**CHAPTER 1**

**1.1 INTRODUCTION**

Weather is the word used to describe the condition of the atmosphere at a specific moment and location, which is continuously changing from hour to hour and from day to day. Climate is the long term average condition of the atmosphere near the earth’s surface.

Temperature, precipitation, humidity, air pressure and wind speed are all factors to consider.

Precipitation Prediction will always help to make decision on agriculture, fisheries, forestry, tourism etc. Monsoon plays a significant role in agriculture production. For countries like India, where agricultural production has been one of the main factors affecting the economy of India, a decent amount of rainfall gives the entire country an economic outlook and boosts the economy. A decent amount of rain enhances crop productivity and also increases water resources. Where an excess amount of rainfall brings a flood, which destroys crops, causes structural damage, threatens human life. In India, floods occurred in 2019 due to excessive rainfall in July and August, which had affected 13 states, Karnataka and Maharashtra were the most severely affected states. The early prediction of rainfall is therefore essential. Rainfall prediction will help farmers to make decision on crop production and harvesting, as well as help prevent flooding, protect human lives and resources.

Precipitation forecast is very challenging due to its uncertain nature and dynamically changing climate. It is an application of science and technology to predict precipitation in advance. It’s always been a challenging task for meteorologists. Forecasting is done through the collection and analysis of weather and climate data.

Temperature, relative humidity, amount of sunshine and wind speed among others; are weather parameters which play a role to determine the amount of rainfall received on land. In this paper we will be identifying the factors that are significant for Precipitation. We have factors related to temperature, wind speed and humidity. We will identify the factors related to temperature, wind speed and humidity that are significant towards precipitation. Factors like temperature, wind speed and humidity will help us to understand the behavior of precipitation.

Nowadays, artificial intelligence techniques are booming in the market, are being used for data analysis and prediction purposes in different sectors. These techniques show better predictive accuracy. In this paper we will identify the predictive model for estimating Precipitation. An appropriate model among traditional time series and artificial neural network will be obtained based on accuracy measure for major impact factors, which will be used for forecasting.

**1.2 Literature Review**

Azad Abdulolhafedh [1] has used the Cochrane-Orcutt procedure to remove autocorrelation in vehicular crash dataset.

Dash, Y. et al. [2] has used three artificial intelligence approaches like K-Nearest Neighbor (KNN), Extreme Learning Machine (ELM), and Artificial Neural Network (ANN), for seasonal forecasting of the monsoon. These three techniques were used for predicting rainfall for the Kerala subdivision. The author found that the Extreme Learning Machine shows better performs as compared to KNN and ANN. ELM structure gives better predictive accuracy with minimal Mean Absolute Percentage Error for both summer monsoon (June-September) and post-monsoon(October-December).

Kashiwao, T., et al. [3] proposed a model to predict local rainfall in the region of Japan. Data were collected from the Japan Meteorological Agency (JMA). The proposed model automatically collected meteorological data of temperature, atmospheric pressure, vapor pressure, amount of precipitation, wind velocity, and humidity. Two methods, such as Multi-layer Perceptron (MLP) and Radial Basis Function Network (RBFN), were used for rainfall prediction. The result of this study showed that the MLP model was superior to that of the RBFN model for rainfall prediction.

Nurcahyo, S. et al. [4] conducted research on the weather forecast of rainfall over Kemayaran Jakarta. The system was built using a combined hybrid Genetic Algorithm (GA) and Partially Connected Feed forward Neural Network (PCFNN) to predict rainfall for 7 days ahead in Kemayaran Jakarta. Rainfall was predicted with 81.52 % accuracy. Dash, Y. et al. [5] in this study, artificial Intelligent (AI) Methods such as Extreme Learning Machine (ELM) and Single Layer Feed-Forward Network (SLFM) were used to predict Summer Monsoon in Kerala. Results of this study showed that ELM shows better results as compared to SLFM. The performance of these techniques was evaluated based on Mean Absolute Error and Root Mean Squared Error.

Dutta, P. S. et al. [6] proposed a model using data mining techniques for monthly rainfall prediction over Assam. Statistical technique - Multiple Linear Regression was used for prediction. The performance of the proposed model was measured in adjusted R-squared. 63% accuracy obtained using the given model.

Basha, C. Z. et al. [7] in this study, deep learning approach has represented for rainfall prediction. Deep learning techniques such as Multilayer Perceptron and Extreme Learning Machine NN were used for predicting rainfall. In this study, the CNN technique was used for taking input from past data. Performance of these techniques evaluated using MSE and RMSE.

Parmar, A. et al. [8] this paper reviewed different approaches and algorithm such as Artificial Neural Network (ANN) - Back-Propagation Neural Network, Multilayer Perceptron, Support Vector Machine (SVR), Layer Recurrent Network, and Extreme Learning Machine for rainfall prediction.

**1.3 OBJECTIVE OF THE STUDY**

* Studying the variable present in the dataset.
* Identifying the factors affecting Precipitation.
* Identifying the predictive model for estimating Precipitation.
* Developing appropriate traditional time series model for major impact factors.
* To obtain better model among Traditional time series and artificial neural network based on accuracy measures.
* Forecasting the impact factors for next 2 years.

**1.4 ABOUT DATA**

The dataset downloaded from the website <https://power.larc.nasa.gov/data-access-viewer/>. The dataset contains monthly average data from January 1981 to May 2022.

Variable Description:-

|  |  |
| --- | --- |
| **Name** | **Abbreviation** |
| Precipitation Corrected | PRECTOTCORR (mm/day) |
| Earth Skin Temperature | TS (◦ c) |
| Temperature at 2 Meters | T2M (◦ c) |
| Dew/Frost Point at 2 Meters | T2MDEW (◦ c) |
| Wet Bulb Temperature at 2 Meters | T2MWET (◦ c) |
| Specific Humidity at 2 Meters | QVM2M (g/kg) |
| Relative Humidity at 2 Meters | RH2M (%) |
| Wind Speed at 2 Meters | WS2M (m/s) |

**1.5 Software Used**

The data was in Excel sheet, saved in CSV file. During the analysis of time series we use R software. The different commands available in R language and packages are used to carry out the analysis.

