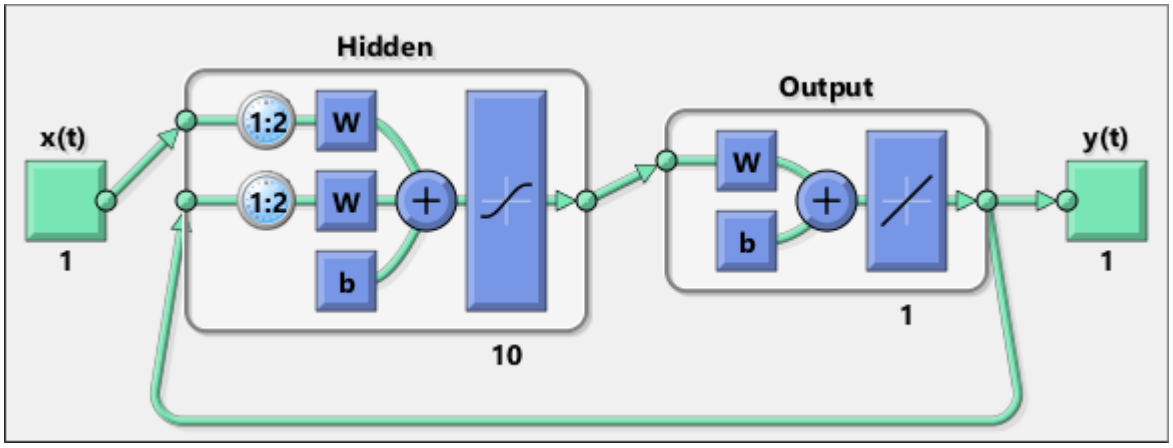


Figure 5. NARX Closed Loop Architecture

All the training is done in open loop including the validation and testing steps. Only after it is trained, it is transformed to closed loop for multistep-ahead prediction. NARX network will work for problems with multiple external input elements and predict series with multiple elements.

2.3.2 REGRESSION TECHNIQUES

Regression analysis is a form of predictive modelling technique which investigates the relationship between a **dependent** (target) and **independent variable (s)** (predictor). This technique is used for forecasting, time series modelling and finding the causal effect relationship between the variables. Regression analysis is an important tool for modelling and analysing data. Here, we fit a curve / line to the data points, in such a manner that the differences between the distances of data points from the curve or line is minimized.

2.3.2.1 Why do we use Regression Techniques?

Regression is concerned with modelling the relationship between variables that is iteratively refined using a measure of error in the predictions made by the model. As mentioned above, regression analysis estimates the relationship between two or more variables. There are multiple benefits of using regression analysis. They are as follows:

1. It indicates the **significant relationships** between dependent variable and independent variable.
2. It indicates the **strength of impact** of multiple independent variables on a dependent variable.

Regression analysis also allows us to compare the effects of variables measured on different scales. These benefits help market researchers / data analysts / data scientists to eliminate and evaluate the best set of variables to be used for building predictive models.

2.3.2.2 Techniques Used in the Research

Supervised learning is where we have input variables (x) and an output variable (Y) and a mapping algorithm is used to relate the input to the output.

$$Y = f(X)$$

The goal is to approximate the mapping function so well that when you have new input data (x) that you can predict the output variables (Y) for that data. Learning stops when the algorithm achieves an acceptable level of performance.

This research deals with the following regression techniques to forecast rainfall:

- Support Vector Regression
- Decision Tree
- Random Forest
- XGBoost

2.3.2.3 Support Vector Regression

Support Vector Regression maintains all the main features that characterize the algorithm (maximal margin). The main idea is to minimize error, individualizing the hyper-plane which maximizes the margin, keeping in mind that part of the error is tolerated. The vectors that define the hyper-plane are the support vectors.

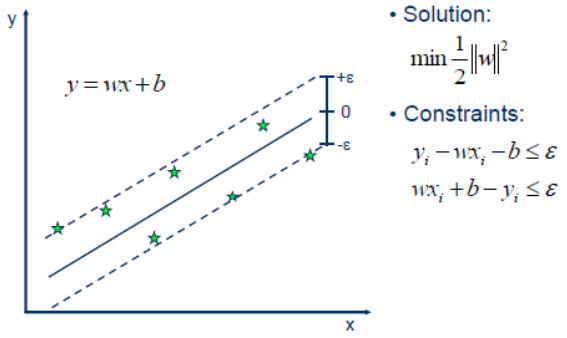


Figure 6. Illustration of Support Vector Machine

ε (epsilon) – margin of tolerance

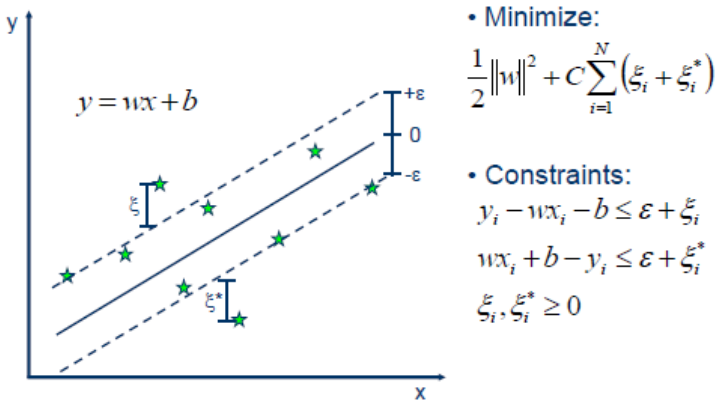


Figure 7. Illustration of SVM with minimum error

- Linear Support Vector Regression

- Non- Linear Support Vector Regression

$$y = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)\cdot\langle\varphi(x_i), \varphi(x)\rangle + b$$

$$y = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)\cdot K(x_i, x) + b$$

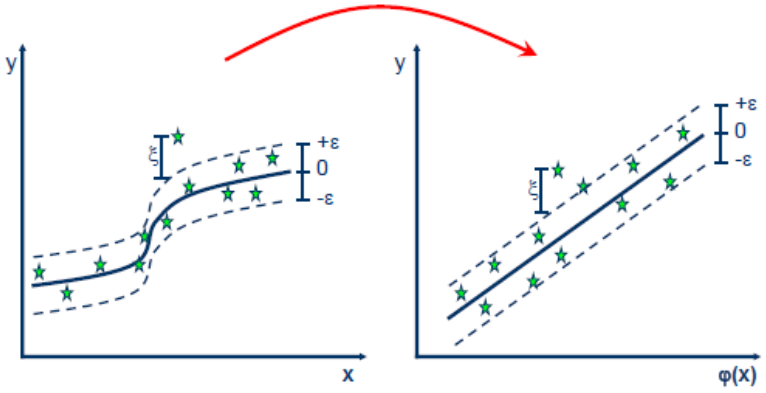


Figure 8. Illustration of Linear and Non-linear SVM

- **Kernel functions:** A kernel is a similarity function. It is a function that is provided to a machine learning algorithm. It takes two inputs and spits out how similar they are. A single kernel function can be used to compute similarity for the non-linear data. One big reason that kernels are considered as opposed to feature vector is that the computational time for kernels are less. Briefly speaking, a kernel is a shortcut that helps us do certain calculation faster which otherwise would involve computations in higher dimensional space.

Polynomial

$$k(\mathrm{x}_i, \mathrm{x}_j) = (\mathrm{x}_i . \mathrm{x}_j)^d$$

Gaussian Radial Basis function

$$k(\mathrm{x}_i, \mathrm{x}_j) = \exp\left(-\frac{\left\|\mathrm{x}_i - \mathrm{x}_j\right\|^2}{2\sigma^2}\right)$$

2.3.2.4 Decision Trees

Decision tree is a supervised learning algorithm which breaks a large dataset into smaller homogenous datasets to make the classification much easy and efficient. The decision tree algorithms can be understood by the following figure 9:

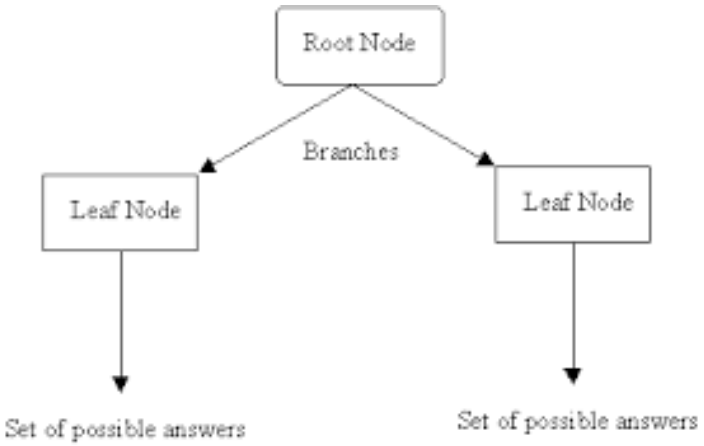


Figure 9. Illustration of Decision Tree Algorithm

Decision tree algorithm places the best attribute of the dataset into the root node and splits the dataset until leaf nodes are reached. Decision trees can be divided into complex, medium and simple trees based on maximum number of splits. There are three split criterions discussed as follows:

- Gini Diversity Index: Gini Index tries to minimize the impurity contained in the training subsets generated after branching the decision tree.
- Twoing Rule: It is rather a goodness measure than an impurity measure. This rule chooses the split at a particular node which maximizes the twoing value function.
- Maximum Deviation Reduction: Splits are chosen such that the standard deviation is reduced. It serves as the best among the three split criterion.

2.3.2.5 Random Forest

Random forests are an ensemble method for regression and it operates by constructing a multitude of decision trees at training time and outputting the mean prediction of the individual trees. It corrects the decision trees habit of overfitting to the training set. In general, the higher the number of trees in the forest gives the high accuracy results. The random forest model is very good at handling tabular data with numerical features, or categorical features with fewer than hundreds of categories. Random forests are able to capture non-linear interaction between the features and the target.

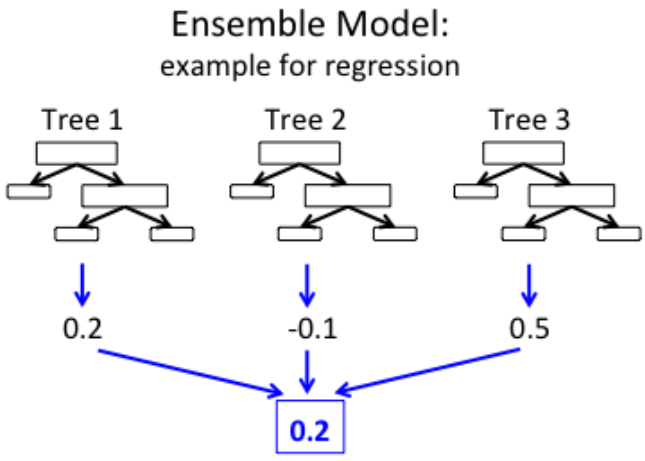


Figure 10. Illustration of Random Forest Algorithm

2.3.2.6 XGBOOST

XGBoost is short for “Extreme Gradient Boosting”. It is used for supervised learning problems, where we use the training data (with multiple features) to classify a target variable as required by the problem statement. The goal of this algorithm is to push the extreme of the computation limits of

machines to provide a ***scalable***, ***portable*** and ***accurate*** classification. XGBoost is an implementation of gradient boosted decision trees designed for speed and performance. The implementation of the algorithm was engineered for efficiency of compute time and memory resources. A design goal was to make the best use of available resources to train the model. Some key algorithm implementation features include:

- Sparse Aware implementation with automatic handling of missing data values.
- Block Structure to support the parallelization of tree construction.
- Continued training so that you can further boost an already fitted model on new data.

The two reasons to use XGBoost are also the two goals of the project:

1. Execution Speed.
2. Computational time.

Generally, XGBoost is fast. Really fast when compared to other implementations of gradient boosting. XGBoost dominates structured or tabular datasets on classification and regression predictive modelling problems. Gradient boosting is an approach where new models are created that predict the residuals or errors of prior models and then added together to make the final prediction. It is called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models. This approach supports both regression and classification predictive modelling problems.

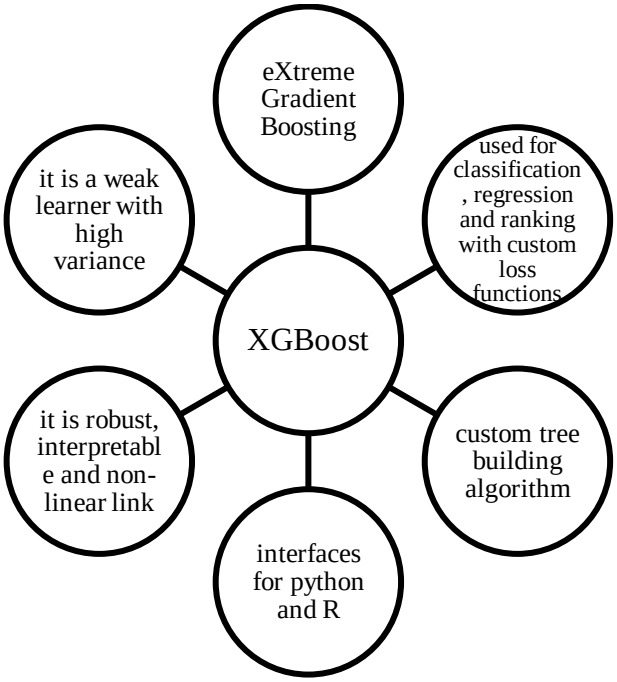


Figure 11. Characteristics of XGBoost

2.3.3 ARTIFICIAL NEURAL NETWORK

When there is a dependency of target or output variable on multiple input parameters there is a requirement to develop a robust and efficient classification model that learns a correlation between the input and output parameters which may not be visible to the mainstream primitive and regression techniques to forecast the rainfall.

An artificial neural network is a signal-processing unit, which is similar in characteristics to the biological neurons. Neural network consists of elements called neurons, nodes or perceptrons, which are the signal processing elements. Nodes are connected to each other by an associated weight that forms the communication link. The output of the model is given by:

$$y = \sum_{i} x_i w_i$$

Where y= output neuron, x_i are the input neurons and w_i are the connection called weights.

i = 0, 1, 2, ……..

The input to the node passes through some activation function to produce an output. Some of the most frequently used activation functions are:

- Log sigmoid transfer function

$$\text{logsig}(n) = 1/(1+\exp(-n))$$

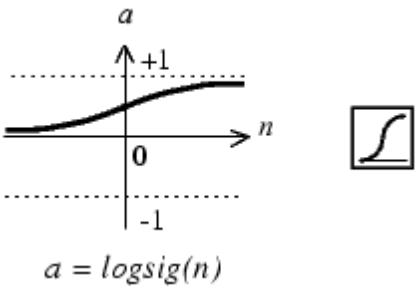


Figure12. Log Sigmoid transfer function

- Tansigmoid Transfer Function

$$\text{tansig}(n) = 2/(1+\exp(-2*n)) - 1$$

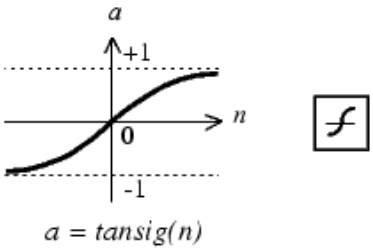


Figure 13. Tansigmoid transfer function

- Purelin Transfer Function

$$a = \text{purelin}(n) = n$$

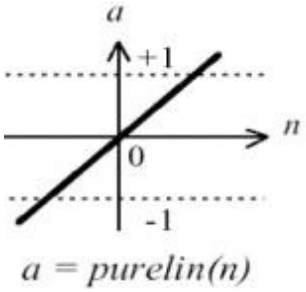


Figure 14. Purelin transfer function

In most of the applications, an extra layer called the hidden layer is also added which is used for complex calculations. It is known as Multi-layer Perceptron Model. The main feature of a neuron and a neural network is its ability to learn from the environment and improve its performance through a learning process.

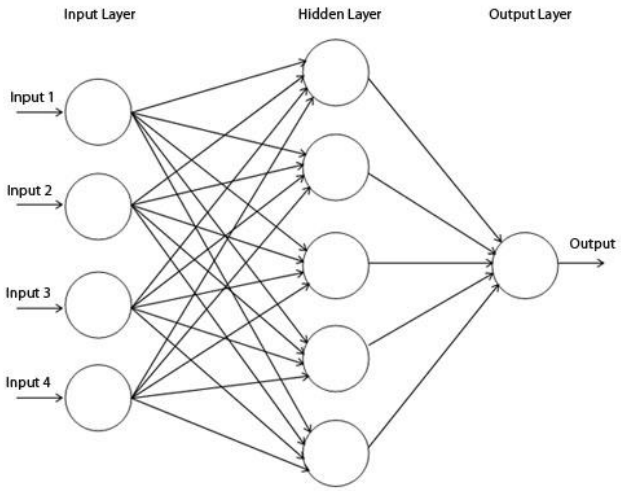


Figure 15. Architecture of Artificial Neural Network

Each time the network will check the value of output with respect to a fixed target to analyse the learning process. If output is equal to the target, then ideally the network need not learn anymore as it has reached the required output, else the network has to learn until the network produces the required output.

If output is not equal to the target, then an error value is computed (difference of output and target) and depending on the value and the sign of the error, a small step of increment or decrement is required or not is decided.

2.3.4 RECURRENT NEURAL NETWORK

Unlike regression predictive modelling, time series also adds the complexity of a sequence dependence among input variables. A powerful type of neural network designed to handle sequence dependence is called recurrent neural network. The Long Short-Term Memory network or LSTM network is a type of recurrent neural network used in deep learning because very large architectures can be successfully trained. It trains the network using Backpropagation Through Time and

overcomes the vanishing gradient problem. Vanishing gradient problem is a difficulty found in training ANN with gradient-based learning methods and backpropagation. Each of the neural network's weights receives an update proportional to the gradient of the error function with respect to the current weight in each iteration of training. The problem lies when the gradient is vanishingly small, effectively preventing the weight from changing its value.

LSTM networks have memory blocks instead of neurons that are connected through layers. Memory blocks has components that makes it smarter than a classical neuron and a memory to store recent sequences. A block contains gates that stores block's state and output. Each gate within a block uses sigmoid activation function to know whether the input sequence is triggered or not.
LSTM can be phrased as a regression problem.

Figure 16 shows the LSTM architecture where LSTM has an internal state variable which is passed from one cell to another and modified by Operation gate.

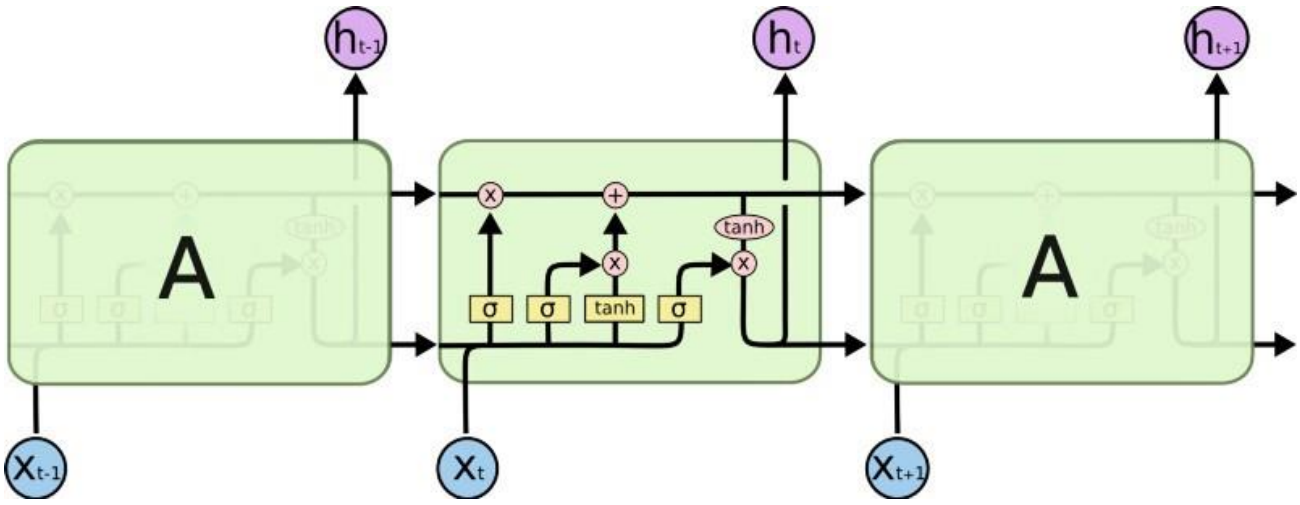


Figure 16. LSTM Architecture

There are three types of gates within a unit of block:

1. Forget gate: conditionally decides what information to throw away from the block.

$$f_t = \sigma\left(W_f.[h_{t-1}, x_t] + b_f\right)$$

It is a sigmoid layer that takes the output at t-1 and the current input at time t and concatenates them into a single tensor and applies a linear transformation followed by a sigmoid. Because of the sigmoid, the output of this gate is between 0 and 1. This number is multiplied with the

internal state and that is why the gate is called a forget gate. If *ft=0* then the previous internal state is completely forgotten, while if *ft=1* it will be passed through unaltered.

2. Input gate: conditionally decides which values from the input to update the memory state.

$$i_t = \sigma\,(W_i.[h_{t-1}, x_t] + b_i)$$

The input gate takes the previous output and the new input and passes them through another sigmoid layer. This gate returns a value between 0 and 1. The value of the input gate is multiplied with the output of the candidate layer.

$$C_t = \tanh(W_C.[h_{t-1}, x_t] + b_C)$$

This layer applies a hyperbolic tangent to the mix of input and previous output, returning a candidate vector to be added to the internal state.

The internal state is updated with this rule:

$$C_t = f_t * C_{t-1} + i_t * C_t$$

The previous state is multiplied by the forget gate and then added to the fraction of the new candidate allowed by the output gate.

3. Output gate: conditionally decides what to output based on input and the memory of the block.

$$O_t = \sigma\,(W_o.[h_{t-1}, x_t] + b_o)$$

$$h_t = O_t * \tanh C_t$$

This gate controls how much of the internal state is passed to the output and it works in a similar way to the other gates.

These three gates described above have independent weights and biases, hence the network will learn how much of the past output to keep, how much of the current input to keep, and how much of the internal state to send out to the output.

2.4 LITERATURE SURVEY

[1] M.Kannan, S.Prabhakaran and P.Ramachandran. Rainfall Forecasting Using Data Mining Technique. International Journal of Engineering and Technology Vol.2 (6), 2010, 397-401.

In this paper Multiple Linear regression is used for the prediction model. The dataset used for training includes the rainfall intensity values of 5 consecutive years during the Monsson season in India which includes the month of September, October and November. The same are used as the predictor values in regression equation generated by the model. Karl Pearson correlation coefficient is used define the correlation between the input and output parameters.