**CHAPTER 2**

**METHODOLOGY**

**2.1 DISCRIPTIVE STATISTICS:-**

Descriptive statistics is the discipline of quantitatively describing the main features of data. Descriptive statistics are distinguished from inferential statistics (or inductive statistics), in that descriptive statistics aim to summarize a sample, rather than use the data to learn about population that the sample of data is thought to represent. The measures that describe the dataset are measures of central tendency and measures of variability or dispersion. Measures of central tendency include mean, median and mode, while measures of variability 4 include the standard deviation (or variance), the minimum and maximum variables, kurtosis and skewness.

**MEASURES OF CENTRAL TENDENCY**

**Mean:** The Arithmetic mean of a set of observations , ,…. defined by,

=

**Median:** Median is determined by sorting the data set from lowest to highest values taking the data point in the middle of the sequence. There is equal number of points above and below the sequence. The median can be determined for ordinal data and as well as interval and ratio data. Unlike, the mean, the median is not influenced by the outliers of extremes of dataset.

**Mode:** The mode is the most frequently occurring value in the dataset. The mode can be very useful for dealing with categorical data. The mode also can be used with ordinal, interval and ratio data. However, in the interval and ratio scales, the data may be spread thinly with no data points having the same value. In such cases, the mode may not exist or may not be very meaningful.

**MEASURES OF VARIABILITY**

**Variance:** The most commonly used measure of dispersion in statistical analysis is called the variance. It is a measure that takes into account all the values in a set of observations.

=

The wider the dispersion of the values around their mean, the greater the variance. The positive square root of the variance is called the standard deviation and is denoted by s.

**Standard Error:** A Standard error is the standard deviation of the sample distribution of a statistic. Standard error is a statistical term that measures the accuracy with which a sample represents a population. In statistics, a sample mean deviates from the actual mean if a population, this deviation is the standard error. The term standard error is used to refer to the standard deviation of various sample statistics such as mean or median.

For example “the standard error of the mean” refers to the standard deviation of the distribution of sample means taken from a population. The smaller the standard error, the more representative the sample will be the overall population.

The standard error is inversely proportional to the sample size; the larger the sample size, the smaller the standard error because the statistic will approach the actual value.

The standard error is considered part of descriptive statistics. It represents the standard deviation of the mean with the dataset. This serves as a measure of variation for random variables, providing a measurement for the spread. The smaller the spread, the more accurate the dataset is said to be.

**MEASURE OF SHAPE KURTOSIS**

In probability theory and statistics, kurtosis is any measure of the peakedness” of the probability distribution of a real- valued random variable. In a similar way to the concept of skewness, kurtosis is a descriptor of the shape of a probability distribution and just as for skewness, there are different ways of quantifying it for a theoretical distribution and corresponding ways of estimating it from a sample from a population. One common measure of kurtosis, originating with Karl Pearson, is based on a scaled version of the fourth moment of the data or population.

The fourth standardized moment is defined as,

= =

Where µ4 is the fourth moment about the mean and σ is the standard deviation. Distributions with zero excess kurtosis are called mesokurtic. A distribution with positive excess kurtosis is called leptokurtic. A distribution with negative excess kurtosis is called platykurtic.

**SKEWNESS:** In probability theory and statistics, skewness is a measure of the extent to which a probability distribution of a real -valued random variable “leans” to one side of the mean. The skewness value can be positive or negative or even undefined. For unimodal distribution, negative skew indicates that the tail on the left side of the probability density function is longer or fatter than the right side- it does not distinguish these shapes. Conversely, positive skew indicates that the tail on the right side is longer or fatter than the left side. In cases where one tail is long but the other tail is fat, skewness does not obey a simple rule. The skewness of a random variable X is the third standardized moment, denoted and defined as,

= E[] = = =

Where µ3 is the third moment about the mean µ,σ is the standard deviation , and E is the expectation operator. The last equality expresses skewness in terms of the ratio of the third cumulate k3 and the 1.5th power of the second cumulate k2. This analogous to the definition of kurtosis as the fourth cumulate normalized by the square of the second cumulate.

**Measures of Accuracy:**

**Mean Absolute Percentage Error (MAPE):**

The mean absolute percentage error (MAPE), also known as mean absolute percentage deviation (MAPD), is a measure of prediction accuracy of a forecasting method in [statistics,](https://en.wikipedia.org/wiki/Statistics) for example in [trend estimation](https://en.wikipedia.org/wiki/Trend_estimation). It usually expresses accuracy as a percentage, and is defined by the formula:

where *At* is the actual value and *Ft* is the forecast value.

The difference between *At* and *Ft* is divided by the Actual value *At* again. The absolute value in this calculation is summed for every forecasted point in time and divided by the number of fitted points *n*. Multiplying by 100% makes it a percentage error.

**Root-Mean-Square Error (RMSE)**

The Root Mean Square Error (RMSE) (also called the root mean square deviation, RMSD) is a frequently used measure of the difference between values predicted by a model and the values actually observed from the environment that is being modelled. These individual differences are also called residuals, and the RMSE serves to aggregate them into a single measure of predictive power.

The RMSE of a model prediction with respect to the estimated variable is defined as the square root of the mean squared error

RMSE =

where is observed values and is modelled values at time/place i.

**Mean Absolute Error (MAE)**

The mean absolute error, or MAE, is calculated as the average of the forecast error values, where all of the forecast values are forced to be positive. Forcing values to be positive is called making them absolute

MAE =

Where {} is the actual observations time series

{} is the estimated or forecasted time series

N is the number of non-missing data points.

**MULTICOLLINEARITY**

The use and interpretation of a multiple regression model often depends explicitly or implicitly on the estimates of the individual regression coefficient. Some examples of inferences that are frequently made include

1) Identifying the relative effects of the repressor variables.