[2] Afolayan Abimbola Helen, Ojokoh Bolanle A., Falaki Samuel O. Comparative Analysis of Rainfall Prediction Models Using Neural Network and Fuzzy Logic. International Journal of Soft Computing and Engineering (IJSCE) ,ISSN: 2231-2307, Volume-5 Issue-6, January 2016.

In this paper ANN and Fuzzy Logic is used for the prediction model. The dataset used for training the model includes rainfall data collected from the automatic weather station in Iju, a town in Akure North Local Government Area of Ondo State . Temperature, surface pressure and other atmospheric parameters were used as the independent variables. Performance analysis was done based on criteria like MSE,MAE, Prediction Error and Prediction Accuracy.ANN model outperformed the Fuzzy Logic model.

[3] Akashdeep Gupta, Anjali Gautam, Chirag Jain, Himanshu Prasad, Neeta Verma. Time Series Analysis of Forecasting Indian Rainfall. International Journal of Inventive Engineering and Sciences (IJIES) ISSN: 2319–9598, Volume-1, Issue-6, May 2013.

In this paper Multi-layered Feed forward ANN is used to develop the prediction model. The dataset used for training involves 140 year monthly data of all India rainfall of 30 metrological subdivisions encompassing 2,8880,324 sq.km. RMSE values are used for evaluating the performances of the models.ANN modelstend to give better results when compared primimtive statistical modelling techniques like Simple and Multiple Linear regressions.The increase in the hidden nodes leads to better results to a point after which the results start to degrade due to increased noises involved.

[4] V.K.Somvanshi, O.P.Pandey, P.K.Agrawal, N.V.Kalanker1, M.Ravi Prakash and Ramesh Chand. Modelling and prediction of rainfall using Artificial neural network and ARIMA techniques. J. Ind. Geophys. Union ,Vol.10, No.2, pp.141-151, April 2006 .

In this paper ANN and Statistical modelling technique like ARIMA is used to develop the prediction model.The dataset used for training the model includes 104 years of mean annual rainfall data from year 1901 to 2003 of Hyderbad region(India).The ANN based model outperforms the ARIMA based in terms of RMSE and R square values thereby giving better prediction results.

[5] Harshani R. K. Nagahamulla, Uditha R. Ratnayake and Asanga Ratnaweera. Monsoon Rainfall Forecasting in Sri Lanka using Artificial Neural Networks. 2011 6th International Conference on Industrial and Information Systems, ICIIS, Sri Lanka, 2011.

In this paper ANN is considered for the model development for it being to tolerate imprecision and uncertainty to a large extend.To model the data correlation analysis is used.The climate indices were used as predictor variables for the model.The technique of network pruning was done to remove weight which failed to give optimal architecture.

[6] Gunawansyah, Thee Houw Liong, Adiwijaya. Prediction and anomaly detection of rainfall using evolving neural network to support planting calender in soreang (Bandung). International Conference on Information and Communication Technology (ICoICT), 2004.

This study proposes ENN which used ANN and Genetic Algorithm(GA) to optimize and find the best weights and biases. From three scenarios, one hidden layer in ANN architecture was sufficient and ENN had good performance in different dataset. The rainfall prediction result used all data (January-December) from 1999-2013 had the accuracy of 84.6%, 66.02% for dry season (April-September) and 79.7% for wet season (October-March).The dataset used for training involves parameters like Rainfall, Sunspot, Cosmic rays, Indian ocean Dipole and Southern Oscillation Index(SOI).

[7] Mislana, Haviluddinb , Sigit Hardwinartoc , Sumaryonod and Marlon Aipassae. Rainfall monthly prediction based on artificial neural network . International Conference on Computer Science and Computational intelligence (ICCSCI), 2015.

In this paper applied an Artificial Neural Network (ANN) with the Backpropagation Neural Network (BPNN) algorithm. In this experiment, the rainfall data were tested using two-hidden layers of BPNN architectures with three different epochs which were [2-50-10-1, epoch 500]; [2-50-20-1, with epochs 1000 and 1500]. The mean square error (MSE) is employed to measure the performance of the classification task. The experimental results showed that the architecture [2-50-20-1, epoch 1000] produced a good result with the value of MSE was 0.00096341.The dataset used for training the

model involves the no linear rainfall data of Tenggarong Station, East Kalimantan-Indonesia in a year.

[8] N. Q. Hung, M. S. Babel, S. Weesakul, and N. K. Tripathi. An artificial neural network model for rainfall forecasting in Bangkok, Thailand. Hydrol. Earth Syst. Sci., 13, 1413–1425, 2009 .

In this paper presents a new approach using an Artificial Neural Network technique to improve rainfall forecast performance. The developed ANN model is being applied for real time rainfall forecasting and flood management in Bangkok, Thailand. the use of a combination of meteorological parameters (relative humidity, air pressure, wet bulb temperature and cloudiness), the rainfall at the point of forecasting and rainfall at the surrounding stations, as an input data, advanced ANN model to apply with continuous data containing rainy and non-rainy period, allowed model to issue forecast at any moment. ANN model were compared to the convenient approach namely simple persistent method where ANN proved to have superior results.The dataset used for training the model includes the Monthly rainfall data from the year 1991-2004

[9] JamilehFarajzadeh ,AhmadFakheri Fard and SaeedLotfi. Modeling of monthly rainfall and runoff of Urmia lake basin using “feed-forward neural network” and “time series analysis” model. Water Resources and Industry 7-8 (2014) 38–48.

In this book chapter various methods for time-series based forecasting are implemented in the presented study Feed-forward Neural Network and Autocorrelation Regressive Integrated Moving Average (ARIMA) models were applied to forecast the monthly rainfall in Urmia lake basin. The results showed that the estimated values of monthly rainfall through Feed-forward NN were close to ARIMA model with coefficient of correlation 0.62 and the root mean square error of 12.43 mm over the 6 years test period.The rainfall amount were predicted for a 6-year period starting from 2012 (2012–2017). Using the runoff coefficient regime which was calculated from parallel data of rainfall over the basin and resulted runoff for the period of 39 years, the future runoff were obtained through predicted rainfall over that period. Monthly rainfall of 228 stations inside and outside of Urmia lake were used as the dataset for training the model.

[10] JianshengWu ,JinLong and MingzheLi . Evolving RBF neural networks for rainfall prediction using hybrid particle swarm optimization and genetic algorithm. Neurocomputing Volume 148, 19 January 2015, Pages 136-142.

In this paper, an effective hybrid optimization strategy by incorporating the adaptive optimization of particle swarm optimization (PSO) into genetic algorithm (GA), namely HPSOGA, is used for

determining the parameters of radial basis function neural networks (number of neurons, their respective centers and radii) automatically. In HPSOGA, individuals in a new generation are created through three approaches to improve the global optimization performance, which are elitist strategy, PSO strategy and GA strategy. The findings reveal that the hybrid optimization strategy proposed here may be used as a promising alternative forecasting tool for higher forecasting accuracy and better generalization ability. The dataset used for training involves the monthly rainfall collected from 24 stations of Liuzhou Meteorology Administration rain gauge networks from 1949-2011.

CHAPTER 3

DESIGN AND METHODOLOGY

This chapter deals with the design and methodology adopted to forecast the rainfall. Section 3.1 depicts the flow chart pertaining to the methodology used for forecasting. Section 3.2 and 3.3 briefly discusses about the acquisition and consolidation of data followed by the pre-processing of the categorical data. The techniques such as imputation, categorical encoding, feature scaling and splitting of dataset are discussed in detail. Section 3.4 discusses about the implementation of primitive techniques, NAR, regression techniques, NARX, Artificial Neural Network and Recurrent Neural Network and explains the development of model with different architectures. Section 3.5 summarises the performance measures for each of the techniques used to evaluate the model.

3.1 PROPOSED METHODOLOGY

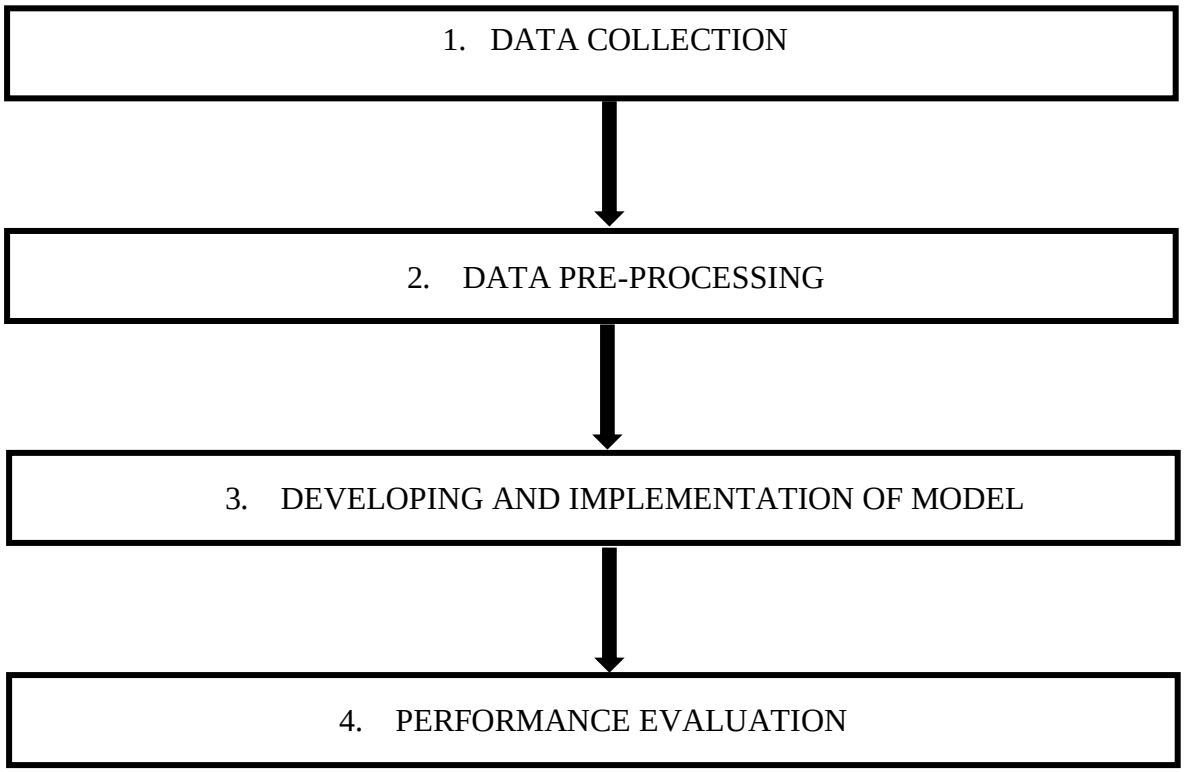


Figure 17. Methodology of the project

3.2 DATA COLLECTION

The first stage of the project is the accumulation and consolidation of the rainfall data of Coonoor in Nilgiris district. The data have been collected from the India Meteorological Department (IMD), Chennai and the Public Works Department of Coonoor for a period of 10 years from 2004-2013. The rainfall intensity of 3 nearby stations such as Coonoor rain gauge station, Coonoor railway station and Runnymedu are also considered as the input parameters. The dataset consolidates rainfall and other parameters such as temperature, relative humidity, cloud coverage, wind speed and wind direction. Table 2 contains all the information of dataset.

The area of study is Coonoor in Nilgiris district, Tamil Nadu, India. There are 14 rain gauge stations in and around Coonoor, which keep a record of the amount of rainfall received. Figure 18 shows all the rain gauge stations near Coonoor.

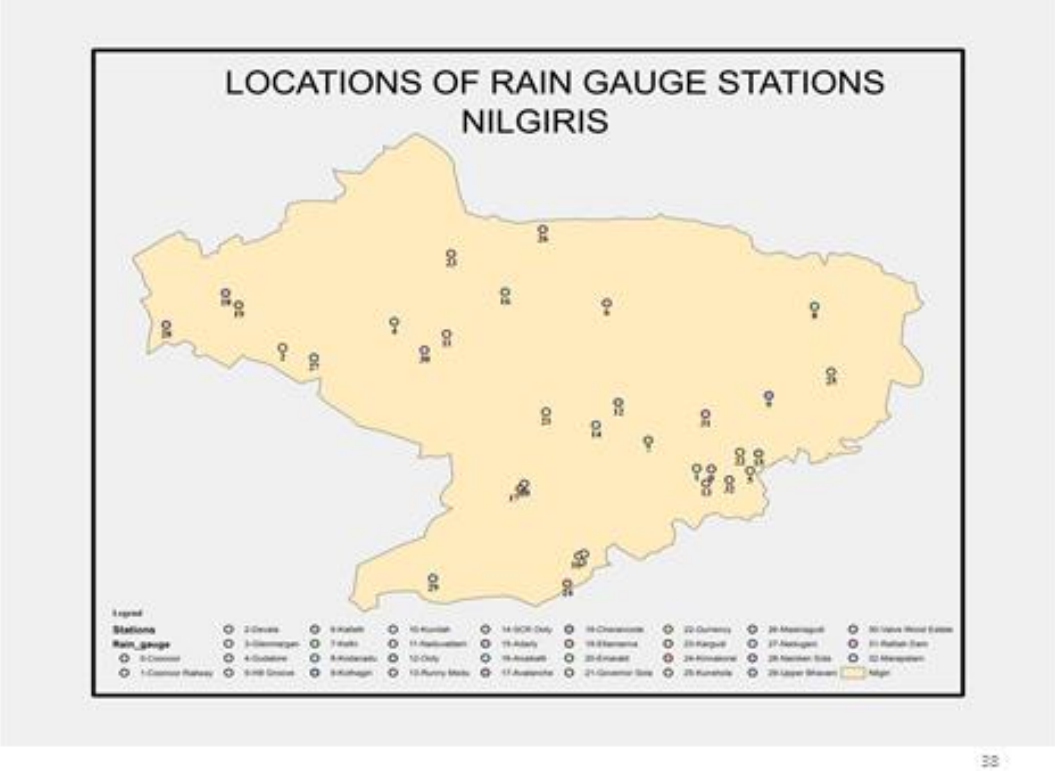


Figure 18. Rain Gauge Stations of Nilgiris

Table 2. Information about dataset

Study Area	Coonoor, Nilgiris, Tamil Nadu
Source of Data	Indian Meteorological Department, Chennai Public Works Department, Nilgiris
Data Acquisition Time Span	10 years i.e. 2004-2013
Input Parameters	Temperature (max and min), Relative humidity (max and min), cloud coverage (max and min), wind speed (max and min), wind direction (17), rainfall intensities of nearby rain gauge stations (3)
Number of Input Parameters	28
Output Parameter	Rainfall (in mm)
Number of Output Parameter	1

Figure 19 shows a portion of the dataset of interest. It shows the column of Wind Direction which needs to be pre-processed before developing the model using any algorithms.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	WD(83ESE)	T(MAX)	T(MIN)	RH(830)	RH(1730)	WS(830)	WS(1730)	CC(830)	CC(1730)	COONOOR	C.RS	RUNNY MEDU	RAINFALL	
2	ESE	20.4	11.8	98	83	4	4	3	5	7.2	0	5	4.8	
3	ESE	20.2	13	100	81	2	6	6	5	17.4	15	9	14.5	
4	SSE	18	10	88	94	2	4	4	8	0	0	0	0	
5	W	16.8	8	53	57	2	4	0	1	0	0	0	0	
6	S	19.6	10.2	82	88	4	0	4	6	0	0	0	0	
7	SE	19.2	12.8	98	88	2	4	0	6	0	0	0	0.5	
8	CALM	19.2	10.8	94	86	0	4	6	7	0	0	0	0	
9	SE	20	11.8	98	94	2	6	3	4	0	0	0	1.2	
10	SSE	20.2	8.4	47	77	2	2	3	2	0	0	0	0	
11	SSE	21.6	9.2	71	90	4	6	1	7	0	0	0	0	
12	SE	19.6	7.6	77	73	2	6	0	0	0	0	0	0	
13	CALM	21.8	8	29	73	0	4	0	0	0	0	0	0	
14	SE	21.8	8.8	36	68	2	6	0	0	0	0	0	0	
15	SE	21.4	9.2	47	64	2	4	0	0	0	0	0	0	
16	SW	22	9.8	39	62	4	2	0	0	0	0	0	0	
17	SW	21	9.8	43	49	2	2	1	0	0	0	0	0	
18	SSE	21	9	41	63	2	2	2	1	0	0	0	0	
19	SW	21.6	10.2	47	64	2	2	2	0	0	0	0	0	
20	ESE	22.2	9.8	68	82	2	4	0	0	0	0	0	0	
21	SSE	21.4	10.4	74	83	2	4	0	4	0	0	0	0	

COONOOR

Figure 19: A sample portion of dataset

3.3 DATA PRE-PROCESSING

Data pre-processing is a data mining techniques which involves the transformation of data into understandable format. Real-time collected data may have outliers, missing data or it may be inconsistent which can lead to improper training and developing of classification models. Data pre-processing techniques generally include data cleaning, data transformation, data integration, data

reduction and data discretization. Figure 20 shows the data pre-processing techniques used in the project for training the model.

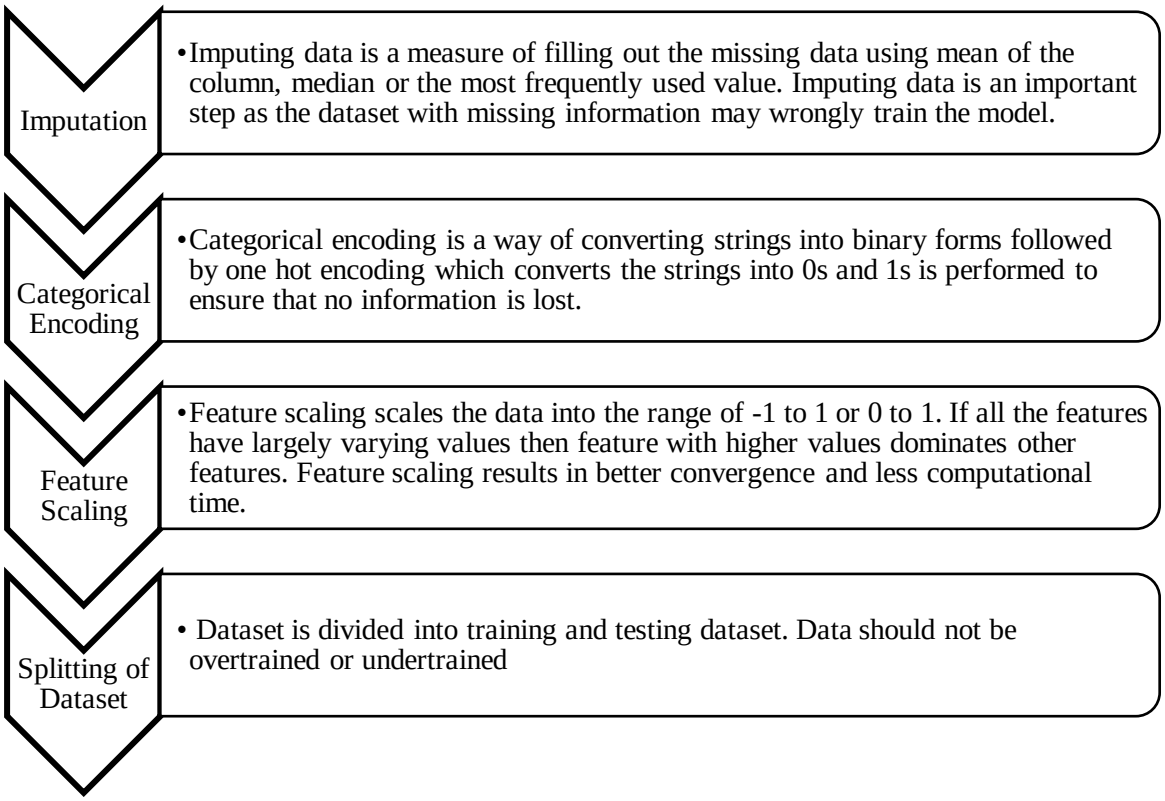


Figure 20. Data Pre-processing techniques

Section 3.3.1 shows the software implementation of each of the data pre-processing techniques which is used in all the algorithms and techniques. It gives an insight on how the individual code blocks can be implemented for all the techniques discussed in Chapter 2.

3.3.1 SOFTWARE IMPLEMENTATION OF DATA PRE-PROCESSING TECHNIQUES

1. Imputation Code Block

```
#TAKING CARE OF MISSING DATA
from sklearn.preprocessing import Imputer
imputer=Imputer(missing_values='NaN',strategy='mean',ax
is=0)
imputer=imputer.fit(X[:,1:13])
X[:,1:13]=imputer.transform(X[:,1:13])
```

2. Categorical Encoding Code Block

```
#ENCODING CATEGORICAL DATA
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder_X=LabelEncoder()
X[:,0]=labelencoder_X.fit_transform(X[:,0])
onehotencoder=OneHotEncoder(categorical_features=[0])
#X = X.as_matrix().astype(np.float)
X=onehotencoder.fit_transform(X).toarray()
labelencoder_y=LabelEncoder()
y=labelencoder_y.fit_transform(y)

X = X[:, 1:]
```

3. Feature Scaling Code Block

```
#FEATURE SCALING
from sklearn.preprocessing import StandardScaler
sc_X=StandardScaler()
X_train=sc_X.fit_transform(X_train)
X_test=sc_X.transform(X_test)
```

4. Splitting of Dataset Code Block

```
#SPLITTING DATASET

from sklearn.cross_validation import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size
=0.2,random_state=0)
```

3.4 DEVELOPMENT AND IMPLEMENTATION OF FORECASTING MODELS

This section deals with the implementation and development of the algorithms discussed in Chapter 2. The algorithms will be implemented on the dataset of Coonoor as shown in figure 19. Table 2 shows the input and output parameters and later discusses the comparison of forecasting models based on univariate (only rainfall intensity) and Multivariate data (rainfall intensity along

with temperature, humidity and so on). The developed models will be evaluated and discussed in the Chapter 5.

3.4.1 Implementation of Baseline/ Naïve method to forecast rainfall

A basic model is developed using this algorithm which predicts the next day rainfall intensity using the immediate previous rainfall intensity. This type of data is known as Univariate data as it uses one input parameter to forecast the rainfall intensity. The Persistence/ Naïve Forecast is implemented in Python platform using different libraries. The following steps are followed to implement the Naïve forecasting technique:

1. Import all the required libraries.
2. Import the dataset with single column of rainfall intensity of Coonoor.
3. Create a lagged dataset which splits the column with a lag of one day (t-1).
4. Filling out the missing values with zero.
5. Split the dataset into training and testing sets.
6. Define the persistence model function.
7. Predict the RMSE value using walk-forward validation code block.
8. Plot predictions and expected results.