2) Prediction and/or estimation.

3) Selection of an appropriate set of variables for the model.

If there is no linear relationship between the repressors, they are said to be orthogonal. When the repressors are orthogonal, inferences such as those illustrated above can be made relatively easily. Unfortunately, in most applications of regression, the repressors are not orthogonal. However, in some situations the repressors are nearly perfectly linearly related and in such cases the inferences based on the regression model can be misleading or erroneous. When there are near-linear dependencies among the repressors, the problem of Multicollinearity is said to exist.

Multicollinearity is a statistical phenomenon in which two or more predictor variables in a MLR model are highly correlated. In this situation coefficient estimates of multiple regressions may change erratically in response to small changes in the model or data. Multicollinearity does not reduce the predictive power of the model; it affects calculation regarding the individual significance of the predictors.

**DEFINITION:**

Multicollinearity is said to exist in the regression model 𝑌 = 𝑋𝛽 + 𝜖, if Rank(X) < k where k denotes the number of repressors in the model including the intercept term.

If Rank(X) < k then (X`X) is singular. In this case columns of X can be written as exact linear combination of remaining repressors. This situation is known as “Perfect Multicollinearity”. If some of the repressors can be written as nearly linear combination of remaining repressors. In this situation (X`X) becomes near singular. This situation is called as “Near Multicollinearity”.

**Effects of Multicollinearity:**

Multicollinearity can create inaccurate estimates of the regression coefficients; inflate the standard errors of the regression coefficients, deflate the partial t tests for the regression coefficients, give false, non-significant, p values, and degrade the predictability of the model.

Multicollinearity has several effects these are described as follows:

1) High variance of coefficients may reduce the precision of estimation.

2) Multicollinearity can result in coefficients appearing to have the wrong sign. Estimates of coefficients may be sensitive to particular sets of sample data.

3) Some variables may be dropped from the model although, they are important in the population.

4) The coefficients are sensitive of to the presence of small number inaccurate data values.

Because Multicollinearity is a serious problem when we need to make inferences or looking for predictive models. So it is very important for us to find a better method to deal with Multicollinearity.

**Measuring Multicollinearity:**

1) Check the sign of the estimate of regression coefficients. If it is different from the sign based on past knowledge, it indicates the presence of Multicollinearity.

2) Check the t statistic of each repressors. If it is small for the repressors which are consider to be significant based on the past knowledge, it indicates the presence of Multicollinearity.

3) If we delete any row or column of the data matrix, X makes a large shift in the estimators; it indicates the presence of Multicollinearity.

4) If most of the repressors are insignificant with very large R2 indicates the presence of Multicollinearity.

5) Variance Inflation Factors: We know that (𝛽̂𝑜𝑙𝑠)= VIF=( ) Where is the coefficient of determination obtained when Xi regress on remaining repressors. In empirical study, it reveals that VIF more than 5 indicates the presence of Multicollinearity.

6) Correlation: Compute pair wise correlation among the repressors, if any one of particular correlation is very high, it indicates that the particular pair causes Multicollinearity.

7) Condition Indices: For any positive semi definite matrix A, the condition number is defined by,

𝜆𝑗 =

Where S1, S2,…..,Sn are the singular values of A. empirical study reveals that VIF more than 30 indicates the presence of Multicollinearity.

Estimation: It is argued in the literature that the presence of near Multicollinearity is more complicated than the perfect Multicollinearity in the model. Therefore it is argued that the biased estimator with smaller mean squared error is preferable to an unbiased estimator with very large variance in the presence of near Multicollinearity. In case of near Multicollinearity OLS estimator exist and unbiased, but has very large variance. This makes t ratio small and estimator imprecise.

**HETROSCEDASTICITY**

**INTRODUCTION:**

Heteroscedasticity occurs when the error variance has non-constant variance. In this case, we can think of the disturbance for each observation as being drawn from a different distribution with a different variance. Stated equivalently, the variance of the observed value of the dependent variable around the regression line is non-constant. We can think of each observed value of the dependent variable as being drawn from a different conditional probability distribution with a different conditional variance. A general linear regression model with the assumption of heteroscedasticity can be expressed as follows,

= ++…….++

Var( = E( = for t = 1,2,…….,n

note that we now have a t subscript attached to sigma squared. This indicates that the disturbance for each of the n-units is drawn from a probability distribution that has a different variance.

**CONSEQUENCES OF HETEROSCEDASTICITY:**

If the error term has non-constant variance, but all other assumptions of the classical linear regression model are satisfied, then the consequences of using the OLS estimator to obtain estimates of the population parameters are:

1. The OLS estimator is still unbiased.
2. The OLS estimator is inefficient; that is, it is not BLUE.
3. The estimated variances and covariance of the OLS estimates are biased and inconsistent.
4. Hypothesis tests are not valid.

**DETECTION OF HETEROSCEDASTICITY:**

There are several ways to use the sample data to detect the existence of heteroscedasticity.

1. Plot the Residuals: The residual for the observation, , is an unbiased estimate of the unknown and unobservable error for that observation, . Thus the squared residuals, , can be used as an estimate of the unknown and unobservable error variance, =E( ) . You can calculate the squared residuals and then plot them against an explanatory variable that you believe might be related to the error variance. If you believe that the error variance may be related to more than one of the explanatory variables, you can plot the squared residuals against each one of these variables. Alternatively, you could plot the squared residuals against the fitted value of the dependent variable obtained from the OLS estimates. Most statistical programs have a command to do these residual plots for you. It must be emphasized that this is not a formal test for heteroscedasticity. It would only suggest whether heteroscedasticity may exist. You should not substitute the residual plot for a formal test.