3.4.2 Implementation of Seasonal Persistence Algorithm to forecast rainfall

As the Naïve forecast is an important to predict the future rainfall intensity of a particular day or month of Coonoor, it averages the rainfall intensities of previous seasons same days or months using Univariate data of rainfall intensities. It can be implemented using the following steps;

1. Import all the libraries.
2. Load the dataset of Coonoor with rainfall intensity.
3. Split the training and testing dataset.
4. Evaluate the mean of different number of years present in training set.
5. Define walk forward validation function which collects the observation list of all the training years/
6. Calculates the mean of the observation list and makes the prediction accordingly.
7. Performance measure based on RMSE is calculated and accordingly graphs are plotted.

3.4.3 Implementation of AutoRegression to forecast rainfall

As the seasonal persistence algorithm, averages the previous seasons data, leap years cannot be included which limits the model design. To develop the Seasonal Persistence Algorithm in this project, February 29 data values have been removed for simplification of the model implementation. This calls for a new technique to forecast the model using regression equation which uses the correlation between the inputs and the output. The following is the procedure of AutoRegression technique:

1. Import all the necessary libraries
2. Load the dataset with Univariate data
3. Split the dataset into training and testing sets.
4. Train the Autoregression model and output the coefficients
5. Make predictions accordingly with the actual and expected rainfall intensity or run the walk forward algorithm code block over time steps in test which is supposed to be mine.
6. Evaluate based on RMSE and plot the results

3.4.4 Implementation of ARIMA to forecast rainfall

ARIMA is a statistical tool which integrates the basic AutoRegression model mentioned in last section with the Moving Average model that uses the dependency between an observation and a residual error applied to lag observations. ARIMA makes the time series stationary by subtracting an observation from an observation at the previous time step. ARIMA can be designed as follows:

1. Import necessary libraries.
2. Import the dataset with Univariate data.
3. p, d, q values are calculated using Grid Search Algorithm and the best (p, q, d) will be used by the ARIMA model to forecast.
4. i) ARIMA model is fitted with the best (p,d,q)
   ii) Plot residual error to evaluate the model.
5. A variation of ARIMA i.e. Rolling Forecast ARIMA model was designed to compare the performance with the basic model.
6. In RF-ARIMA, model was fitted using the best [p,d,q] value and after each iteration of testing, the testing set gets added to the training model and evaluated again.
7. RF-ARIMA will be evaluated based on the MSE value of the model.

3.4.5 Implementation of Non-Linear AutoRegressive Neural Network to forecast rainfall

The dataset of Coonoor with Univariate data of rainfall intensity is evaluated using NAR network where the past values of y(t) is used to forecast the y(t). NAR has been implemented in MATLAB with the following specifications:

1. Open the Neural Network Toolbox (type nnstart in command window).
2. Use the time series app to implement NAR model.
3. Import the dataset from the workspace.
4. Split the dataset into training, testing and validation.
5. A balanced number of hidden neurons and delays are chosen.
6. Train the network using Levenberg-Marquardt which plots the error histogram along with time series response, error autocorrelation plot and input-error correlation plot.
7. The MSE values of training, testing and validation with regression values are compared with the different architectures of NAR. (discusses in next chapter)

3.4.6 Implementation of NARX Neural Network to forecast rainfall

The dataset of Coonoor with Multivariate data of rainfall intensity with temperature, relative humidity, wind speed, wind direction, cloud coverage and nearby rain gauge stations is evaluated using NARX network where the past values of y(t) and x(t) is used to forecast the y(t). NARX has been implemented in MATLAB with the following specifications:

1. Open the Neural Network Toolbox (type nnstart in command window).
2. Use the time series app to implement NARX model.
3. Import the dataset from the workspace.
4. Split the dataset into training, testing and validation.
5. A balanced number of hidden neurons and delays are chosen.
6. Train the network using Levenberg-Marquardt which plots the error histogram along with time series response, error autocorrelation plot and input-error correlation plot.
7. The MSE values of training, testing and validation with regression values are compared with the different architectures of NARX. (discussed in Chapter 5).

3.4.7 Implementation of Regression Techniques to forecast rainfall

Implementation of SVR, DT, RF and XGBOOST

1. **Support Vector Regression**
    a. Import Library
    b. Import dataset of Coonoor.
    c. Convert string to float or integer.
    d. Split the dataset into training and testing set.
    e. Create SVR regression object with kernel function.
    f. Fit the model to the training set, make predictions and calculate MSE.
2. **Decision Trees**
    a. Import Library
    b. Import dataset of Coonoor.
    c. Convert string to float or integer.
    d. Split the dataset into training and testing based on an attribute and an attribute value.
    e. Calculate the gini index for a split dataset and select the best split point for a dataset.
    f. Create DT regression object with gini index splitting criterion.
    g. Build the tree regressor model and check the scores.
    h. Fit the model to the training set, make predictions and calculate MSE.
3. **Random Forest**
    a. Import Library
    b. Import dataset of Coonoor.
    c. Convert string to float or integer.
    d. Split the dataset into training and testing based on an attribute and an attribute value.
    e. Calculate the Gini index for a split dataset and select the best split point for a dataset.
    f. Create DT regression object with Gini index splitting criterion.
    g. Build the DT and make prediction.

h. Create a random subsample from the dataset and make prediction with a list of bagged trees.
i. Fit the model to the training set, make predictions and calculate MSE.

4. XGBoost

a. Import the libraries
b. Import the dataset
c. Encode the categorical data
d. Split the dataset into training and testing set
e. Fitting XGBoost to the training set using XGBRegressor.
f. Predict the test set results.
g. Apply k-fold cross validate and validate the results.

3.4.8 Implementation of Artificial Neural Network to forecast rainfall

When there is a dependency of target or output variable on multiple input parameters there is a requirement to develop a robust and efficient prediction model that learns a correlation between the input and output parameters which may not be visible to the mainstream regression techniques to forecast the rainfall in Coonoor.

ANN model was implemented and analyzed on Python platform. Various models were developed based on different number of hidden layers and nodes.

Models described in Table 3 uses **'sigmoid'** activation function in the hidden layer and linear activation function for output layer.

Table 3. ANN Architectures for Prediction

S. No.	Hidden layers	Hidden nodes
Model-1	2	12
Model-2	2	13
Model-3	2	14
Model-4	2	15
Model-5	2	16
Model-6	3	12
Model-7	3	13
Model-8	3	14
Model-9	3	15
Model-10	3	16

Nominal hidden nodes were calculated by

= (No. of input parameters + No. of Output Parameters)/2

i.e. = (28+1)/2 = 14 hidden nodes

All the models were evaluated based on the RMSE and loss value. The different models of ANN developed for the Coonoor rainfall dataset with 28 inputs were tabulated and compared for the best ANN model for prediction with nominal error.

The best architecture for rainfall forecasting in ANN can be implemented by PCA and k-PCA feature extraction techniques.

PCA: It is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components.

kernel-PCA: It is an extension of PCA using techniques of kernel methods. Using a kernel, the originally linear operations of PCA are performed in a reproducing kernel Hilbert space.

3.4.9 Implementation of Recurrent Neural Network to forecast rainfall

LSTM RNN has been implemented in the project on both Univariate and Multivariate data and compared based on RMSE values. LSTM is a type of RNN where large architectures are successfully trained using Backpropagation through Time. Let's look at the software implementation of Recurrent Neural Network:

1. Import the required libraries by RNN.
2. Import the training set with rainfall intensities.
3. Take care of the missing values using Imputation Code block discussed earlier.
4. Feature Scale the input attributes using Feature Scaling Code block.
5. Create a data structure with 60, 120, 180 time steps and 1 output.
6. Initialise the RNN model.
7. Initialise LSTM layers and dropout regularisation.
8. Compile and fit the output RNN architecture to the training set.
9. Make predictions and visualise the results with the RMSE value.

3.5 PERFORMANCE EVALUATION OF FORECASTING MODELS

Forecasting rainfall being a sensitive analysis due to its application for disaster management, it's highly important to choose a model with best performance. For based on the forecasted rainfall, the required processing techniques are required. Forecast of rainfall with huge error can prove to be fatal. Hence, evaluation of the model is an important measure for sustainability of human and environmental health.

3.5.1 Performance measures of rainfall forecasting models using Statistical Modelling

- **Root Mean Square Error**: RMSE is a frequently used measure of the differences between values predicted by a model and the values actually observed. It is a measure of accuracy, to compare classification errors of different models for a particular data and not between datasets, as it is scale-dependent. It is the square root of the average of squared errors.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

- **Line Plots**: Line plot is a statistical graph which represents the data recorded in experiments or surveys. Line plot is a statistical graph which represents the data recorded in experiments or surveys.
- **Box and Whisker Plots**: A box plot is a method for graphically depicting groups of numerical data through the quartiles. Box plots have line extending vertically from bees known as whiskers indicating variability outside the upper and lower quartiles.
- **Histogram**: It is a plot whose area is proportional to the frequency of a variable and whose width is equal to the class interval.
- **Heat Maps**: A representation of data in the form of a diagram in which data values are represented as colors.
- **Auto Correlation Plot**: It is a plot of the sample autocorrelations versus the time lags.
- **Density Plots**: It visualizes the distribution of data over a continuous interval or time period. The peaks of density plot help display where values are concentrated over the interval.

3.5.2 Performance Measures of rainfall forecasting models using NAR and NARX

- **Mean Square Error**: MSE of an estimator measures the average of the squares of the errors, that is, the difference between the estimator and what is estimated.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

- **Error Histogram**: It shows the histogram plot of errors for each training set.
- **Error Auto Correlation**: It shows the autocorrelation of the error within the set confidence limit.
- **Input Error Correlation**: It takes an input time series and an error time series and plots the cross-correlation of inputs to errors across varying lags.

3.5.3 Performance Measures of rainfall forecasting models using Regression Techniques

- **R-square**: It is a statistical measure of how close the data are to the fitted regression line. It is the percentage of the response variable variation explained by linear model.
- **Adjusted R-square**: It is a modified version of r-squared that has been adjusted for the number of predictors in the model.
- **Root Mean Square Error**

3.5.4 Performance Measures of rainfall forecasting models using ANN and RNN

- **Loss**: Loss functions for classification are computationally feasible loss functions representing the price paid for inaccurate classification.
- **Root Mean Square Error**

CHAPTER – 4

SOFTWARE IMPLEMENTATION

4.1 INTRODUCTION TO MATLAB TOOLBOXES

MATLAB is a high-level language and interactive environment for numerical computation, visualization, and programming. MATLAB can be used to analyse data, develop algorithms, and create models and applications. The language, tools, and built-in math functions provide a path to explore multiple approaches and reach a solution faster than with spreadsheets or traditional programming languages, such as C/C++ or Java.

MATLAB is a wide platform for a range of applications, including signal processing and communications, image and video processing, control systems, test and measurement, computational finance, and computational biology. More than a million engineers and scientists in industry and academia use MATLAB, the language of technical computing.

This project explored and employed the Neural Network Toolbox of MATLAB 2015a to design and improve various algorithms for classification.

4.1.1 NEURAL NETWORK TOOLBOX

Neural Network Toolbox provides algorithms, pre-trained models, and apps to create, train, visualize, and simulate both shallow and deep neural networks. You can perform classification, regression, clustering, dimensionality reduction, time-series forecasting, and dynamic system modeling and control. Neural Network Toolbox includes command-line functions and apps for creating, training, and simulating shallow neural networks. The apps make it easy to develop neural networks for tasks such as classification, regression (including time-series regression), and clustering. After creating your networks in these tools, you can automatically generate MATLAB code to capture your work and automate tasks.

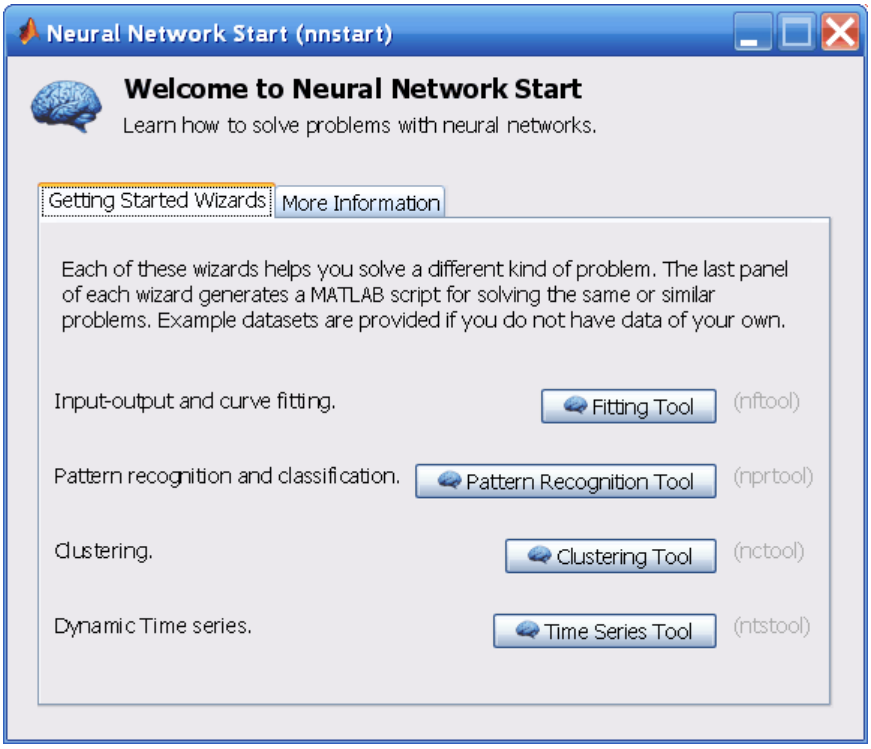


Figure 21. Neural Network Toolbox

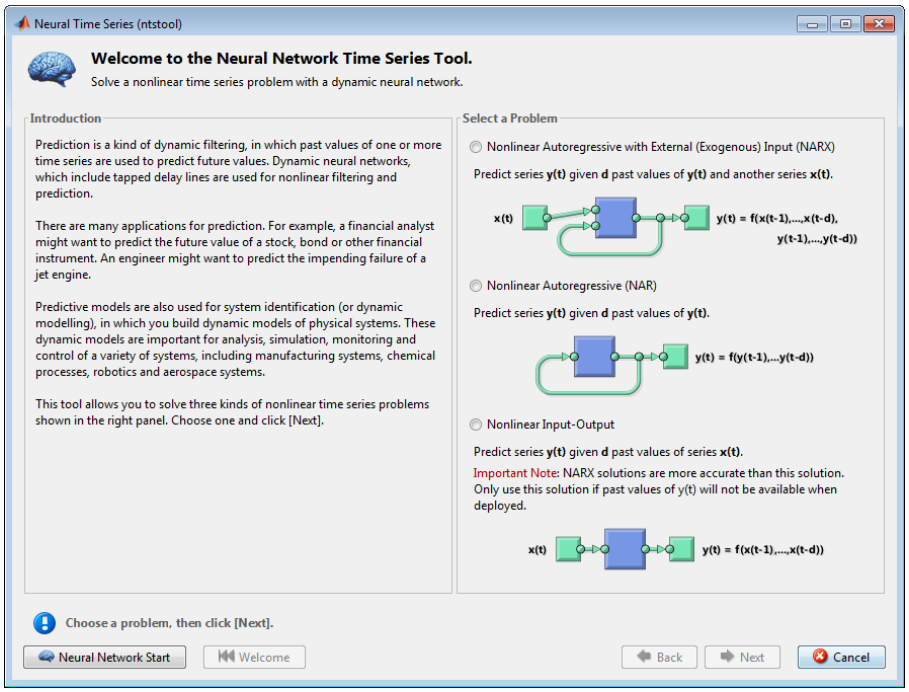


Figure 22. Getting Started with Neural Network Toolbox

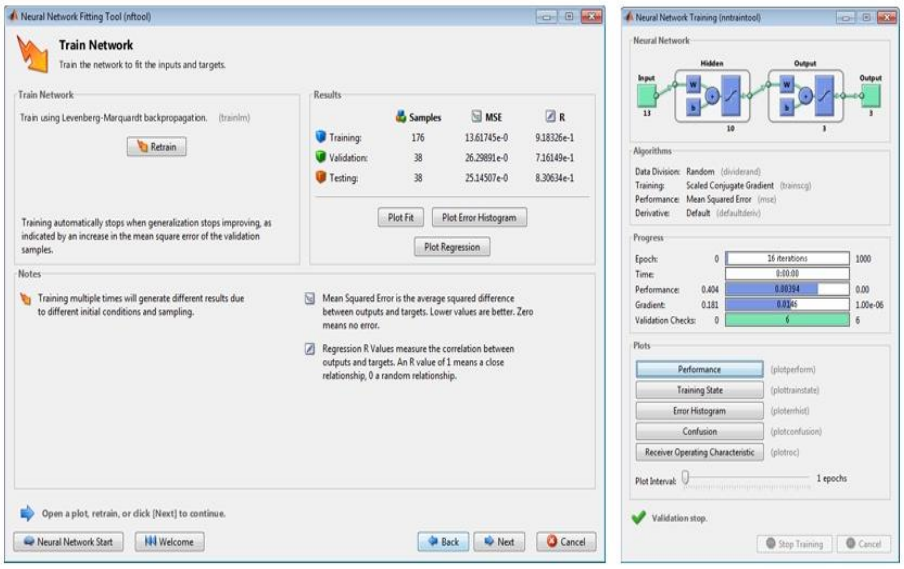


Figure 23. Evaluation Check Window of NN Toolbox

4.2 INTRODUCTION TO PYTHON

Python is an object-oriented, high-level programming language with integrated dynamic semantics primarily for web and app development.

Python is relatively simple, so it's easy to learn since it requires a unique syntax that focuses on readability. Developers can read and translate Python code much easier than other languages. In turn, this reduces the cost of program maintenance and development because it allows teams to work collaboratively without significant language and experience barriers.

Additionally, Python supports the use of modules and packages, which means that programs can be designed in a modular style and code can be reused across a variety of projects. Once you've developed a module or package you need, it can be scaled for use in other projects, and it's easy to import or export these modules.

Figure 24 shows the workspace of Python platform which is used for the implementation of ANN and Genetic Algorithm.

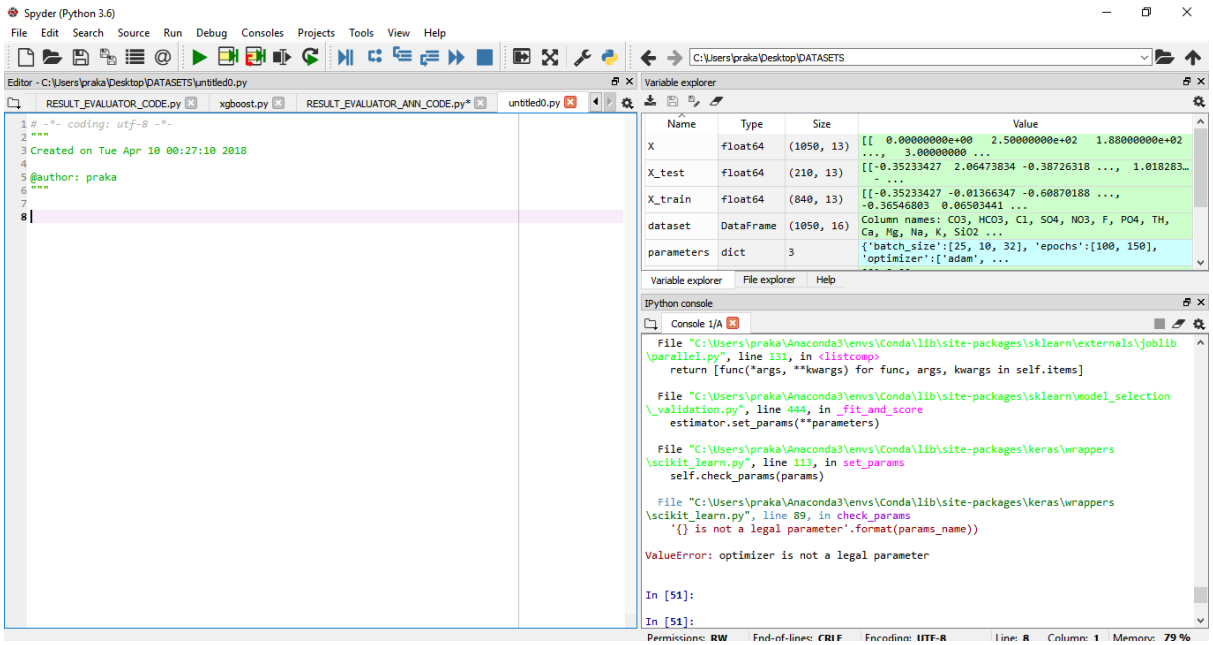


Figure 24. Python Workspace

CHAPTER 5

RESULTS AND DISCUSSIONS

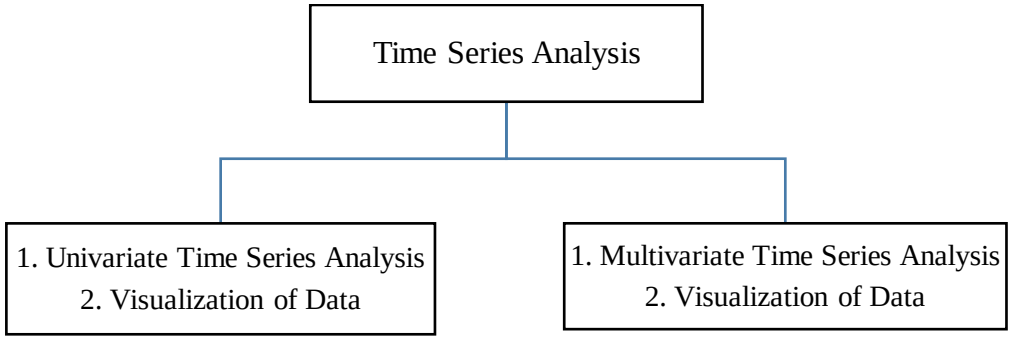
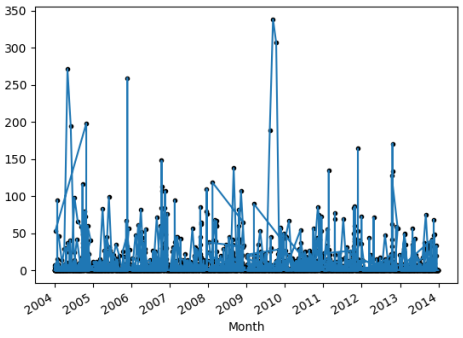


Figure 25. Time Series Analysis Division

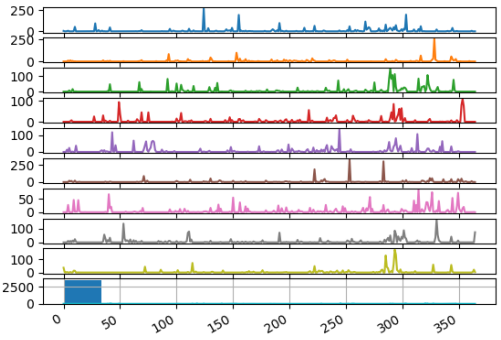
This chapter discusses about the implementation results of all the techniques discussed in Chapter 3. Section 5.1 visualizes the Univariate data based on line, box and whisker, lag, series, density and autocorrelation plots, histogram and heat maps. Section 5.2 shows the results for Univariate time series analysis of Baseline/Naïve forecast, Seasonal Persistence algorithm, Autoregression, ARIMA, NAR and RNN (uni). Later, section 5.3 discusses different multivariate time series analysis. Results of NARX, Regression techniques, Artificial neural network and Recurrent neural network are discussed in detail. Section 5.4 compares the Univariate and multivariate time series analysis and suggests the best technique to forecast the rainfall for the dataset in hand.