2. Breusch-Pagan Test and Harvey-Godfrey Test: There are a set of heteroscedasticity tests that require an assumption about the structure of the heteroscedasticity, if it exists. That is, to use these tests you must choose a specific functional form for the relationship between the error variance and the variables that you believe determine the error variance. The major difference between these tests is the functional form that each test assumes. Two of these tests are the Breusch-Pagan test and the Harvey-Godfrey Test. The Breusch-Pagan test assumes the error variance is a linear function of one or more variables. The Harvey-Godfrey Test assumes the error variance is an exponential function of one or more variables. The variables are usually assumed to be one or more of the explanatory variables in the regression equation.

**AUTOCORRELATION:-**

**INTRODUCTION**

One of the basic ideal conditions of the model is that successive values of the random variable 𝜀 are temporarily independent, that is, the value of 𝜀 assumes any one of the period is independent from the values which it is assumed in any previous period.

This assumption implies that Cov(𝜀𝑖𝜀𝑗 )=0 for i≠j i.e., the disturbances in the equation and the equation are uncorrelated. This assumption is no longer true when we deal with time series data, where successive observations are highly correlated. Such correlation is known as “Autocorrelation or serial correlation”.

Autocorrelation is the correlation between members of the same series for different periods. In this case E(𝜀𝑖𝜀𝑗 ) ≠0 and hence D(𝜀) = Ω ≠ 𝐼.

Hence the diagonal elements of Ω are zero but the off diagonal elements are non-zero. Therefore autocorrelation is a particular case of heteroscedasticity.

**SOURCES OF AUTOCORRELATION**

Auto correlated values of the disturbance term 𝜀 may be observed for the following reasons.

1. Omitting (or excluding) relevant repressors from the regression model.

2. Misclassification of the mathematical form of the model.

3. Interpolations in the statistical observations.

4. Mis-specification of the true random term 𝜀.

5. Seasonal adjustment procedure.

**CONSEQUENCES OF AUTOCORRELATION IN OLSE**

When the disturbance term exhibits serial autocorrelation the values as well as the standard errors of the parameter are affected.

In this case OLS estimate of 𝛽 exists and can be shown to be unbiased, consistent and asymptotically normal. But V(𝛽̂ols)≠ 𝜀 i.e. with auto correlated values of the disturbance term the OLS variance of the parameter estimates are likely to be larger than those of other econometric methods.

Therefore

1. OLS estimator in case of auto correlated error is inefficient.

2. The inferences based on the variance covariance matrix of 𝛽̂ is misleading, that is usual tests developed based on OLSE are not valid.

**TESTS FOR AUTOCORRELATION**

We know that some rough idea of the existence of pattern of autocorrelation may be gained by plotting the regression residuals either against their own lagged values or against time. However, there are more accurate tests for the incidence of autocorrelation namely ‘Durbin-Watson test’ and ‘Durbin h statistics’ respectively.

**Durbin Watson Test**

Based on graphical observations Durbin and Watson developed a test for detecting autocorrelation, which is applicable to small samples. However, the test is appropriated only for the first order autoregressive scheme ( = 𝜌+).

The test is based on the following assumptions on the regression model

1. The model should include the intercept term. If such term, is not present, then rerun the regression such that the model intercept term.
2. The model should not include the lagged endogenous variable.
3. X is not stochastic, i.e., are not random variables.
4. It follows first order autoregressive scheme i.e., 𝜀𝑡~(1).

The test maybe outlined as follows

The null hypothesis is, H0: 𝜌 = 0

Or H0: The 𝜀’s are not correlated with a first order scheme.

This hypothesis is tested against the alternative hypothesis

H1: 𝜌 ≠ 0 Or H1: the 𝜀’s are correlated with a first order scheme.

Under the above assumptions Durbin-Watson suggested a test statistic for testing the autocorrelation as 𝑑 =

Where denotes the OLS residuals given by = -

For large n, 𝑑 ≈ 2 − 2𝜌̂, where 𝜌̂ =

From this expression we observe that the values of‘d’ lies between 0 and 4.

Unlike t ,, F statistic there is no unique critical region that can be used for detecting autocorrelation. However Durbin-Watson succeeded in determining two bounds for d namely the lower bound and the upper bound. They have shown that these bounds are not depending on X and Y. Hence it depends on number of observations and the number of repressors ‘k’ including intercept term. Depending upon the computed value of ‘d’ and comparing it with and , we will decide upon the presence and absence of autocorrelation. We have the four different cases.

1. If d\* < we reject the null hypothesis of no autocorrelation and accept that there is a positive autocorrelation of first order.
2. If d\* > (4-) we reject the null hypothesis of no autocorrelation and accept that there is a negative autocorrelation of first order.
3. If < d\* < (4-) we accept the null hypothesis of no autocorrelation.
4. If < d\*<or (4- ) < d\*< (4-) the test is inconclusive.

**Method to remove Autocorrelation**

**The Cochrane-Orcutt Procedure:-**

The Cochrane-Orcutt procedure uses the ordinary least square residuals to obtain the value of rho which minimizes the sum of squared residuals. Rho is then used to transform the observations of the variables. The process continues until convergence is reached. Considering the general ordinary least squared regression model;

= α + β + --- (1)

Where,

: the dependent variable at time (lag) t,

α: the intercept,

β: the vector of regression coefficient,

: the vector of explanatory variables at time (lag) t,

: the error term of the model at time (lag) t.

When applying the DW test, if the DW statistic revealed that the temporal autocorrelation exists among the model error terms, then the residuals must be modeled for the first order autoregressive term AR(1) such that:

= ρ + ---- (2)

Where,

ρ: the temporal autocorrelation coefficient (rho) between pairs of observations, 0< ρ<1,

: the error term of the residuals at time (lag) t.

The Cochrane-Orcutt procedure is obtained by taking a quasi-differencing or generalizes differencing, such that the sum of squared residuals is minimized.

– ρ = α (1- ρ) + β () + --- (3)

The Cochrane-Orcutt iterative procedure starts by obtaining parameters estimates by the ordinary least square regression (OLS). Applying equation (2), the OLS residuals are then used to obtain an estimate of rho from the OLS regression. This estimate rho is then used to produce transformed observations, and parameter estimates are obtained again by applying OLS to the transformed model. A new estimate of rho is computed and another round of parameter estimates is obtained. The iteration stops when successive parameter estimates differ by less than 0.001.

**Remedial Measures for Multicollinearity:**

1. Ridge Regression.

2. Lasso Regression.

3. Elastic net Regression.

**2.2 MULTIPLE LINEAR REGRESSION**

In general, the purpose of Multiple Linear Regressions (MLR) model is to learn the linear combination of two or more predictor variables is used to explain the variation in a response. In essence, the additional predictors are used to explain the variation in the response not explained by a simple linear regression fit.

The general form of a multiple regression is,

Y=+ ++⋯++

Where Y is the dependent variable and, are the k predictor variables. Each of the predictor variables must be numerical. The coefficients,…, measure the effect of each predictor after taking account of the effect of all other predictors in the model. Thus, the coefficients measure the marginal effects of the predictor variables.

This can be written in matrix notation as 𝑌 = 𝑋𝛽 + 𝜖

Where Y is known as observational vector on the response variable and X is matrix of observations on the repressors, β is the vector of regression coefficient and 𝜖 is an unobserved error vector.

ASSUMPTIONS:

The classical linear regression model consists of a set of assumptions about how a data set will be produced by an underlying “data generating process”. The assumptions describe the forms of the model and relationships among its parts and imply appropriate estimation and inference procedures.

1) X is non-stochastic. This means that regressions, are fixed constants and are not random variables.

2) X are linearly independent that is Rank(X) = k. this means that the regressions, are independent that is they are not correlated.

3) E() =0, this means that average error term takes the value zero.

4) V() = for all t=1….k and cov()=0 for all i≠j this means that error term are uncorrelated and has constant variance for all units.

5), ,…. are independently distributed normal variables with mean zero and common variance 𝜎 2 and ’s and ’s are uncorrelated.

6) Lt = 𝑄 is finite and non-singular this means that regression model does not include too small or too big regression.

These assumptions are called basic ideal conditions.

In multiple regression model 𝑌 = 𝑋𝛽 + 𝜖 we assume that all basic ideal conditions are satisfied. Under this situation OLSE of the model parameters are BLUE, efficient, sufficient and also asymptotically efficient.

**2.3 Ridge Regression:**

Ridge Regression is a technique for analyzing multiple regression data that suffer from multicollinearity. When multicollinearity occurs, least squares estimates are unbiased, but their variances are large so they may be far from the true value. By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors. It is hoped that the net effect will be to give estimates that are more reliable. Another biased regression technique, principal components regression, is also available in NCSS. Ridge regression is the more popular of the two methods. Ridge Regression is technique which penalizes the size of the regression coefficients in order to deal with multicollinearity variables. It is based on the Thikhonov regulation named after the mathematician Andrev Thikhonov.

Definition:

Ridge estimator of the regression parameter β in 𝑌 = 𝑋𝛽 + 𝜖 is defined as, 𝛽̃𝑟𝑖𝑑𝑔𝑒 = 𝑋′𝑌 Where µ is called ridge parameter.

Ridge regression is like least squares but shrinks the estimated coefficients towards zero. Given a response vector y ∈ Rn and a predictor matrix X ∈ Rn×p, the ridge regression coefficients are defined as

𝑟𝑖𝑑𝑔𝑒 = 𝑎𝑟𝑔𝑚𝑖𝑛 + λ

Here λ ≥ 0 is a tuning parameter, which controls the strength of the penalty term.

Note that:

♣ When λ = 0, we get the linear regression estimate

♣ When λ = ∞, we get 𝛽̂ridge = 0

♣ For λ in between, we are balancing two ideas: fitting a linear model of y on X, and shrinking the coefficients.

Tuning parameter λ can be found using cross validation method.

**Cross-validation** arises frequently in the context of assessing how a particular model fit predicts future observations and how to optimally select model parameters.

Methods for cross-validation usually involve withholding a random subset of the data during model fitting (training set) and quantifying how accurate the withheld data are predicted (testing set) and repeating this process to get a measure of prediction accuracy. When this partitioning procedure happens once, it's called the **holdout method**.

= + λ

Ridge regression uses a type of shrinkage estimator called a ridge estimator. Shrinkage estimators theoretically produce new estimators that are shrunk closer to the “true” population parameters. The ridge estimator is especially good at improving the least-squares estimate when multicollinearity is present.

**2.4 LASSO REGRESSION**

In statistics one of the main goals is to build a model that better represent a dataset, this process include the task of features selection. The only aim of the researcher is to build a model that describes a response variable; in order to do so one of the first question that the researcher should be able to answer is which features/variables should I take into consideration? or Which are the most important attributes to describe the response variable? The goal of is to describe the Lasso method. The LASSO method will be analyzed for both the linear models and the Generalized Linear Models. Furthermore in order to test the efficiency of LASSO, the method will be applied to real data, and the results will be analyzed and described.

**Feature Selection**

The feature selection is the process that choose a reduced number of explanatory variable to describe a response variable. The main reasons why feature selection is used are:

• make the model easier to interpret, removing variables that are redundant and do not add any information.

• reduce the size of the problem to enable algorithms to work faster, making it possible to handle with high-dimensional data.

• reduce overfitting.

The variable selection is even more important for the high dimensional datasets; here the number of features is very high, sometimes higher than the number of observation. In these situation it is hard to easily say which of the variables are relevant and which ones are irrelevant, and on the other hand, it is difficult, due to dimensionality issues, to build and interpret a model that takes into consideration all the variables. For these reasons the feature selection is an important task. In the literature there are several types of methods to complete the feature selection task. First the Filter Methods select the features by ranking them on how useful they are for the model, to compute the usefulness score statistical test and correlation results are used (e.g. Chi-square, ANOVA, Pearson’s correlation).