5.1 VISUALIZATION OF UNIVARIATE DATA

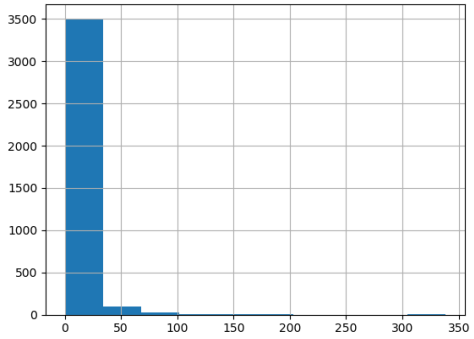
Figure 26 visualises the data using Line plots, Histograms, Density plots, Box and Whisker plots, Lag plots, Autocorrelation plot and Heat maps of the rainfall intensity values within various time ranges. Line, whisker, lag plots have been plotted and visualised weekly as well as yearly whereas density, histogram and autocorrelation plots have been visualised for 10 years from 2004-2013. It shows the distribution of observations to provide valuable diagnostics of the Univariate data to develop a better forecasting model. Heat map is a plot which represents the rainfall intensity value based on the contrast of the colours present in the plot. Columns represents the number of months and rows represent the number of days in a month. It shows the distribution of rainfall intensity over the different days in a month.


Month

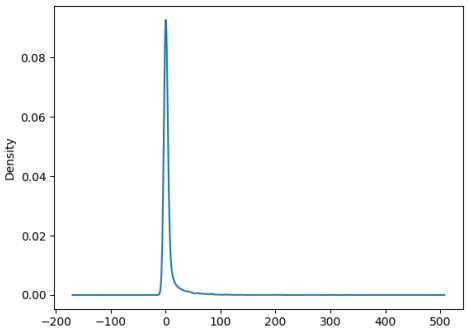
(a)



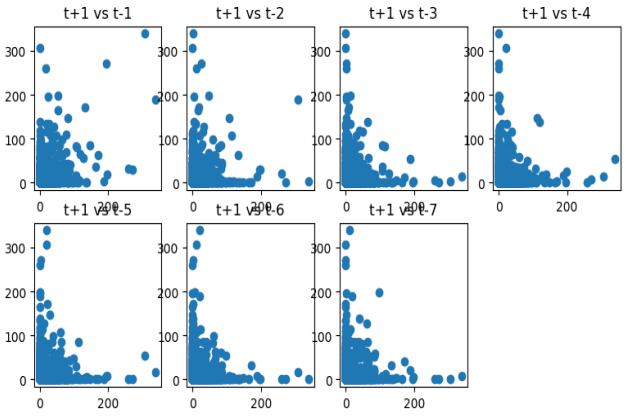
(b)



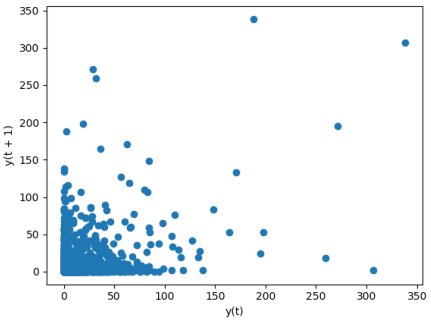
(c)


Density

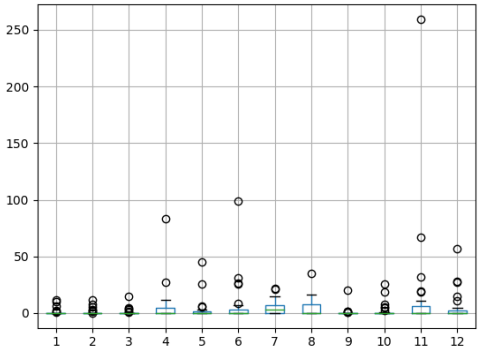
(d)


t+1 vs t-1
t+1 vs t-2
t+1 vs t-3
t+1 vs t-4
t+1 vs t-5
t+1 vs t-6
t+1 vs t-7

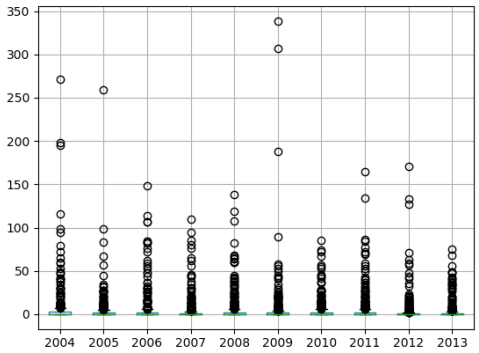
(e)


y(t + 1)
y(t)

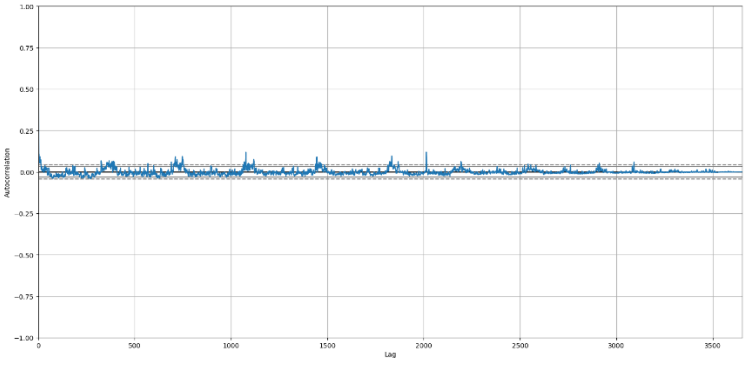
(f)


250
200
150
100
50
0
1 2 3 4 5 6 7 8 9 10 11 12

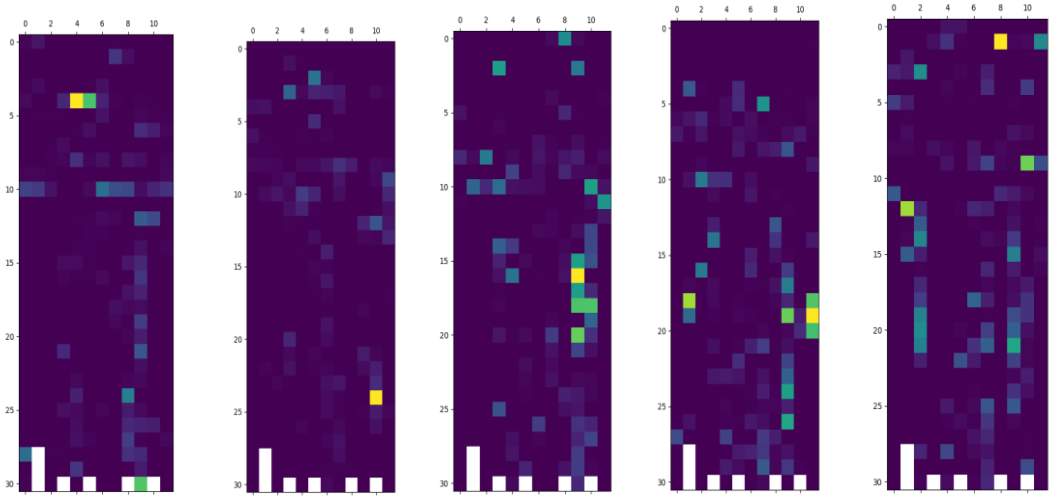
(g)


350
300
250
200
150
100
50
0
2004 2005 2006 2007 2008 2009 2010 2011 2012 2013

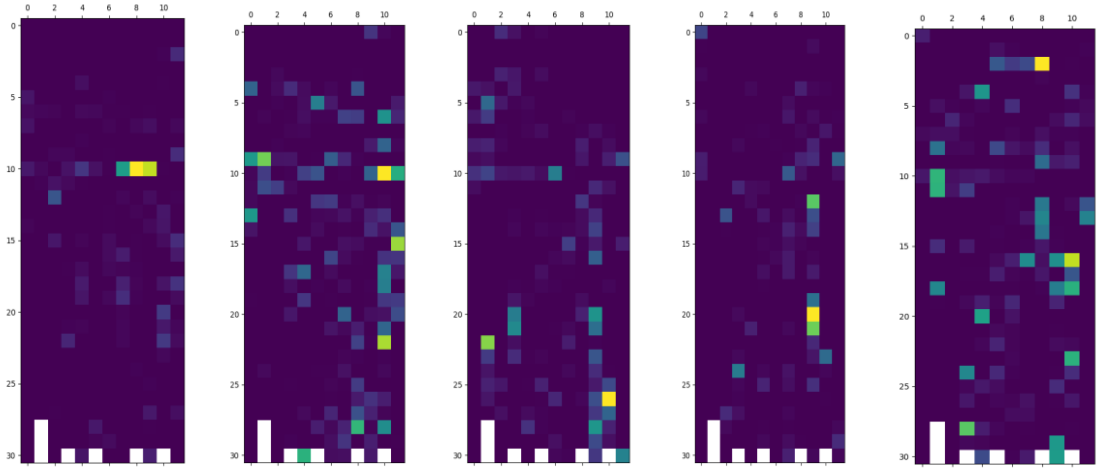
(h)


Autocorrelation
Lag

(i)



(j) (k) (l) (m) (n)



(o) (p) (q) (r) (s)

Figure 26. (a) Monthly Series Plot (b) Yearly Series Plot (c) Histogram Plot (d) Density Plot (e) Lag Plot Weekly (f) Lag Plot Yearly (g) Whiskers Plot Monthly (h) Whiskers Plot Yearly (i) Auto Correlation Plot (j)-(s) Heat Maps of all 10 years.

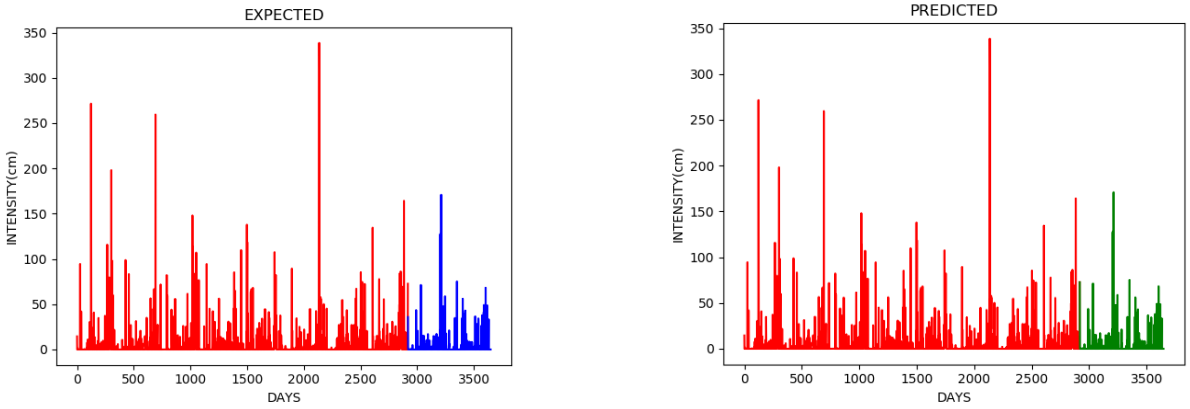
5.2 UNIVARIATE TIME SERIES ANALYSIS

Univariate time series include the following algorithms to forecast the model as shown in following sections:

a. Persistence algorithm / Naïve forecast
b. Seasonal Persistence Algorithm
c. AutoRegression
d. Auto Regression Integrated Moving Average
e. Non-linear Auto Regressive Time Series Analysis
f. Recurrent Neural Network

5.2.1 BASELINE / NAÏVE FORECAST

As discussed, this method acts as a reference to other modelling techniques. As can be seen in table 4 the time step (t-1) is used to forecast the rainfall amount of time step (t). This method is the baseline to all other forecasting methods. Figure 27 shows the expected and predicted amount of rainfall from 2004-2013.



(a) (b)

Figure 27. (a) Expected Rainfall Intensity (b) Predicted Rainfall Intensity

Table 4. Naïve Baseline Forecast Model

Expected amount of Rainfall	Predicted amount of rainfall
72.6	35.8
35.8	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0.3
0.3	0.4
0.4	0

5.2.2 SEASONAL PERSISTENCE ALGORITHM

Figure 28 shows the yearly seasonal variation of rainfall over 365-366 days (i.e. 12 months)in 10 years. Table 5 shows that to reach to the minimal error point with Seasonal persistence technique for the target dataset we have to take into consideration past 8 years of data.

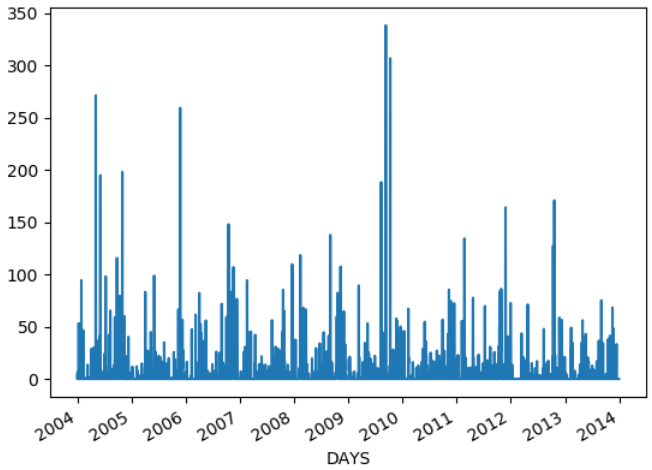
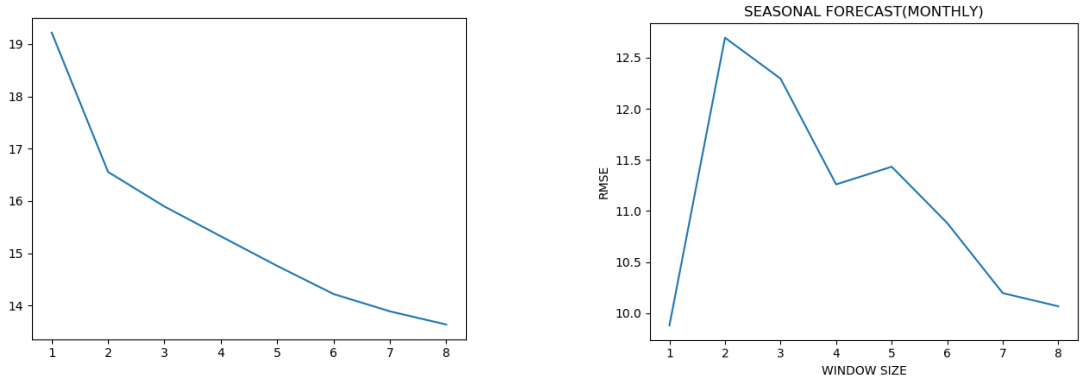


Figure 28. Seasonal variation of rainfall (10 years).

Table 5. RMSE of Monthly and Daily days of Seasonal Persistent Algorithm

Years	RMSE_M	RMSE_D
1	9.88026	19.2212
2	12.6957	16.5564
3	12.295	15.8951
4	11.2601	15.3252
5	11.4332	14.7592
6	10.8819	14.2212
7	10.196	13.8897
8	10.0673	13.6372

Figure 29 shows that the root mean square error value decreases continuously throughout the training period of 8 years whereas in figure although initially root mean square error increased for three years but it gradually decreased for the next five years. In general it can be concluded that, as the sliding window size in terms of number of years increases for the seasonal persistence model, forecasting of rainfall can be done more accurately and precisely.



(a) (b)

Figure 29. (a) Daily Persistence Model (b) Monthly Persistence Model

5.2.3 AUTO REGRESSION

Figure 30 (a): shows a plot of the Rainfall intensity values on the different days in a year from 2004 -2013. It helps to understand the generalised pattern followed by the rainfall over the different seasons like Autumn, Monsoon, Summer and Winter in a year.

Figure 30 (b):is a Line plot which gives a deeper insight into the correlation between the rainfall intensity value of a day with that of the previous days in the same month. This helps to decide on the intensity values to be considered while formulating the regression equation. The decreasing trend shown by the line plots indicates that as we go behind in the number of the days in a month, there is very little correlation between the intensity values.

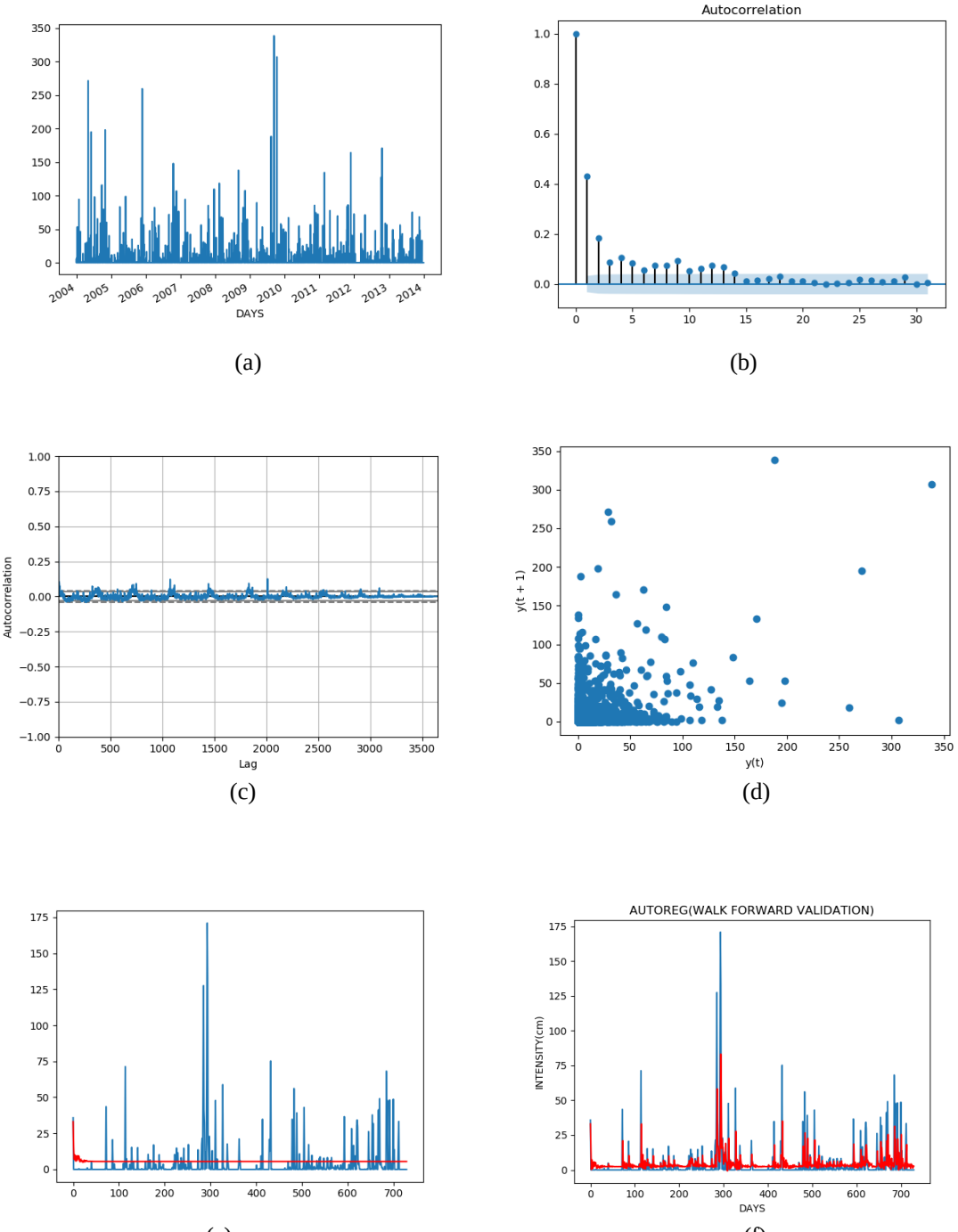
Figure 30 (c): is a complete autocorrelation plot which depicts the correlation maintained by the rainfall intensity value of any day with that of the any other day within a span of 10 years in the dataset. It gives an insight that the past yearly data shows limited effects on the future rainfall intensity values as far as the development of the rainfall prediction model is considered.

Figure 30 (d) is a basic lag plot which shows how the rainfall intensity value is changing from one day to the next immediate day in a year.

Figure 30 (e) shows the result of the simple Auto regression model developed without the inclusion of the Walk forward validation algorithm. As it can be seen in the figure the model makes an average prediction of the rainfall pattern when compared to the results of the model depicted in the FIG f .

Figure 30 (f) shows the results of the Auto regression model with the inclusion of the Walk forward validation algorithm. The model gives superior results when compared to the simple Autoregression

model for here the training of the model is done with all samples available in the dataset unlike the Simple Autoregression model where the training of the model is done with samples present just in the training set.



(a) (b)

(c) (d)

(e) (f)

Figure 30. (a) Series Plot (b) Monthly Auto Correlation Plot (c) Daily Auto Correlation Plot (d) Lag Plot (e) Prediction Using Auto Regression (f) Prediction Using Auto Regression Walk Forward Validation

Table 6 gives the correlation coefficient values between the rainfall intensity values of a day separately with that of the past 12 days. The correlation coefficient values are the main criteria based

on which it is decided, the intensity values how many past days should be considered for the formulation of the Autoregression equation.

Table 6. Correlation Chart for Auto Regression Model

	t-1	t		t-2	t		t-3	t
t-1	1	0.429966	t-2	1	0.18373	t-3	1	0.088745
t	0.42966	1	t	0.18373	1	t	0.088745	1
	t-4	t		t-5	t		t-6	t
t-4	1	0.106374	t-5	1	0.08528	t-6	1	0.054986
t	0.106374	1	t	0.08528	1	t	0.054986	1
	t-7	t		t-8	t		t-9	t
t-7	1	0.075243	t-8	1	0.073265	t-9	1	0.092135
t	0.075243	1	t	0.073265	1	t	0.092135	1
	t-10	t		t-11	t		t-12	t
t-10	1	0.05275	t-11	1	0.061504	t-12	1	0.07474
T	0.05275	1	T	0.061504	1	T	0.07474	1

Table 7 shows the predicted and actual values of rainfall intensity for both the models mentioned above are tabulated and it can be seen that the second model outperforms the first in terms of the RMSE value.