Secondly Wrapper Methods generates different subsets of features, each subset is then used to build a model and train the learning algorithm. The best subset is selected by testing the algorithm. To select the features for the subsets different criteria are used (e.g. Forward and Backward selection). Finally the Embedded Methods are a combination between the two previous methods. The Embedded Methods included the LASSO methods that is going to be studied in details in this.

**What is LASSO?**

**LASSO - Least Absolute Shrinkage and Selection Operator** - was first formulated by Robert Tibshirani in 1996. It is a powerful method that perform two main tasks: regularization and feature selection. The LASSO method puts a constraint on the sum of the absolute values of the model parameters, the sum has to be less than a fixed value (upper bound). In order to do so the method apply a shrinking (regularization) process where it penalizes the coefficients of the regression variables shrinking some of them to zero. During features selection process the variables that still have a non-zero coefficient after the shrinking process are selected to be part of the model. The goal of this process is to minimize the prediction error.

In practice the tuning parameter λ, that controls the strength of the penalty, assume a great importance. Indeed when λ is sufficiently large then coefficients are forced to be exactly equal to zero, this way dimensionality can be reduced. The larger is the parameter λ the more number of coefficients are shrinked to zero. On the other hand if λ = 0 we have an OLS (Ordinary Least Square) regression.

There are many advantages in using LASSO method, first of all it can provide a very good prediction accuracy, because shrinking and removing the coefficients can reduce variance without a substantial increase of the bias, this is especially useful when you have a small number of observation and a large number of features. In terms of the tuning parameter λ we know that bias increases and variance decreases when λ increases, indeed a trade-off between bias and variance has to be found. Moreover the LASSO helps to increase the model interpretability by eliminating irrelevant variables that are not associated with the response variable, this way also overfitting is reduced. This is the point where we are more interested in because in this paper the focus is on the feature selection task.

LASSO in Linear Models In this chapter we introduce the linear models together with a brief explanation of their main features. Moreover we explain how to use the LASSO method in this context.

**Linear Model**

Statistical Model

A statistical model is a mathematical representation of a real-world problem. The model should summarize and explain the data as close as possible to the reality but it also should be simple and easy to understand and apply. Usually the researcher has some data collected from the real world and his purpose is to build a model on them without losing too much information. A model can be composed by two different type of variables.

**Response variable** (dependent variable) is the focus of the experiment, it is the output of the model that the researcher wants to investigate on. The response variable can be a single one, Univariate Models, or can be multiple, Multivariate Models. In this paper we only consider Univariate Models.

**Explanatory variables** (independent variables) are measured or set by the researcher, these are the input of the model. These variables are called explanatory because they explain how the response variable is affected by their changes.

During the modelling process one of the phases is the features selection that can be done using the LASSO method. In this task the role of the explanatory variables is central, in fact it is really important, in order to build a good model, to choose the right variables that influence the response variable. In particular in the high-dimensional datasets, where the number of variables is bigger than the number of observations, the selection of features gains greater importance.

The response variable can have different type of probability distributions. When the response variable is normally distributed we consider the Linear Model while when the response variable follows different distributions we talk about Generalized Linear Model.

**Linear Model**

Linear Model, often called Linear Regression Model, is the model that describes the relationship between response Yi and explanatory variables Xij. The case of one explanatory variable is called Simple Linear Regression while the case with two or more explanatory variables is 23 called Multiple Linear Regression. The first assumption of the Linear Regression Model is that the response variable is normally distributed.

~ N ( , ; i = 1,2,……,n

Another assumption is the linearity of the model that is a linear relationship between the response variable and the explanatory variables. The Linear model can be expressed as follow

=+ ++⋯++ ;i= 1,2,……,n

where the parameters ,,..., are the regression coefficients and k is the number of explanatory variables. Moreover represent the random error, we assume they have 0 mean, constant variance and they are independent.

i.e ~ N (0,

The vector-notation is also used

Y = Xβ + ε

with response vector , design matrix , coefficient vector βk×1 and the error vector .

The goal of linear regression is to fit a straight line to a number of points minimizing the sum of squared residuals. Regression models are used for many purposes, for instance analysis of variance (ANOVA), parameters estimation, prediction and variable selection. Later, in the practical example, we will show how to implement some of them and we will focus on variable selection using LASSO method.

**The LASSO estimator**

LASSO is a regularization and variable selection method for statistical models. We first introduce this method for linear regression case. The LASSO minimizes the sum of squared errors, with a upper bound on the sum of the absolute values of the model parameters. There are different mathematical form to introduce this topic, we will refer to the formulation used by Buhlmann and van de Geer [1]. The lasso estimate is defined by the solution to the l1 optimization problem

Minimize ( ) subject to < t

where t is the upper bound for the sum of the coefficients. This optimization problem is equivalent to the parameter estimation that follows

) = argmin ( + λ )

Where = , = and λ ≥ 0 is the parameter that controls the strength of the penalty, the larger the value of λ, the greater the amount of shrinkage. The relation between λ and the upper bound t is a reverse relationship. Indeed as t becomes infinity, the problem becomes an ordinary least squares and λ becomes 0. Vice versa as t becomes 0, all coefficients shrink to 0 and λ goes to infinity. In this research paper we are going to use LASSO for its variable selection property. When we minimize the optimization problem some coefficients are shrank to zero, i.e. (λ) = 0, for some values of j (depending on the value of the parameter λ). In this way the features with coefficient equal to zero are excluded from the model. For this reason LASSO is a powerful method for feature selection while other methods (e.g. Ridge Regression) are not.

=

**Uses of LASSO Regression:**

Lasso is a regularization technique. We use lasso to:

• Reduce the number of predictors in a regression model.

• Identify important predictors.

• Select among redundant predictors.

• Produce shrinkage estimates with potentially lower predictive errors than ordinary least squares.