Table 7. Expected vs Predicted Rainfall Intensity for Auto Regression Model

AutoRegression			**AutoRegression (Walk Forward Validation)**	
Expected amount of Rainfall	**Predicted Amount of Rainfall**		**Expected amount of Rainfall**	**Predicted Amount of Rainfall**
35.8	33.0599		35.8	33.0599
0	15.7632		0	16.9382
0	7.03883		0	0.245693
0	10.175		0	7.28758
0	9.49442		0	5.80882
0	5.95194		0	0.933555
0	7.82142		0	4.91533
0	5.85771		0	2.37111
0	9.52433		0	5.79746
0	6.42905		0	2.51351
RMSE = 13.347			**RMSE=12.024**	

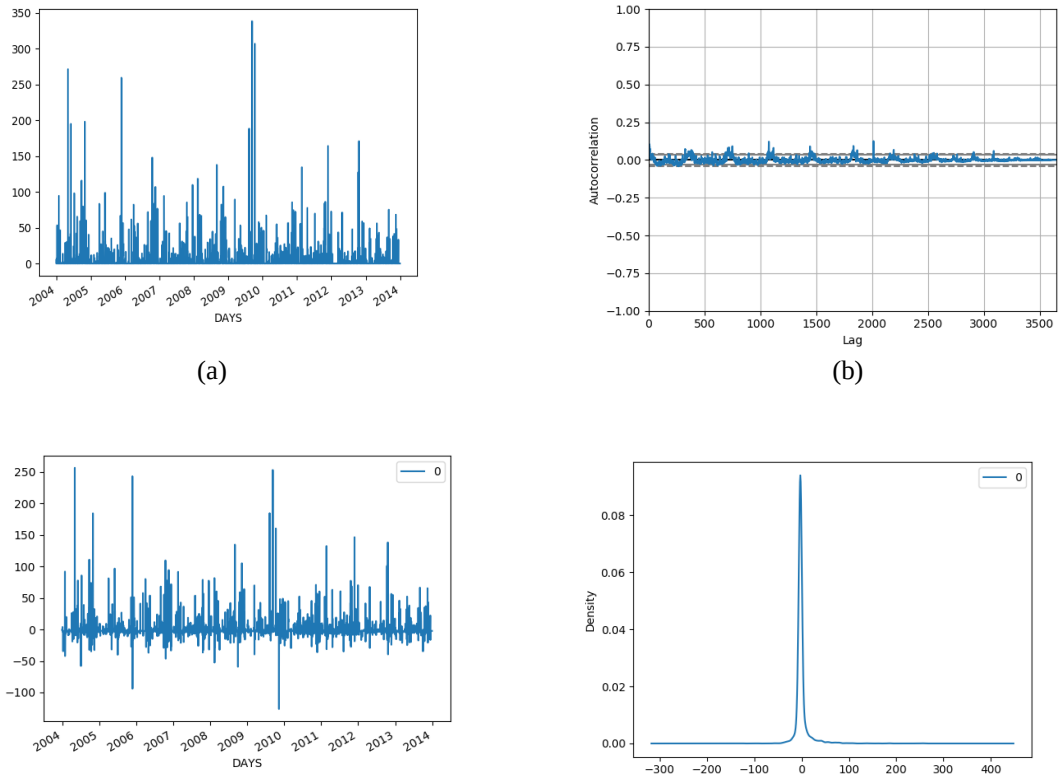
5.2.4 ARIMA

Figure 31 (a) shows the series plot of rainfall intensity values over the different days in a year from 2004 to 2013.It gives an overall idea about the distribution of rainfall intensity values in a year which can be used for the modelling of the regression equation

Figure 31 (b) is a complete autocorrelation plot which depicts the correlation maintained by the rainfall intensity value of any day with that of the any other day over a span of 10 years in the dataset. It gives an insight that the past yearly data shows limited effects on the future rainfall intensity values as far as the development of the rainfall prediction model is considered.

Figure 31 (c) shows the residual error plot which represents the error caused by the prediction model while predicting the daily rainfall intensity values in a year. It gives an idea about the efficiency of the model.

Figure 31 (d) represents the probability density function of the errors calculated during the prediction with which the general trend followed by the prediction errors like Gaussian, Rayleigh etc can be determined.



(a) (b)

(c) (d)

Figure 31. (a) Series Plot of ARIMA (b) Auto Correlation Plot (c) Basic Residual Error Yearly (d) Basic Residual Error Density Plot

Table 8 shows the results of Sensitivity analysis done on the ARIMA model based on the P,D and Q values. The best model with the least RMSE value has a P, D, Q value of 4, 0, and 2 respectively.

Table 8. ARIMA Grid Search Results [p,d,q] vs MSE

ARIMA-GRID SEARCH RESULTS			
ARIMA(0, 0, 0)	MSE=13.337	ARIMA(4, 0, 0)	MSE=12.050
ARIMA(0, 0, 1)	MSE=12.226	ARIMA(4, 0, 1)	MSE=11.992
ARIMA(0, 0, 2)	MSE=12.048	**ARIMA(4, 0, 2)**	**MSE=11.961**
ARIMA(0, 1, 0)	MSE=14.305	ARIMA(4, 1, 0)	MSE=12.898
ARIMA(0, 1, 1)	MSE=12.851	ARIMA(4, 1, 2)	MSE=12.013
ARIMA(0, 1, 2)	MSE=12.150	ARIMA(4, 2, 0)	MSE=15.814
ARIMA(0, 2, 0)	MSE=23.222	ARIMA(6, 0, 0)	MSE=12.050
ARIMA(0, 2, 1)	MSE=14.311	ARIMA(6, 0, 1)	MSE=12.077
ARIMA(1, 0, 0)	MSE=12.024	ARIMA(6, 0, 2)	MSE=11.999
ARIMA(1, 0, 1)	MSE=12.029	ARIMA(6, 1, 0)	MSE=12.493
ARIMA(1, 0, 2)	MSE=12.049	ARIMA(6, 1, 1)	MSE=12.050
ARIMA(1, 1, 0)	MSE=13.679	ARIMA(6, 1, 2)	MSE=12.082
ARIMA(1, 1, 1)	MSE=12.043	ARIMA(6, 2, 0)	MSE=15.192
ARIMA(1, 1, 2)	MSE=12.047	ARIMA(8, 0, 0)	MSE=11.996
ARIMA(1, 2, 0)	MSE=19.045	ARIMA(8, 1, 0)	MSE=12.299
ARIMA(2, 0, 0)	MSE=12.028	ARIMA(8, 1, 0)	MSE=12.299
ARIMA(2, 0, 1)	MSE=12.019	ARIMA(8, 1, 1)	MSE=11.997
ARIMA(2, 0, 2)	MSE=11.975	ARIMA(8, 1, 2)	MSE=12.034
ARIMA(2, 1, 0)	MSE=13.365	ARIMA(8, 2, 0)	MSE=14.125
ARIMA(2, 2, 0)	MSE=17.469	ARIMA(10, 1, 1)	MSE=12.009

Table 9 gives the details about the best ARIMA model obtained from the sensitivity analysis(ARIMA(4,0,2)) like the number of observation used for the model development ,Log likelihood value etc.

Table 10 and 11 shows the rainfall intensity values of the past days which are used for the formulation of the regression equation along with their respective coefficients in the equation. The ‘P’ value given in the table corresponding to each coefficient describes how much influence each of them has in deciding the predicted result. The lesser the P value , the more is the correlation of the parameter with the output.

Table 9. Basic ARIMA Model Results (1)

ARMA Model Results			
Dependent Variable	Rainfall	**No. of Observations**	3650
Model	ARMA(4,2)	**Log likelihood**	-15304.116
Method	Css-mle	**S.D of Innovations**	16.022
AIC	30624.233	**BIC**	30673.853
Sample	01-01-2004	**HQIC**	30641.904

Table 10. Basic ARIMA Model Results (2)

	Coeff	**Std.err**	**z**	**|P|>z**	**[0.025**	**0.975]**
const	5.51510	0.633	8.133	0.000	3.910	6.392
Ar.L1.RAINFALL	0.8612	0.163	5.269	0.000	0.541	1.181
Ar.L1.RAINFALL	0.1374	0.209	0.659	0.510	-0.271	0.546
Ar.L1.RAINFALL	-0.1598	0.065	-2.471	0.014	-0.287	-0.033
Ar.L1.RAINFALL	0.0643	0.018	3.551	0.000	0.029	0.100
Ar.L1.RAINFALL	-0.4375	0.163	-2.681	0.007	-0.757	-0.118
Ar.L1.RAINFALL	-0.3307	0.145	-2.278	0.023	-0.615	-0.046

Table 11. Basic ARIMA Model Results (3)

	Real	**Imaginary**	**Modulus**	**Frequency**
AR.1	1.1054	-0.0000j	1.1054	-0.0000
AR.2	-1.9625	-0.0000j	1.9625	-0.5000
AR.3	1.6718	-2.0926j	2.6784	-0.1427
AR.4	1.6718	+2.0926j	2.6784	0.1427
MA.1	1.1989	+0.0000j	1.1989	0.0000
MA.2	-2.5218	+0.0000j	2.5218	0.5000

Table 12 shows the Predicted and Expected values of rainfall intensity using the best ARIMA model.

Table 12. Expected vs Predicted Rainfall Intensity of ARIMA

ARIMA FOLDING FORECAST RESULTS	
EXPECTED	**PREDICTED**
35.8	33.2715
0	16.9727
0	0.456497
0	5.56906
0	5.38867
0	4.63656
0	4.28323
0	3.89451
0	3.62006
0	3.37882

Figure 32 shows a plot of the expected and predicted values using the best ARIMA model (4, 0, 2). It can be seen that the model (4, 0, 2) has managed to capture the general rainfall pattern better when compared to the previous basic Auto regression models.

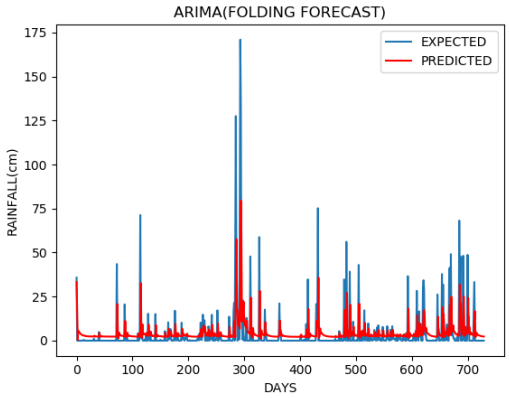
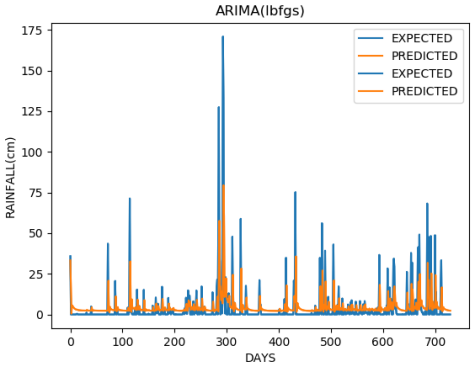


Figure 32. ARIMA Folding Forecast Prediction Plot

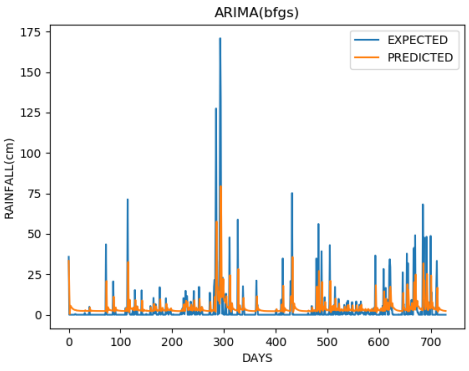
Figure 33 (a), (b), (c), (d), (e), (f) shows how predicted rainfall values are changing with respect to the different Solver functions used for the model described above.

Figure 33 (g) shows a bar graph which represents the RMSE values obtained for the different Solver functions. It can be seen that in terms of RMSE value all the Solver functions seems to showcase similar performance.

Figure 33 (h) shows a bar graph which gives a comparison between the Execution time of the different solver functions. Although all the Solver functions gave comparable results in terms of RMSE value, the 'NM' function produce the results with lesser execution time.



(a)



(b)

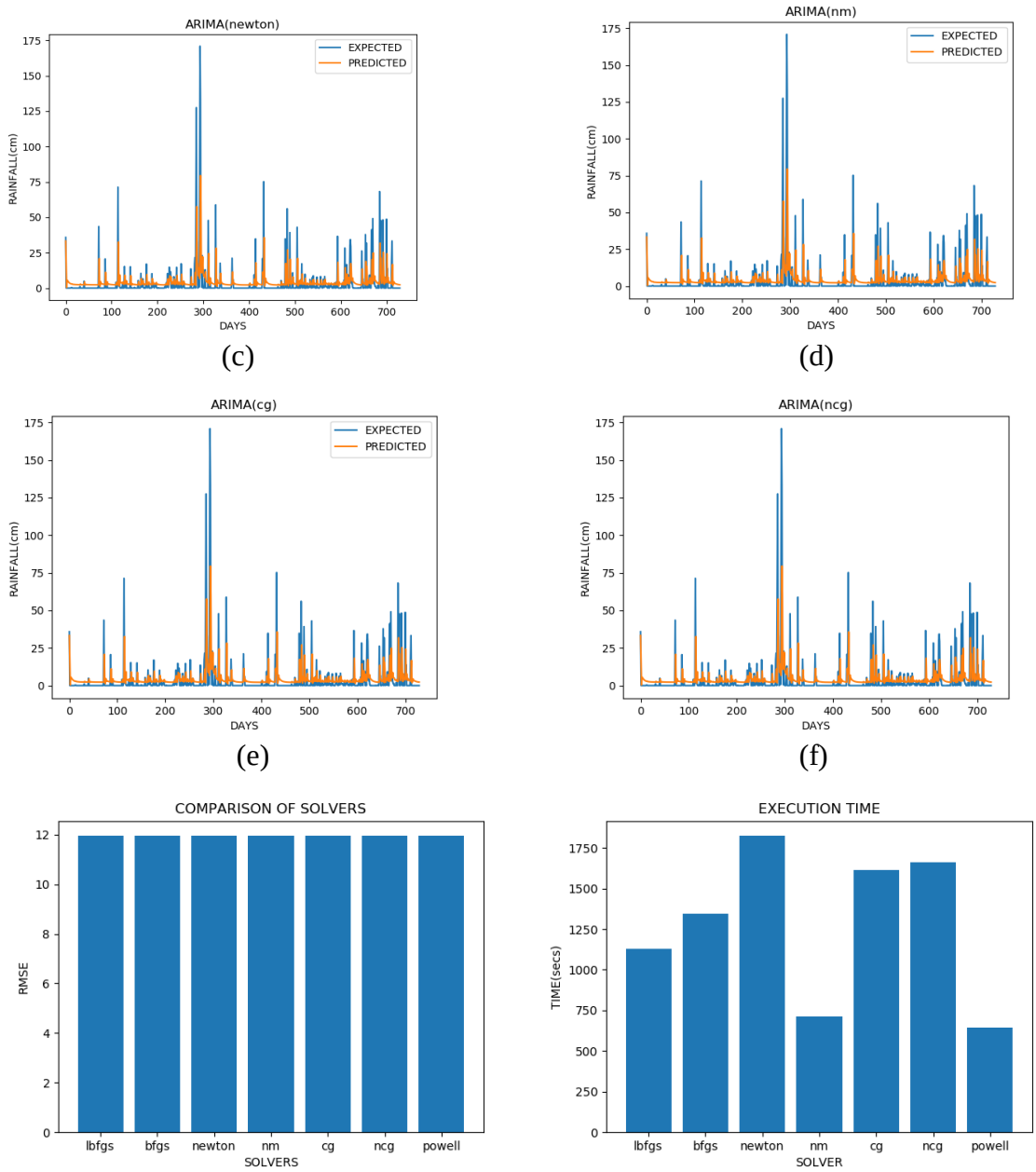


Figure 33. (a) ARIMA lbfgs (b) ARIMA bfgs (c) ARIMA newton (d) ARIMA nm (e) ARIMA cg (f) ARIMA ncg (g) Comparison of Solvers (h) Comparison of Execution time of each solver

Table 13 shows a comparison between Solver functions in terms of RMSE value and Execution time.

Table 13. RMSE and Execution Time vs Solvers

RMSE VS SOLVERS		**TIME VS SOLVERS**	
Lbfgs	11.96136	Lbfgs	1129.091
Bfgs	11.96155	Bfgs	1345.201
newton	11.96136	newton	1825.439

Nm	11.96074	Nm	710.3942
Cg	11.96151	Cg	1613.26
Ncg	11.96136	Ncg	1662.535
powell	11.96092	powell	646.4367

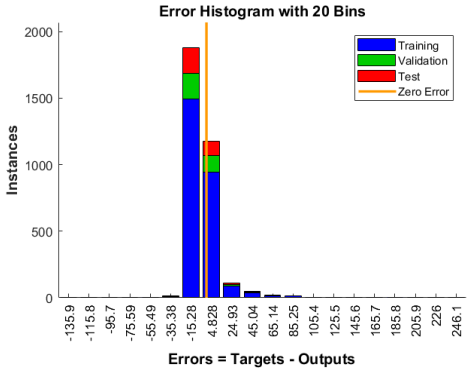
5.2.5 NON-LINEAR AUTO REGRESSIVE NEURAL NETWORK

As it can be seen in the table 14 the highlighted architectures gives the best results suggesting that the same can be chosen to get an optimised result for the Coonoor dataset. The best architecture with 10 hidden neurons and 12 delays (time steps) gives a RMSE of 9.659.The significant fluctuation in the RMSE values are mainly because of the fact that the prediction model tend to perform better when the past rainfall intensity values chosen for the development share a close relation with the future ones and vice-versa.

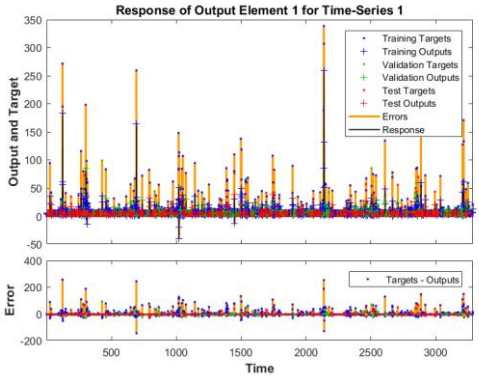
Table 14. Architecture of NAR Neural Network

Architecture [hidden neurons_delays(days)]	**RMSE**
10_1	13.81827
10_2	12.95932
10_3	**11.25532**
10_4	14.0556
10_5	26.86604
10_6	18.88689
10_7	27.40681
10_8	15.58281
10_9	20.07578
10_10	23.17773
10_11	16.44078
10_12	**9.659405**
10_13	22.3719
10_14	**10.48257**
10_27	13.06266
10_30	22.1836
10_31	12.75723

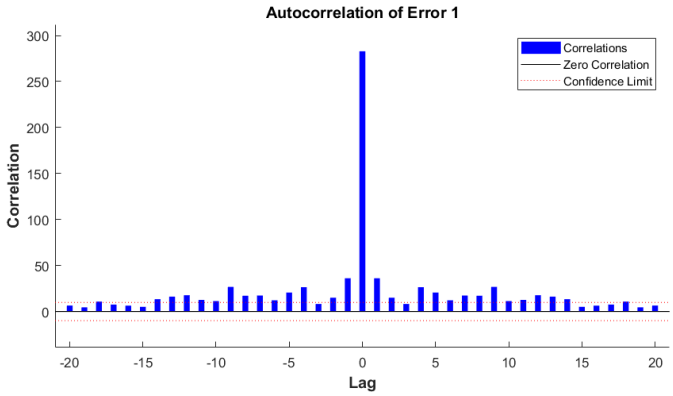
Figure 34 (a), 35(a), 36 (a) shows the Error histogram for the models chosen above. Similarly figure 34 (b), 35 (b), 36 (b) shows the Time series response of the training, validation and testing data of the Coonoor dataset. Figure 34 (c), 35 (c), 36 (c) represents the Error Autocorrelation plot of the dataset.



(a)

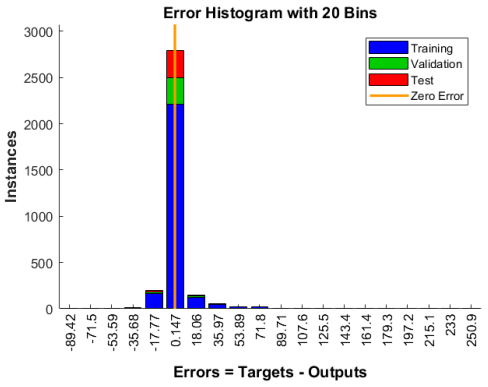


(b)

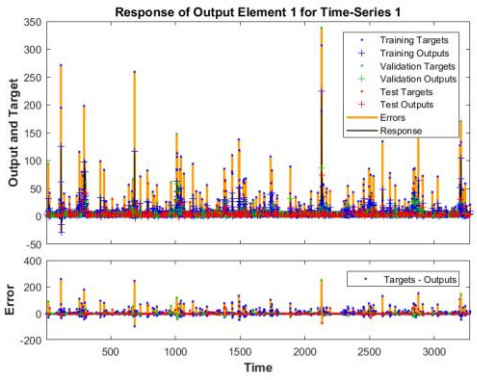


(c)

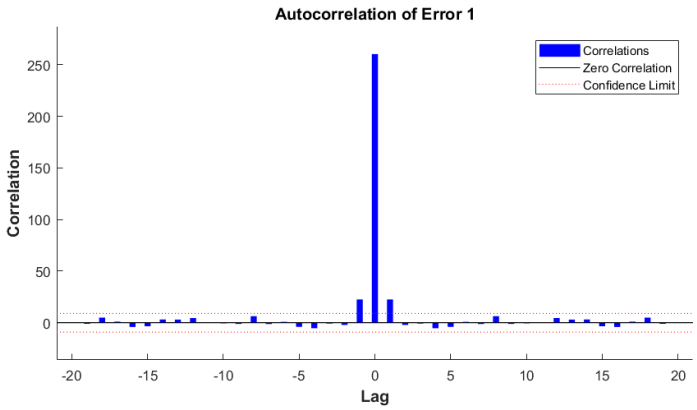
Figure 34. (a) Error Histogram of [10_12] (b) Time Series Response (c) Error Auto Correlation



(a)

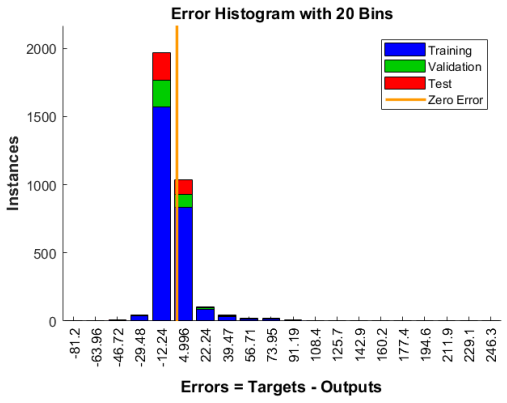


(b)

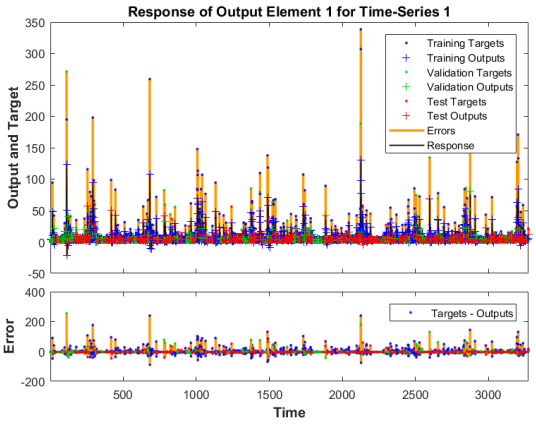


(c)

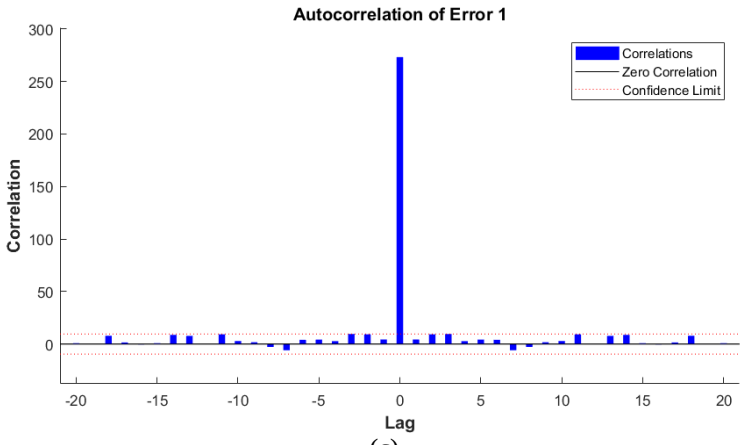
Figure 35. (a) Error Histogram of [10_3] (b) Time Series Response (c) Error Auto Correlation



(a)



(b)



(c)

Figure 36. (a) Error Histogram of [10_14] (b) Time Series Response (c) Error Auto Correlation

5.2.6 RECURRENT NEURAL NETWORK UNIVARIATE TIME SERIES ANALYSIS

Table 15 shows that the best architecture for the dataset in hand is [120, 50, 2]. It indicates that to forecast the rainfall for the next day with nominal error, the rainfall intensities for the past 120 days should be considered.

Table 15. Comparison of RNN architectures using RMSE and Relative Error

Architecture [number of time steps, number of neurons, number of layers]	**RMSE**	**Relative Error**
[30, 50, 2] (A)	11.08081	0.162
[30, 50, 3] (B)	10.98472	0.161
[60, 50, 2] (C)	11.23886	0.165
[60, 50, 3] (D)	11.09363	0.163
[120, 50, 2] (E)	**10.98221**	**0.161**
[120, 50, 3] (F)	11.0389	0.162
[30, 100, 2] (G)	11.3939	0.167
[30, 100, 3] (H)	11.07735	0.162
[60, 100, 2] (I)	11.1436	0.163
[60, 100, 3] (J)	11.00452	0.161
[120, 100, 2] (K)	11.07338	0.162
[120, 100, 3] (L)	11.21365	0.164

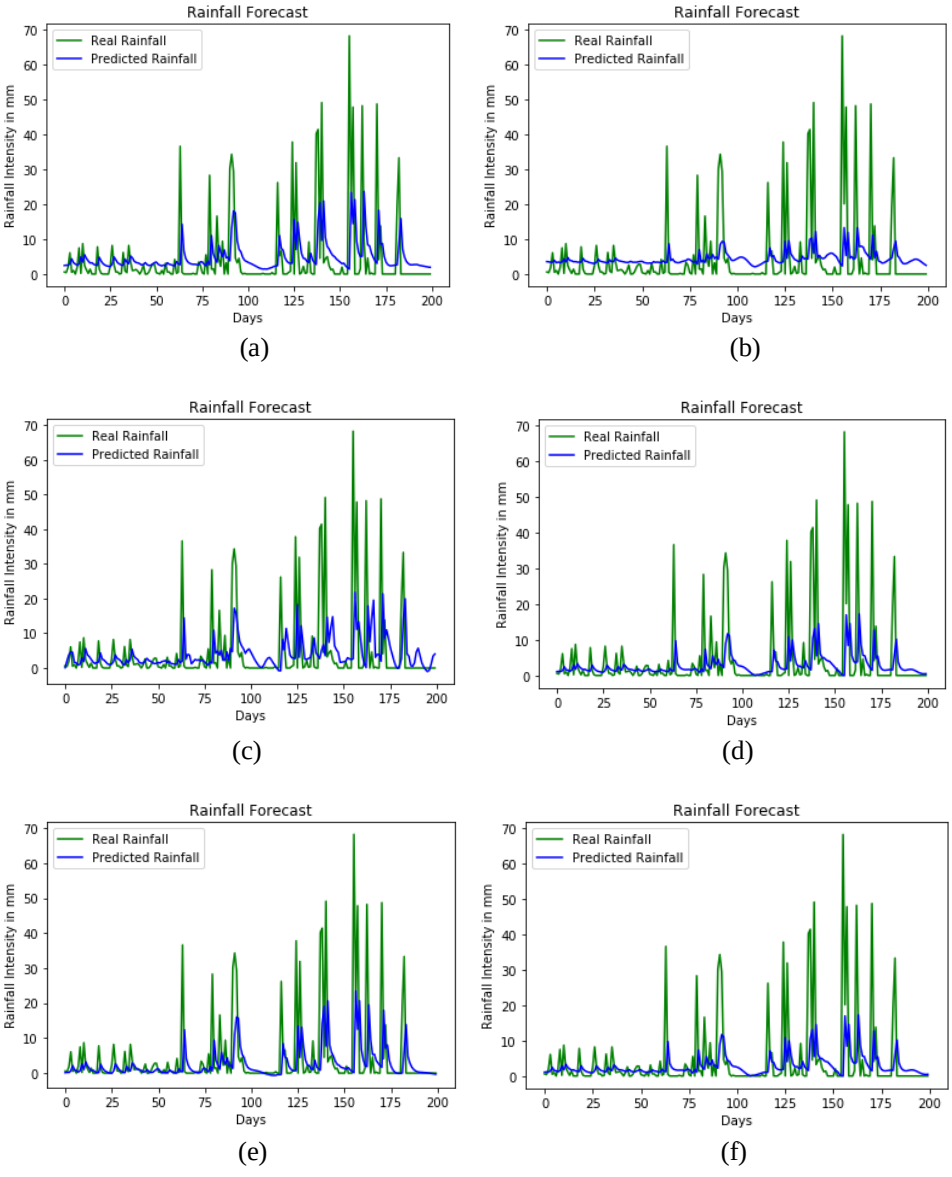
Table 16 shows the expected versus predicted rainfall intensities of all the architectures.

Table 16. Expected versus Predicted Rainfall Intensity of RNN

Y_real	**Y_pred**											
	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**
0.6	2.449	3.564	0.111	1.084	2.447	2.636	0.107	1.212	3.329	1.657	1.492	2.660
0.4	2.545	3.509	0.215	1.139	2.501	2.703	2.008	1.251	3.328	1.327	1.546	2.806
1.9	2.511	3.357	0.184	1.101	2.456	2.651	3.756	1.180	3.328	1.101	1.546	2.910
6.1	2.930	3.430	0.633	1.401	2.848	3.015	4.593	1.538	3.346	1.311	1.963	3.349
0.5	4.360	3.912	2.122	2.410	4.219	4.217	4.525	2.336	3.415	2.406	3.090	4.394
1	3.233	3.219	0.966	1.567	3.173	3.183	1.655	1.661	3.385	2.200	2.238	3.560
0	2.868	3.460	0.751	1.453	2.849	3.160	1.437	2.196	3.355	2.218	2.296	3.860
2	2.502	3.234	0.304	1.188	2.436	2.890	0.882	1.367	3.335	1.983	1.867	3.308
7.5	2.962	3.473	0.711	1.520	2.832	3.304	1.068	1.867	3.355	2.195	2.045	3.721
0.2	4.862	4.262	2.577	2.842	4.634	4.837	2.921	2.483	3.445	3.329	3.291	4.937
8.7	3.433	3.461	1.026	1.733	3.289	3.362	1.822	1.694	3.403	2.686	2.081	3.501
2.4	5.535	4.719	3.217	3.316	5.184	5.321	5.615	3.771	3.469	3.770	4.053	5.948
1.2	4.463	3.912	2.026	2.461	4.184	4.159	3.804	1.953	3.451	3.277	3.066	4.230
0.2	3.754	3.816	1.328	2.015	3.402	3.685	3.290	2.841	3.406	2.736	2.800	4.472
1.4	3.149	3.525	0.729	1.640	2.798	3.247	2.536	1.174	3.369	2.103	2.375	3.617
0	3.246	3.572	0.825	1.743	2.892	3.348	1.733	1.962	3.370	1.979	2.322	3.778
0	2.782	3.414	0.393	1.445	2.555	2.900	1.297	0.615	3.355	1.658	1.869	3.020

0	2.606	3.421	0.224	1.330	2.429	2.700	1.543	1.468	3.338	1.542	1.744	3.007
7.8	2.493	3.411	0.137	1.254	2.399	2.556	1.741	0.693	3.327	1.523	1.654	2.731
1.4	4.797	4.474	2.548	2.880	4.655	4.506	4.331	2.857	3.428	3.111	3.449	4.434

Figure 37 (a)- (l) shows the plot between Predicted and Real rainfall for 200 days using the Model (A)- Model(L) mentioned in the Table 15 respectively. The predicted graph of model (E) has the highest similarity to the actual rainfall graph as expected from the low RMSE value of the model in comparison with the other prediction models.



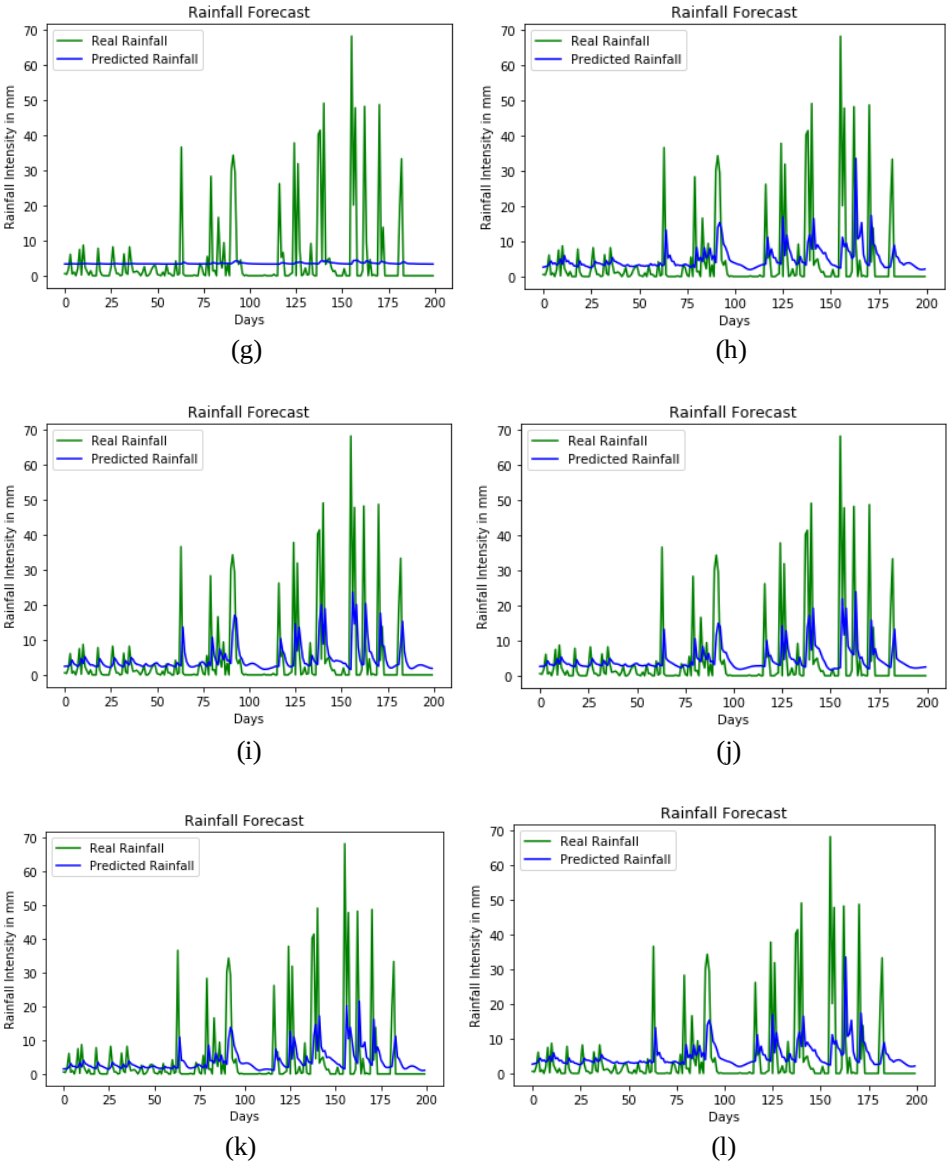


FIG 37 (a)[30,50,2] (b)[30,50,3] (c)[60,50,2] (d)[60,50,3] (e)[120,50,2] (f)[120,50,3] (g)[30,100,2] (h)[30,100,3] (i)[60,100,2] (j)[60,100,3] (k)[120,100,2] (l)[120,100,3]

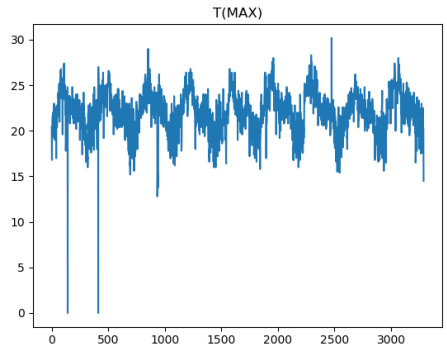
5.3 MULTIVARIATE TIME SERIES ANALYSIS

Multivariate dataset includes rainfall intensity information along with temperature, relative humidity, wind speed, wind direction, cloud coverage and nearby rain gauge stations. The following techniques are used to forecast rainfall using Multivariate dataset:

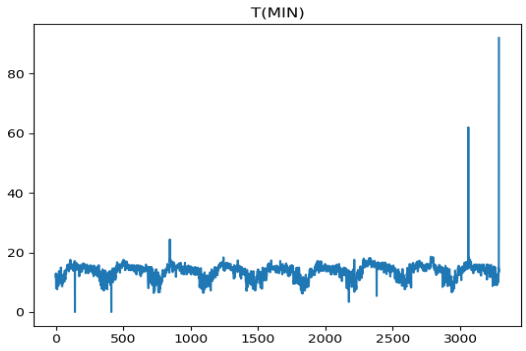
1. Non-linear AutoRegressive with Exogenous Inputs
2. Regression Techniques
3. Artificial Neural Network
4. Recurrent Neural Network

5.3.1 VISUALIZATION OF MULTIVARIATE DATA

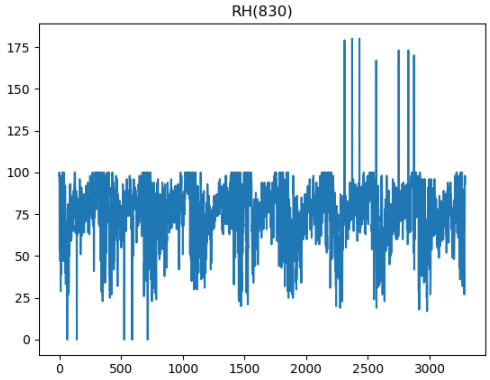
Figure 38 (a) and 38 (b) shows the general pattern followed by the temperature over different days in 9 years in their respective units. Figure 38 (c) and 38 (d) shows how the relative humidity change over different days in 9 years. Figure 38 (e) show the variation of wind speed over different days in 9 years. Figure 38 (f) and 38 (g) shows the pattern of cloud coverage over 9 years. Figure 38 (h), 38 (i) and 38 (j) shows the rainfall pattern over different days in 9 years.



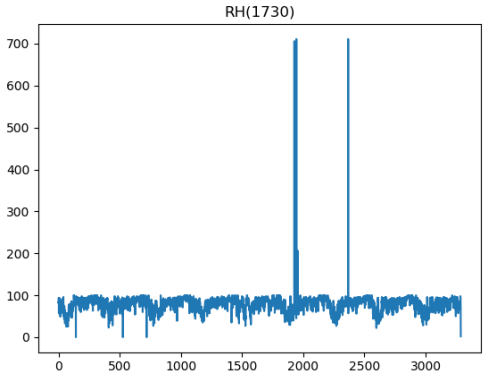
(a)



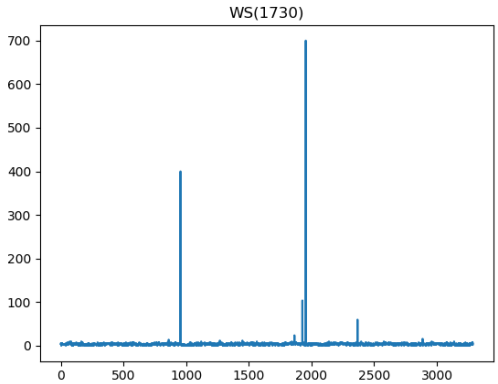
(b)



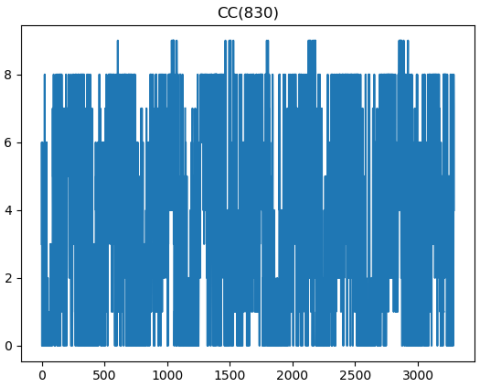
(c)



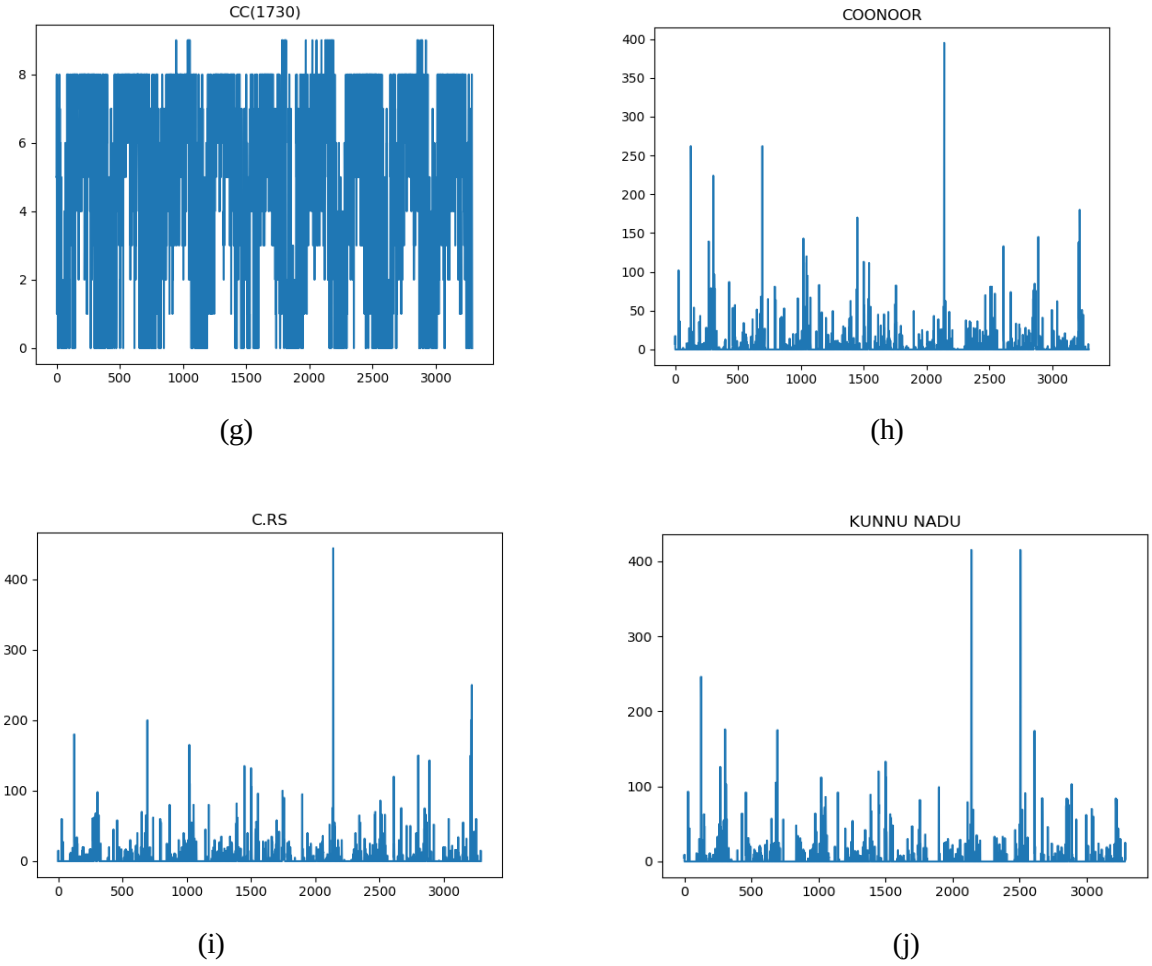
(d)



(e)



(f)



(g) (h)

(i) (j)

Figure 38. Visualisation of Multivariate Analysis: (a) Temp (max) (b) Temp (min) (c) RH (max) (d) RH (min) (e) WS (max) (f) cloud coverage (max) (g) cloud coverage (min) (h) Coonoor rainfall (i) Coonoor railway station rainfall (j) kunnu nadu rainfall

5.3.2 NARX NETWORK

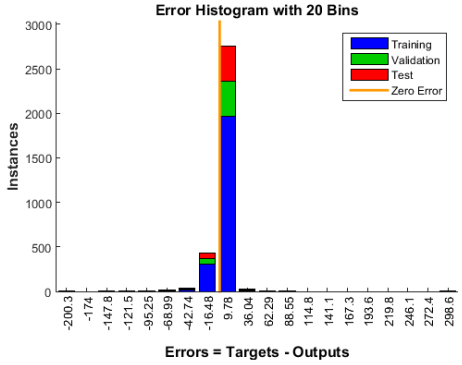
As it can be seen in table 17 the highlighted architectures gives the best results suggesting that the same can be chosen to get an optimised result for the Coonoor dataset. Table 17 shows that the best architecture of NARX is with 10 hidden neurons and 10 delays (time steps). It can be clearly seen from table 14 that NARX has less RMSE value as compared to NAR architecture. Also, it can be noted that for an efficient forecasting model Multivariate time series analysis provides better results than Univariate time series analysis.