Unlike ridge regression, as the penalty term increases, lasso sets more coefficients to zero. This means that the lasso estimator is a smaller model, with fewer predictors. As such, lasso is an alternative to stepwise regression and other model selection and dimensionality reduction techniques.

**Limitations of Lasso regression:**

Although the lasso has shown success in many situations, it has some limitations.

Consider the following three scenarios.

1. In the p>n case, the lasso selects at most n variables before it saturates, because of the nature of the convex optimization problem. This seems to be a limiting feature for a variable selection method. Moreover, the lasso is not well defined unless the bound on the L1-norm of the coefficients is smaller than a certain value.
2. If there is a group of variables among which the pairwise correlations are very high, then the lasso tends to select only one variable from the group and does not care which one is selected.
3. For usual n>p situations, if there are high correlations between predictors, it has been empirically observed that the prediction performance of the lasso is dominated by ridge regression (Tibshirani, 1996).

**2.5 Elastic net regression:**

We propose the elastic net, a new regularization and variable selection method. Real world data and a simulation study show that the elastic net often outperforms the lasso, while enjoying a similar sparsity of representation. In addition, the elastic net encourages a grouping effect, where strongly correlated predictors tend to be in or out of the model together. The elastic net is particularly useful when the number of predictors (p) is much bigger than the number of observations (n). By contrast, the lasso is not a very satisfactory variable selection method in the p ≥ n case.

Our goal is to find a new method that works as well as the lasso whenever the lasso does the best, and can fix the problems that were highlighted above, i.e. it should mimic the ideal variable selection method in scenarios (a) and (b), especially with micro array data, and it should deliver better prediction performance than the lasso in scenario (c).

**Definition of Elastic Net:-**

The elastic net technique solves this regularization problem. For an α strictly between 0 and 1. Elastic net is the same as lasso when α = 1. As α shrinks toward 0, elastic net approaches ridge regression. In statistics and, in particular, in the fitting of linear logistic regression models , the elastic net is a regularized regression method that linearly combines the L1 and L2 of the lasso and ridge methods.

The elastic net method overcomes the limitations of the LASSO (least absolute shrinkage and selection operator) method which uses a penalty function based on

=

Use of this penalty function has several limitations. For example, in the "large p, small n" case (high-dimensional data with few examples), the LASSO selects at most n variables before it saturates. Also if there is a group of highly correlated variables, then the LASSO tends to select one variable from a group and ignore the others. To overcome these limitations, the elastic net adds a quadratic part to the penalty (), which when used alone is ridge regression (known also as Tikhonov regularization). The estimates from the elastic net method are defined by

= + λ [ (1- α ) + α

The L1 part of the penalty generates a sparse model.

• The quadratic part of the penalty

– Removes the limitation on the number of selected variables.

– Encourages grouping effect.

– Stabilizes the L1 regularization path.

The Elastic Net addresses the aforementioned “over-regularization” by balancing between LASSO and ridge penalties. In particular, a hyper parameter, namely Alpha, would be used to regularize the model such that the model would become a LASSO in case of Alpha = 1 and a ridge in case of Alpha = 0. In practice, Alpha can be tuned easily by the cross-validation.

**2.6 TIME SERIES ANALYSIS**

Traditional methods of time series analysis are mainly concerned with decomposing the variation in a series into components representing trend, seasonal variation and other cyclic changes. Any variation changes are attributed to irregular fluctuations. This approach is not always the best but is particularly valuable when the variation is dominated by trend and seasonality. However, it is worth noting that a decomposition into ten and seasonal variation is generally not unique unless certain assumptions are made. Thus some sort of modeling, either explicit or implicit maybe involved in carrying out these descriptive techniques and this demonstrates the blurred border line that always exists between descriptive and inferential techniques in statistics.

**Objectives of time series analysis**

Our purposes are to study techniques for drawing inferences from such series. Before we can do this, however, it is necessary to set up hypothetical probability model to represent the data. After an appropriate family of models has been chosen, it is then possible to estimate parameters, check for goodness of fit to the data, and possibly to use the fitted model to enhance the our understanding of the mechanism generating the series. Once a satisfactory model has been developed, it may be used in a variety of ways depending on the particular field of application.

The model may be used simply to provide a compact description of data. For the interpretation of economic statistics such as unemployment figures, it is important to recognize the presence of seasonal components and to remove them so as not to confuse them with long term trends. This process is known as seasonal adjustment. Other applications of time series models include separation of noise from signals, prediction of future values of series, testing hypothesis such as global warming using recorded temperature data, predicting future sales using previous sales data and controlling future values of a series by adjusting parameters.

Time series models are also useful in simulation studies, for example the performance of a reservoir depends heavily on the random daily inputs of water to the system. If these are modeled as a time series, then we can use the fitted model to simulate the large number of independent sequences of daily inputs. Knowing the size and mode of operation of the reservoir, we can determine the fraction of the simulated input sequences that cause reservoir to run out of water in a given time period. This fraction will then be an estimate of the probability of emptiness of the reservoir at some time in the given period.

**Components of Time Series**

A time series is a collection of observations of well-defined data items obtained through repeated measurements over time (often at equally spaced time intervals). An observed time series can be decomposed into four components: the trend (long term direction), the seasonal (systematic, calendar related movements), the cyclical (regular or periodic fluctuations) and the irregular (unsystematic, short term fluctuations). Therefore, a time series Xt can be represented in terms of these four components as:

Xt = Mt + St + Ct + I

Mt is the trend component

St is the seasonal component

Ct is the cyclical component

I is the irregular component

The different sources of variation are describes as follows:

**Seasonal *variation***

Many time series, such as sales figures and temperature readings exhibit variation that is annual in period. For example, unemployment is typically ‘high’ in winter but ‘low’ in summer. This yearly variation is easy to understand and can readily be estimated if seasonality is of direct interest. Alternatively, seasonal variation can be removed from the data, to give depersonalized data, if seasonality is not of direct interest.