Table 17. Architecture of NARX Neural Network

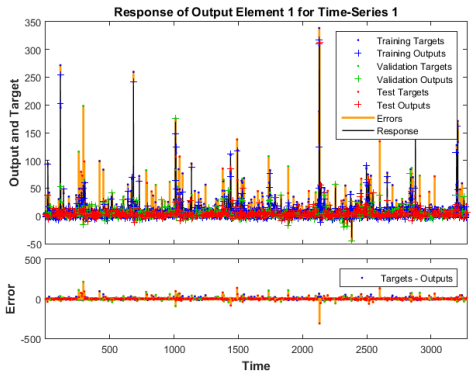
Architecture	**RMSE**
10_1	16.37803
10_2	**12.94967**
10_3	16.28038
10_4	15.46977

10_5	16.29680
10_6	18.35886
10_7	17.24221
10_8	16.82197
10_9	18.12545
10_10	**7.69252**
10_11	16.99664
10_12	**11.56524**
10_13	14.03552
10_14	14.66993
10_27	15.66056
10_30	20.52900
10_31	17.32575

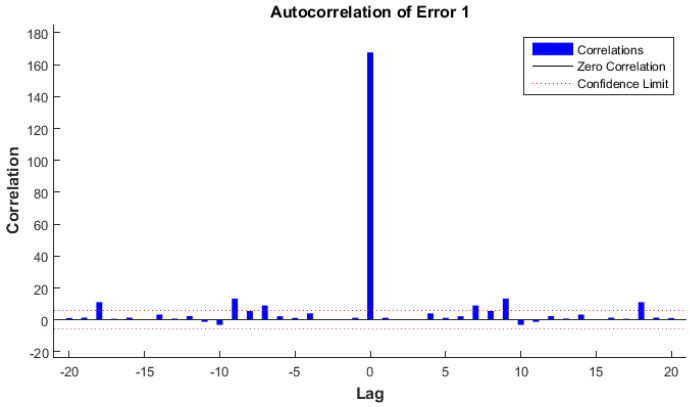
Figure 39 (a), 40 (a), 41 (a) shows the Error histogram for the models chosen above. Similarly figure 39 (b), 40 (b), 41 (b) shows the time series response of the training, validation and testing data of the Coonoor dataset. Figure 39 (c), 40 (c), 41 (c) represents the Error Autocorrelation plot of the dataset. Figure 39 (d), 40 (d), 41 (d) represents the correlation between the Inputs and the prediction error.



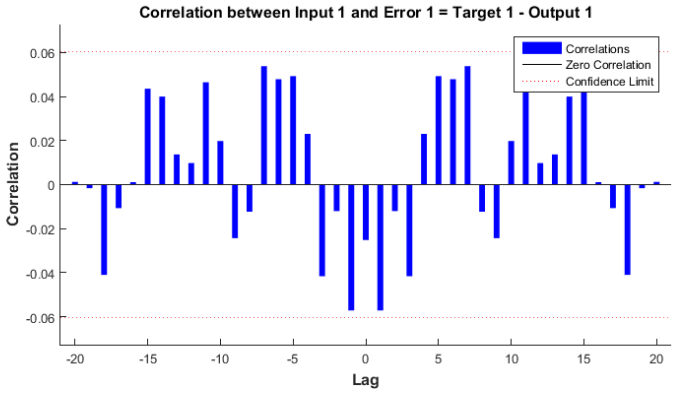
(a)



(b)

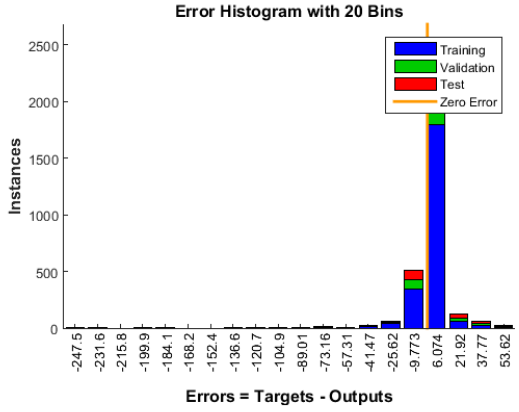


(c)

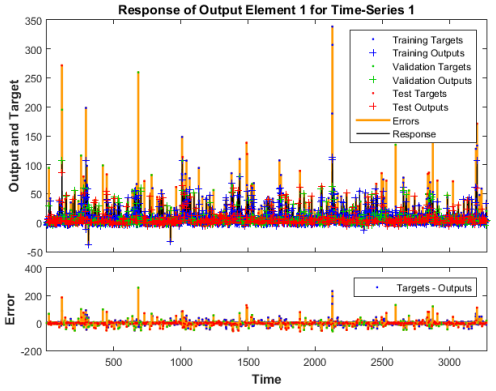


(d)

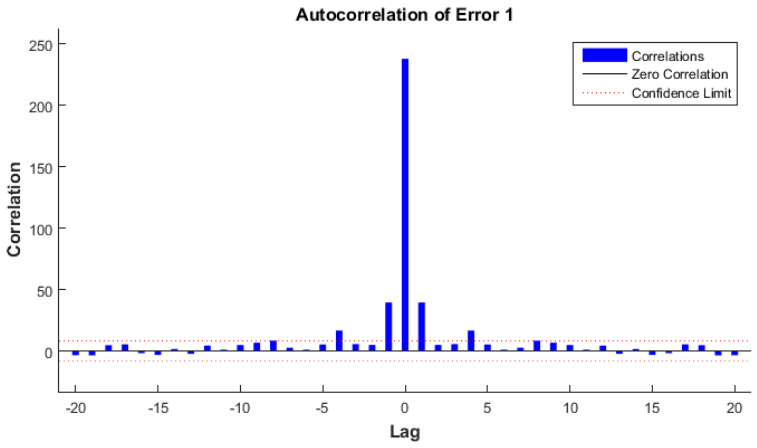
Figure 39. (a) Error Histogram of [10_10] (b) Time Series Response (c) Error Auto Correlation (d) Input-Error Correlation



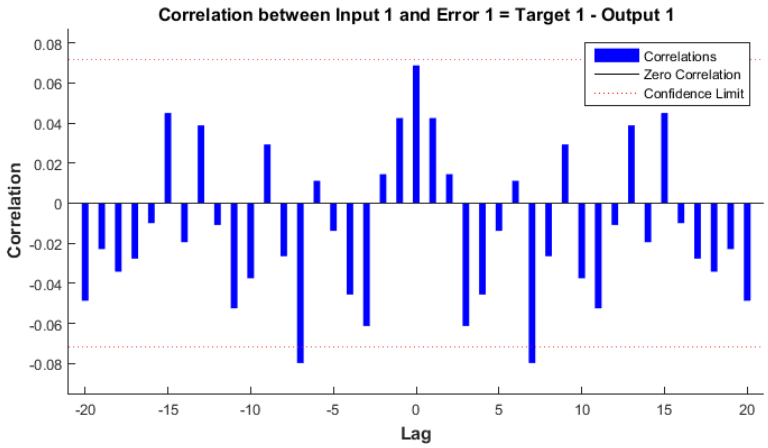
(a)



(b)

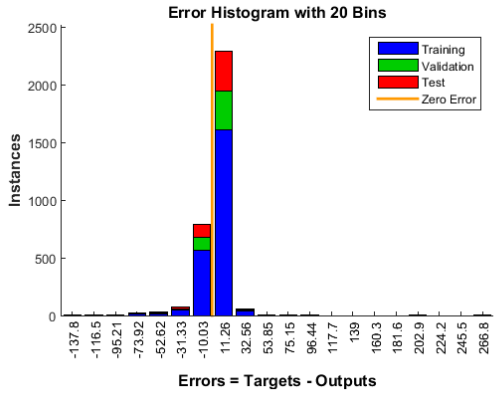


(c)

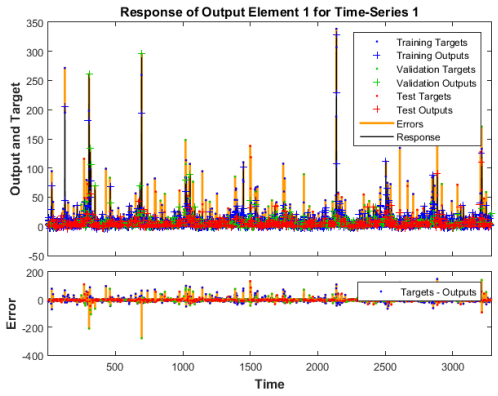


(d)

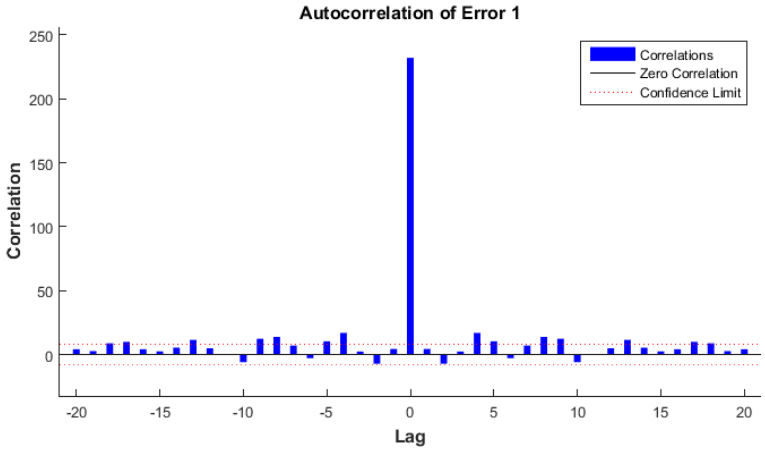
Figure 40. (a) Error Histogram of [10_2] (b) Time Series Response (c) Error Auto Correlation (d) Input-Error Correlation



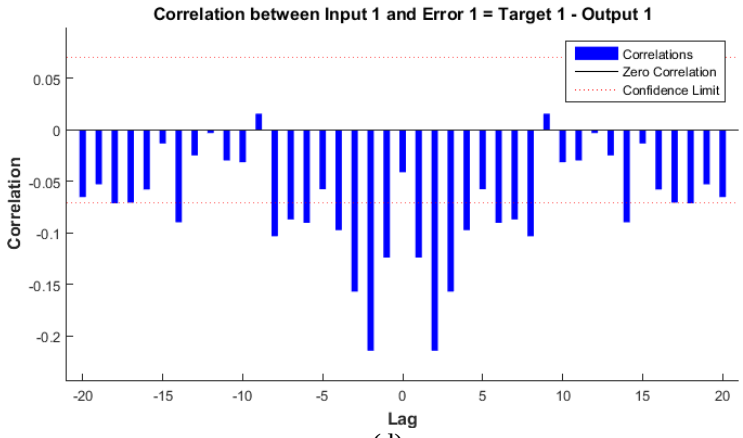
(a)



(b)



(c)



(d)

Figure 41. (a) Error Histogram of [10_12] (b) Time Series Response (c) Error Auto Correlation (d) Input-Error Correlation

5.3.3 Regression Techniques

Table 18 shows the comparison based on r-square and adjusted r-square values of SVR, DT and RF indicating that RF is the best model than the rest of the models.

Table 18. Comparison of SVR, RF and DT techniques

Forecasting Model	R-Square	Adjusted R-square
SVR	0.814	0.806
DT	0.904	0.900
RF	0.981	0.980

Model 1D represents the Decision Tree technique with mean absolute error as the evaluation criterion whereas Model 2D represents the Decision Tree based on the mean square error. Model 1R, 2R and 3R represents the Random Forests technique using number of estimators as 10, 100 and 300 respectively indicating that as the number of estimators' increases, RMSE value decreases. Model 1S and 2S represents the Support Vector Regression Model based on Polynomial and Radial Basis kernel function respectively.

Table 19 shows the expected vs predicted rainfall based on all the regression techniques and the RMSE and relative error has been tabulated in table 19.

As can be seen in the table 20 RF with 300 number of estimators (Model 3R) performs best for the dataset at hand suggesting that forecasting of rainfall is best with Random Forest. Though, XGboost outperforms in terms of execution speed but the RMSE compared to RF is not so good.

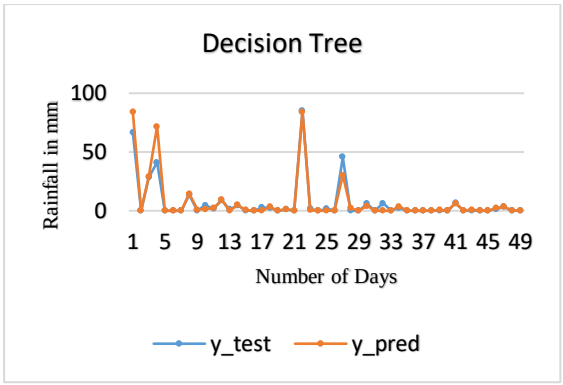
Table 19. Comparison of SVR, RF, DT and XGBoost techniques based on Expected vs Predicted

	Actual Rainfall	Predicted Rainfall (SVR)	Predicted Rainfall (DT)	Predicted Rainfall (RF)	Predicted Rainfall (XGBoost)
Day 1	66.7	50.9822	84.3	75.136	48.8
Day 2	0	1.10645	0	0.0026	0
Day 3	28.4	21.1975	29.3	24.463	16.3
Day 4	41.1	55.5796	71.8	63.354	56.5
Day 5	0	-0.48903	0	0.031	0
Day 6	0	1.07324	0	0	0
Day 7	0	0.5702	0	0.007	0
Day 8	13.4	8.8994	14.6	6.514	0
Day 9	0	4.7177	0.8	6.246	0
Day 10	4.6	5.9811	1.5	2.94	0
Day 11	1.6	2.5014	2.3	2.14467	0.4
Day 12	9	7.1762	9.4	7.338	5
Day 13	1	0.6504	0	0.1373	0
Day 14	4.8	5.8424	5	5.725	3.6
Day 15	0	0.6174	0.7	1.148	0
Day 16	0	-1.1598	0	0	0
Day 17	2.8	0.69156	0	0.1253	0
Day 18	2.5	5.32732	3.2	5.6403	2.6
Day 19	0	-0.32794	0	0.2346	0
Day 20	1	2.07851	1.4	1.2413	0

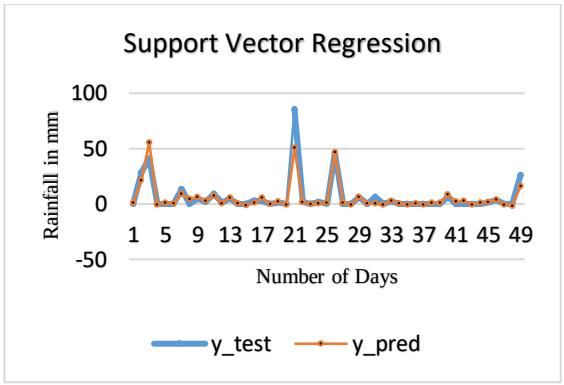
Table 20. Overall Relative and Root Mean Square Error

	Relative Error	RMSE
Model 1D	0.0375	6.410
Model 2D	0.040	6.915
Model 1R	0.032	5.556
Model 2R	0.030	5.228
Model 3R	0.029	5.085
Model 1S	0.048	8.290
Model 2S	0.046	7.422
XGBoost	0.050	8.361

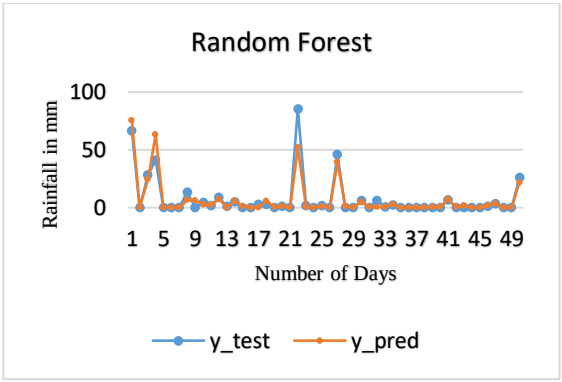
Figure 42 (a), (b) and (c) shows the best predicted and expected values for Decision trees, SVR and Random forest using most optimum hyperparameters such as Decision trees with MSE as the evaluation criteria, Random forest with the number of estimators as 300 and Support Vector machine with the RBF kernel function.



a)



b)



c)

FIG 42 (a) Expected vs Predicted rainfall using Decision trees (b)Expected vs Predicted rainfall using Support Vector Regression (c)Expected vs Predicted rainfall using Random forest.

5.3.4 Artificial Neural Network Evaluation of forecasting models

Table 21 shows the performance comparison of different ANN architectures in terms of RMSE values where the [2,12] architecture outperforms the other tested architectures with an RMSE value of 5.321. Table 22 shows the real and predicted rainfall intensity of various forecasting models.

Table 21. Architecture of Artificial Neural Network

Architecture [hidden layers, hidden nodes]	RMSE	Loss
[2.12]	5.321	0.0631
[2,13]	5.507	0.0619
[2,14]	5.701	0.0531
[2,15]	6.043	0.0603
[2,16]	5.872	0.0569
[3,12]	5.708	0.0622

[3,13]	6.773	0.0448
[3,14]	7.424	0.0571
[3,15]	7.339	0.0613
[3,16]	6.589	0.0352

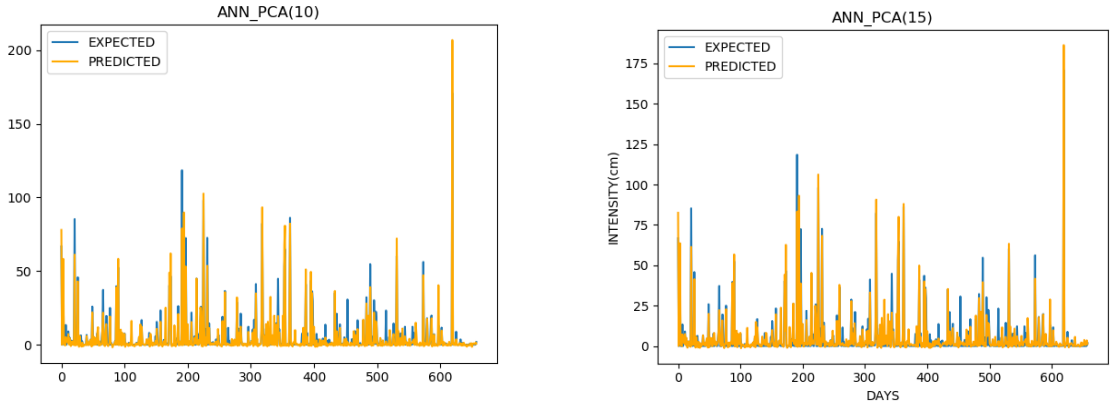
Table 22. Expected vs Predicted Rainfall Intensity of different architecture

Y_test	**Y_preda**									
	[2,12]	**[2,13]**	**[2,14]**	**[2,15]**	**[2,16]**	**[3,12]**	**[3,13]**	**[3,14]**	**[3,15]**	**[3,16]**
66.7	89.38	76.54	87.68	93.47	83.08	89.45	52.34	101.25	99.10	89.82
0	-1.24	-0.83	-0.93	-0.177	-0.33	-1.26	0.54	-0.78	-0.39	0.24
28.4	27.51	23.36	24.56	26.93	22.22	23.48	34.61	24.49	21.21	24.45
41.1	65.58	71.05	75.6	79.49	74.82	83.12	82.83	85.58	82.10	82.66
0	-0.91	-1.34	-0.75	-0.42	-0.56	-1.28	0.55	-1.10	-0.27	0.17
0	-0.88	-0.95	-0.89	0.012	-0.17	-0.97	0.55	-0.49	-0.06	0.90
0	-1.67	-1.09	-1.42	-0.71	-1.03	-1.68	0.51	-1.11	-0.72	0.02
13.4	6.95	3.57	4.51	4.40	3.67	2.94	5.23	3.83	4.10	5.05
0	3.47	1.43	2.194	2.20	1.46	0.80	2.83	1.67	1.72	2.32
4.6	3.11	2.08	3.672	3.939	3.32	2.84	3.02	2.75	2.68	3.63
1.6	2.69	0.56	1.463	1.959	1.73	0.45	1.80	1.10	1.70	3.21
9	8.74	6.35	7.882	8.570	8.70	6.86	7.01	7.49	8.34	8.95
1	0.27	-0.20	0.390	0.699	-0.09	-0.61	0.70	-0.16	0.28	0.88
4.8	8.36	9.51	9.466	8.234	8.31	6.00	5.39	9.26	2.74	9.57
0	1.00	0.85	1.673	2.051	0.90	-0.03	0.56	0.36	0.73	2.45
0	0.17	-1.15	-0.66	-0.66	-1.27	-1.26	0.52	-1.10	-0.89	-0.02
2.8	0.33	-0.37	0.370	0.301	-0.32	-0.41	1.02	0.11	0.38	1.15
2.5	10.89	7.31	8.255	9.002	9.86	6.46	11.57	7.78	7.99	10.19
0	-0.21	-1.08	-0.841	-0.09	-0.74	-1.04	0.92	-0.93	-0.28	0.28
1	1.91	0.38	1.091	1.15	0.72	-0.003	2.09	0.39	0.53	1.69

Figure 43 shows the Artificial Neural Network model of [2, 12] architecture using Principal Components Analysis feature extraction (Dimensionality Reduction) technique to remove the least correlated inputs from the dataset. Figure 43 (a) shows the expected and predicted rainfall of ANN architecture using 10 features whereas figure 43(b) shows the expected and predicted rainfall of ANN architecture using 15 features.

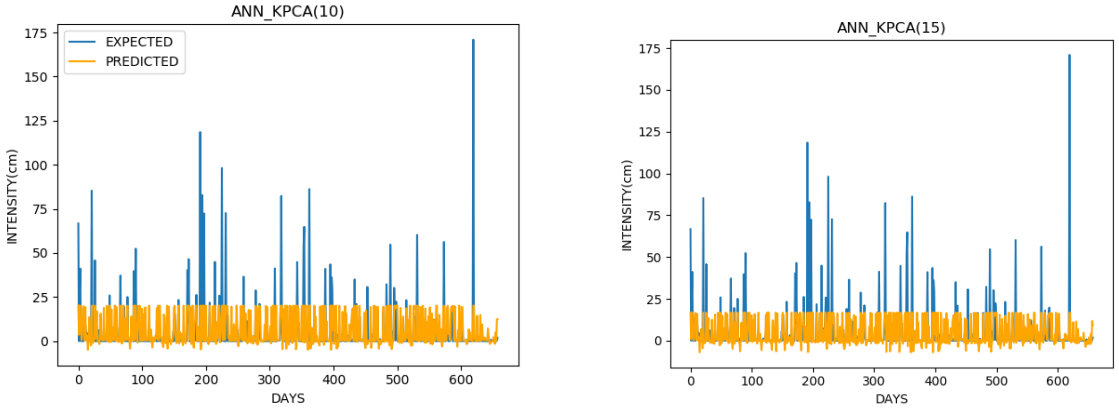
Figure 44 shows the ANN model of [2,12] architecture using extended PCA known as kernel PCA to reduce the features. Figure 44 (a) shows the expected and predicted rainfall of ANN architecture using 10 features and figure 44 (b) shows the k-PCA using 15 features. Figure 45 (a) and 45 (b) shows the dropout regularisation with 10% and 20% dropout of neurons in every training stage for 15 input feature architecture using PCA.

Figure 46 (a) and 46 (b) shows the dropout regularisation with 10% and 20% dropout of neurons in every training stage for 15 input feature architecture using k-PCA.



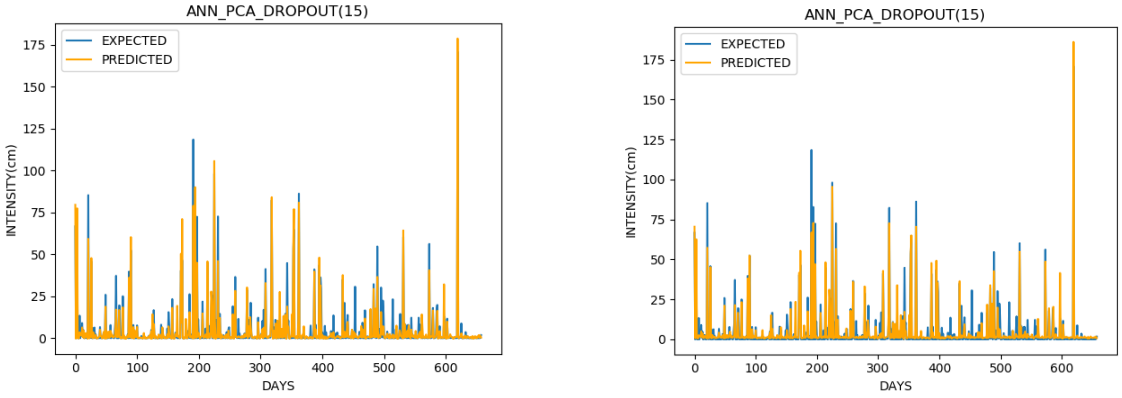
(a) (b)

Figure 43. (a) ANN_PCA (10 features) (b) ANN_PCA (15 features)



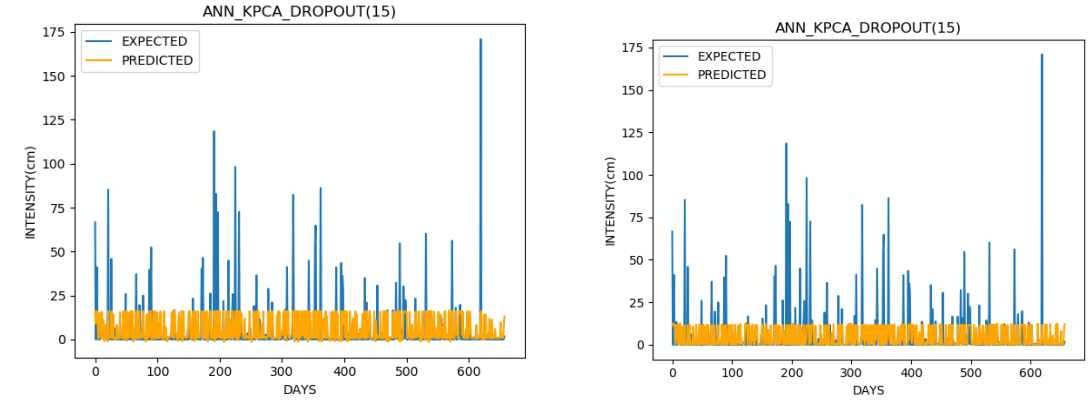
(a) (b)

Figure 44. (a) ANN_k-PCA (10 features) (b) ANN_k-PCA (15 features)



(a) (b)

Figure 45. (a) ANN_PCA Dropout (0.1) ((b) ANN_PCA Dropout (0.2)



(a) (b)

Figure 46. (a) ANN_k-PCA Dropout (0.1) (b) ANN_k-PCA Dropout (0.2)

Table 23 shows the Expected versus Predicted Rainfall intensity for ANN architecture with PCA and k-PCA feature extraction techniques. It can be clearly seen the ANN architecture [2,12] with PCA feature extraction technique gives the best result among all other architecture to forecast rainfall.

Table 23. PCA versus k-PCA ANN Architecture

Real Rainfall	**Predicted Rainfall**	**Predicted Rainfall**	**Predicted Rainfall**	**Predicted Rainfall**
	ANN_PCA (10)	**ANN_PCA (15)**	**ANN_KPCA (10)**	**ANN_KCPA (15)**
66.7	81.32	82.52	16.79	13.66
0	-0.24	0.03	0.73	9.75
28.4	20.18	22.59	16.79	13.66
41.1	60.47	63.68	16.79	13.66
0	0.38	1.50	3.26	10.24
0	0.31	0.29	0.43	-0.15
0	-0.61	-0.12	0.26	9.39
13.4	4.00	7.25	15.09	13.13
0	2.00	3.36	16.67	13.31
4.6	1.58	2.51	14.89	12.01
1.6	1.28	1.68	0.54	-0.86
9	6.99	7.41	16.03	13.30
1	0.17	0.69	0.095	-0.41
4.8	4.22	5.15	2.39	9.81
0	0.71	0.62	0.196	13.62
0	-1.18	-1.04	-6.99	2.96
2.8	0.09	1.181	0.54	-0.87
2.5	2.62	2.91	0.91	10.77
0	-0.06	0.34	-0.42	0.004
1	0.59	2.44	0.54	-0.82
RMSE	**4.929**	**4.788**	**13.125**	**12.256**

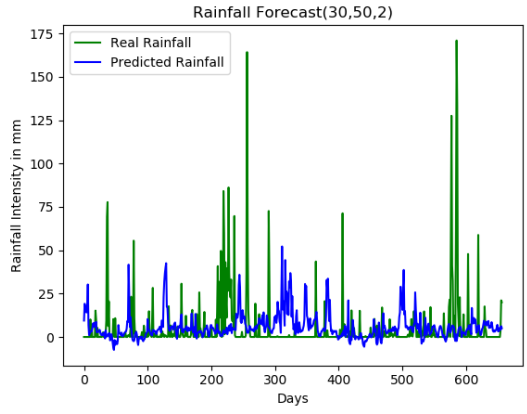
5.3.5 Recurrent Neural Network Multivariate Time Series Analysis

Table 24 shows the RMSE values of all the RNN architectures used for multivariate analysis. It is noted that it has very poor RMSE value than the Univariate RNN architectures, hence, for the dataset at hand.

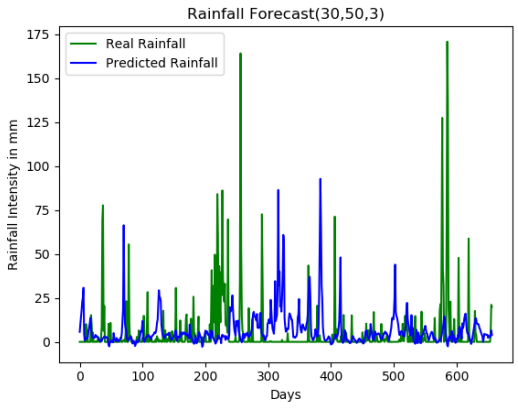
Table 24. RNN Architecture based on RNN

Architecture [number of time steps, number of neurons, number of layers]	**RMSE**
[30, 50, 2]	18.511
[30, 50, 3]	19.721
[60, 50, 2]	16.101
[60, 50, 3]	17.115
[120, 50, 2]	16.029
[120, 50, 3]	15.863
[30,100 2]	19.705
[30, 100, 3]	18.584
[60, 100, 2]	16.669
[60, 100, 3]	**15.670**
[120, 100, 2]	16.687
[120, 100, 3]	16.734

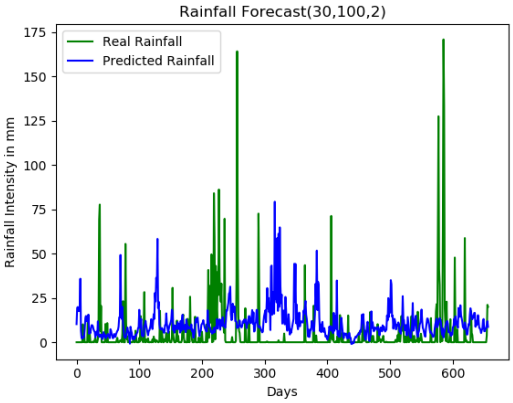
Figure 47 shows the real and predicted rainfall using Recurrent Neural Network (LSTM) with various forecasting models and time stamps. The architectures are modelled in the form of [number of time steps, number of neurons, number of layers].



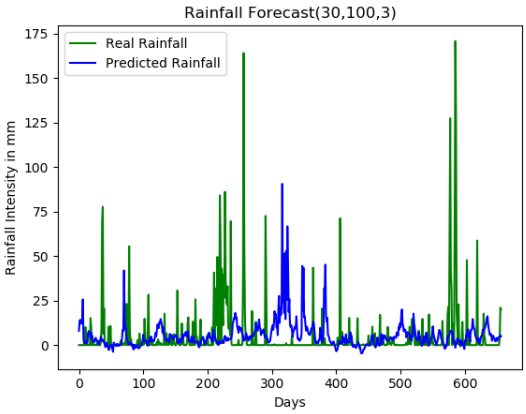
(a)



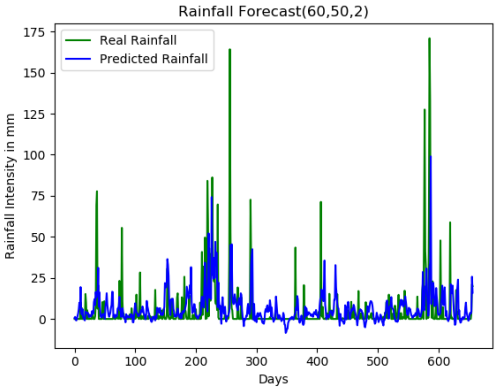
(b)


Rainfall Forecast(30,100,2)
Real Rainfall
Predicted Rainfall
Rainfall Intensity in mm
Days

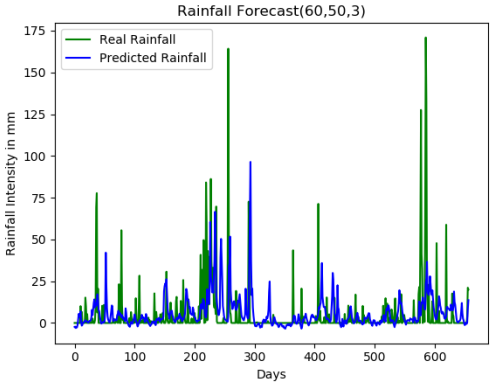
(c)


Rainfall Forecast(30,100,3)
Real Rainfall
Predicted Rainfall
Rainfall Intensity in mm
Days

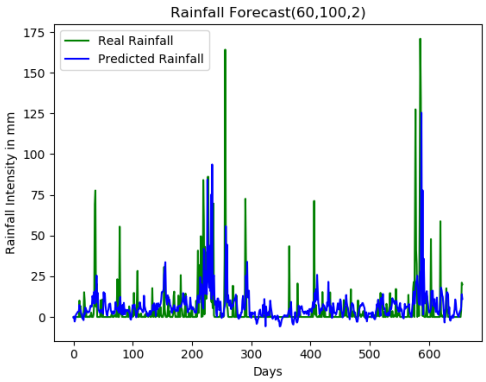
(d)


Rainfall Forecast(60,50,2)
Real Rainfall
Predicted Rainfall
Rainfall Intensity in mm
Days

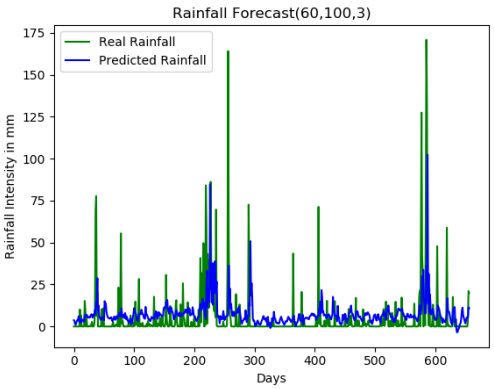
(e)


Rainfall Forecast(60,50,3)
Real Rainfall
Predicted Rainfall
Rainfall Intensity in mm
Days

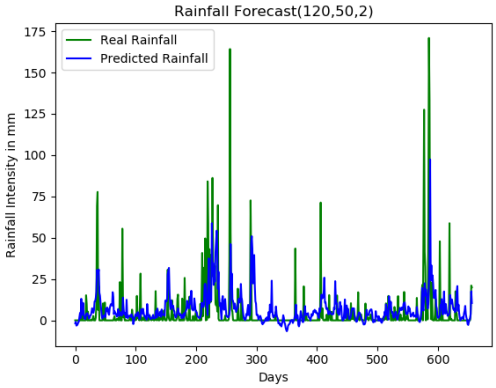
(f)


Rainfall Forecast(60,100,2)
Real Rainfall
Predicted Rainfall
Rainfall Intensity in mm
Days

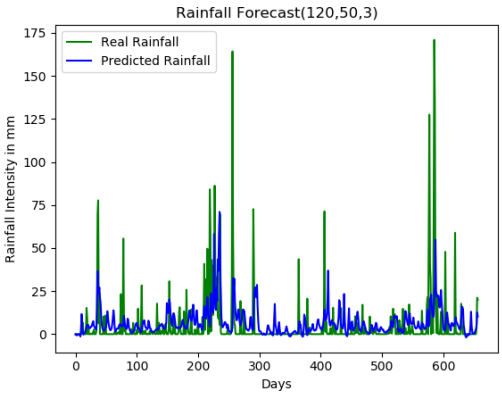
(g)


Rainfall Forecast(60,100,3)
Real Rainfall
Predicted Rainfall
Rainfall Intensity in mm
Days

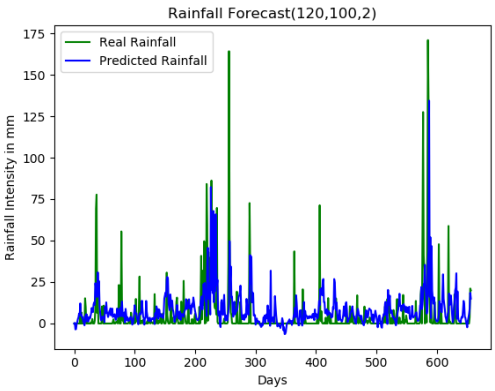
(h)


Rainfall Forecast(120,50,2)
Real Rainfall
Predicted Rainfall
Rainfall Intensity in mm
Days

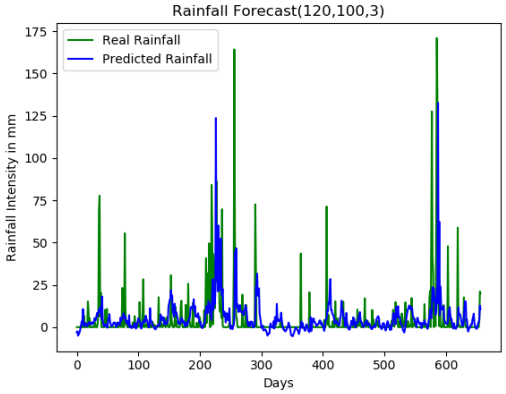
(i)


Rainfall Forecast(120,50,3)
Real Rainfall
Predicted Rainfall
Rainfall Intensity in mm
Days

(j)



(k)



(l)

Figure 47. RNN architectures: (a) [30, 50, 2] (b) [30, 50, 3] (c) [30, 100, 2] (d) [30, 100, 3] (e) [60, 50, 2] (f) [60, 50, 3] (g) [60, 100, 2] (h) [60, 100, 3] (i) [120, 50, 2] (j) [120, 50, 3] (k) [120, 100, 2] (l) [120, 100, 3]

CHAPTER 6

CONCLUSIONS

The results discussed in chapter 5 can be summarised as follows:

- The implementation of Univariate data on Persistence, Seasonal Persistence, Auto Regression and ARIMA suggests that ARIMA performs better with a RMSE value of 11.961 followed by Auto Regression with 12.024 and Seasonal Persistence (Walk Forward) with 13.63.
- Later, development of models using NAR and NARX suggests that Non-linear Auto Regressive Neural Network with external inputs such as temperature, relative humidity, etc. other than rainfall gives better result as compared to NAR with RMSE of 7.6925.
- Univariate Recurrent Neural Network trained on the rainfall data alone tends to perform better than the Multivariate RNN trained on multiple input parameters (atmospheric factors along with rainfall).
- Amongst the designed and tested architectures, the model with 12 neurons and 2 layers seem to give the most accurate and stable performance.
- Amongst the feature extraction techniques implemented to reduce the computational complexity PCA with 15 features extracted seemed to outperform kernel-PCA technique.
- The problems associated with high bias and high variance were taken care of using Dropout Regularization and k-fold Cross Validation technique. The reduction of RMSE value of the model where Drouput Regularization was implemented validates the same.
- On an overall comparison between the Univariate and Multivariate Time Series Analysus techniques, the ANN forecasting model with the architecture [2,12] gave the most promising results with the most minimal RMSE value of 4.788.
- As the region of target may vary, the non-linearity existing in the correlation between input and output parameters may increase or decrease. The best model to tolerate such high variation would be Artificial Neural Network though with an increased computational complexity. Machine learning algorithms like Random Forest seem to be another good option for forecasting model development in terms of computational complexity.

6.2 FUTURE SCOPE OF THE RESEARCH

- The forecasting model developed in this project can be extended to any other region of interest with the same data pre-processing techniques and few other suitable changes done in the algorithms used in this project in terms of hyper-parameters.
- The dataset under consideration in this project being continuous and non-linear can also be modelled using fuzzy logic algorithms like ANFIS (Adaptive Neuro-Fuzzy Inference System).
- Advanced feature extraction techniques like Genetic Algorithm can be included with fuzzy logic algorithms.
- The developed forecasted models can be adopted by meteorological departments to improve their disaster management.

References

[1] M.Kannan, S.Prabhakaran and P.Ramachandran. Rainfall Forecasting Using Data Mining Technique. International Journal of Engineering and Technology Vol.2 (6), 2010, 397-401.

[2] Afolayan Abimbola Helen, Ojokoh Bolanle A., Falaki Samuel O. Comparative Analysis of Rainfall Prediction Models Using Neural Network and Fuzzy Logic. International Journal of Soft Computing and Engineering (IJSCE) ,ISSN: 2231-2307, Volume-5 Issue-6, January 2016.

[3] Akashdeep Gupta, Anjali Gautam, Chirag Jain, Himanshu Prasad, Neeta Verma. Time Series Analysis of Forecasting Indian Rainfall. International Journal of Inventive Engineering and Sciences (IJIES) ISSN: 2319–9598, Volume-1, Issue-6, May 2013.

[4] V.K.Somvanshi, O.P.Pandey, P.K.Agrawal, N.V.Kalanker1, M.Ravi Prakash and Ramesh Chand. Modelling and prediction of rainfall using Artificial neural network and ARIMA techniques. J. Ind. Geophys. Union ,Vol.10, No.2, pp.141-151, April 2006 .

[5] Harshani R. K. Nagahamulla, Uditha R. Ratnayake and Asanga Ratnaweera. Monsoon Rainfall Forecasting in Sri Lanka using Artificial Neural Networks. 2011 6th International Conference on Industrial and Information Systems, ICIIS, Sri Lanka, 2011.

[6] Gunawansyah, Thee Houw Liong, Adiwijaya. Prediction and anomaly detection of rainfall using evolving neural network to support planting calender in soreang (Bandung). International Conference on Information and Communication Technology (ICoICT), 2004.

[7] Mislana, Haviluddinb , Sigit Hardwinartoc , Sumaryonod and Marlon Aipassae. Rainfall monthly prediction based on artificial neural network . International Conference on Computer Science and Computational intelligence (ICCSCI), 2015.

[8] N. Q. Hung, M. S. Babel, S. Weesakul, and N. K. Tripathi. An artificial neural network model for rainfall forecasting in Bangkok, Thailand. Hydrol. Earth Syst. Sci., 13, 1413–1425, 2009 .

[9] JamilehFarajzadeh ,AhmadFakheri Fard and SaeedLotfi. Modeling of monthly rainfall and runoff of Urmia lake basin using “feed-forward neural network” and “time series analysis” model. Water Resources and Industry 7-8 (2014) 38–48.

[10] JianshengWu ,JinLong and MingzheLi . Evolving RBF neural networks for rainfall prediction using hybrid particle swarm optimization and genetic algorithm. Neurocomputing Volume 148, 19 January 2015, Pages 136-142.

[11] https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/

[12] https://machinelearningmastery.com/visualize-deep-learning-neural-network-model-keras/

[13] https://machinelearningmastery.com/tune-arima-parameters-python/

[14] https://machinelearningmastery.com/grid-search-arima-hyperparameters-with-python/

[15] https://machinelearningmastery.com/autoregression-models-time-series-forecasting-python/

[16] https://machinelearningmastery.com/persistence-time-series-forecasting-with-python/

[17] https://machinelearningmastery.com/seasonal-persistence-forecasting-python/

[18] https://machinelearningmastery.com/time-series-data-visualization-with-python/

[19]https://s3.amazonaws.com/MLMastery/time_series_forecasting_with_python_mini_course.pdf?__s=qsfpiisrojpcbrqteydz

Curriculum Vitae

NAME: Tharun V.P

REGISTRATION NUMBER: 14BEC0664

COURSE: B.TECH

BRANCH: Electronics and Communication Engineering

UNIVERSITY: Vellore Institute of Technology, Vellore

EMAIL ID: tharunv.pvit@gmail.com

MOBILE NO.: 9994319022

RESIDENTIAL ADDRESS: “Tusharam House”, Olavanna, Calicut, Kerala-673025

CGPA: 8.3