***Other cyclic variation***

Apart from seasonal effect, some time series exhibit variation at a fixed period due to some other physical course. An example is daily variation in temperature. In addition some time series exhibit oscillations which are predictable to some extent. For example, economic data are sometimes thought to be affected by business cycles with a period varying from about three or four years to more than ten years, depending on the variable measured. However, the existence of such business cycle is the subject of some controversy, and there is increasing evidence that any such cycles are not symmetric. An economy usually behaves differently when going into recession, rather than emerging from recession.

**Trend**

This may be loosely defined as long-term change in the mean level. A difficulty with this definition is deciding what is meant by long term. For example, climatic variables sometime exhibit cyclic variation over a very long time period such 50 years. If one just had 20 years of data, this long term oscillation may look like a trend, but if several hundred years of data were available, then the long term cyclic variation wound be visible. Nevertheless in the short term it may still be more meaningful to think of such a long term oscillation as a trend. Thus in speaking of a trend, we must take into account the number of observations available and make a subjective assessment of what is meant by the phrase ‘long term’. As for seasonality, methods are available either for estimating the trend, or for removing it so that the analyst can look more closely at other sources of variation.

**Other Irregular Fluctuations**

After trend and cyclic variations have been removed from a set of data, we are left with a series of residuals that may or may not be random. In due course, we will examine various techniques for analyzing series of this type, either to see whether any cyclic variation is still left in the residuals, or whether apparently irregular variation may be explained in terms of probability models, such as moving average (MA) or autoregressive (AR) models.

**Stationary of a Time Series Process**

A time series is said to be stationary if its underlying generating process is based on a constant mean and constant variance with its autocorrelation function (ACF) essentially constant through time. Thus if we consider different subsets of a realization the different subsets will typically have means, variances and ACF that do not significant.

**Strictly stationary**

A time series is said to be strictly stationary if the joint distribution associated with m observations.

X( t1) , X( t2) , … , X( tm) made at any set of time t1, t2, …, tm.

And X( t1 + k ) , X( t2+ k ) , … , X( tm+ k ) are the same. That is the joint distribution of any set of observations must be unaffected by shifting all times of the observations forward or backward by integer amount k.

**Weak stationary (Covariance stationary)**

A time series is said to be weak stationary, if its first and second moments are finite and do not change through time.

i.e. E [X (t)] =, a constant for all t.

V [X (t)] < ∞, is finite

Cov (, ) = (k)

A time series is said to be stationary if there is no systematic change in mean, if there is no systematic change in variance and if strictly periodic variations have been removed. Much of the probability theory of time series is concerned with stationary time series, and for this reason time series analysis often requires one to transform a non-stationary series into a stationary one so as to use this theory. For example, it may be of interest to remove the trend and seasonal variation from a set of data.

Applications of the time series models are twofold; to obtain an understanding of the underlying forces and structure that produced the observed data and to fit a model and proceed to forecasting, monitoring or even feedback and feed forward control. The three main assumptions of time series analysis are the following:

1. **The time series data must be dependent**

This assumption is tested using the Ljung–Box test as follows:

H0: The data are independently distributed (i.e. the correlations in the population from which

the sample is taken are 0, so that any observed correlations in the data result from

randomness of the sampling process).

H1: The data are not independently distributed; they exhibit serial correlation.

The test statistic is:

Where *n* is the sample size, is the sample autocorrelation at lag *k* and h is the number of lags being tested. Reject H0 if the p-value is less than 0.05. The data needs to exhibit serial correlation in order to perform time series analysis.

1. **The time series data must be stationary.**

The stationarity of the time series data is tested using the Augmented Dickey–Fuller test (ADF). It is a test for a unit root in a time series sample. The intuition behind the test is as follows. If the series y is stationary (or trend stationary), then it has a tendency to return to a constant (or deterministically trending) mean. Therefore large values will tend to be followed by smaller values (negative changes), and small values by larger values (positive changes). Accordingly, the level of the series will be a significant predictor of next period's change, and will have a negative coefficient. If, on the other hand, the series is integrated, then positive changes and negative changes will occur with probabilities that do not depend on the current level of the series; in a random walk, ‘where you are’ now does not affect which way ‘you will go next’. The testing procedure for the ADF test is applied to the model

where is the variable of interest, t is the time index and 𝜀𝑡 is the error term. 𝛼 is a constant, 𝛽 the coefficient on a time trend and γ the lag order of the autoregressive process. By including lags of the order γ the ADF formulation allows for higher-order autoregressive processes. This means that the lag length γ has to be determined when applying the test. One possible approach is to test down from high orders and examine the t-values on coefficients. The unit root test is then carried out under the null hypothesis γ = 0 against the alternative hypothesis of γ < 0. Once a value for the test statistic (𝐴𝐷𝐹= 𝛾̂(𝛾̂)) is computed it can be compared to the relevant critical value for the Dickey–Fuller Test. If the test statistic is less (this test is non-symmetrical so we do not consider an absolute value) than the (larger negative) critical value, then the null hypothesis of γ = 0 is rejected and no unit root is present. The null hypothesis to be tested is:

H0: the data are not stationary and needs to be differenced to make it stationary.

H1: the data are stationary and doesn’t need to be differenced.

Reject the null hypothesis if the calculated p value is less than 0.05.

**Time Series Model**

Some important time series models which play important role in modern time series analysis are,

**Moving Average (MA) Process**

A time series {, t= 0, +1, + 2,….} is said to be a moving average of order q can be expressed as

Here { } is a white noise process and are parameters.

**Auto Regressive (AR) Process**

A time series { , t= 0, +1 , + 2 ,…. } is said to be a autoregressive of order p can be expressed as

Here are parameters and { } is a white noise.

**Auto Regressive Moving Average (ARMA) Process**

A time series { , t=0, ±1, ±2...} is said to be a autoregressive moving average of order (p, q) denoted as ARMA (p, q) can be expressed as,

Here the ... and ... are constants and { } is white noise process.

Using backward shift operator we can write it as

Ф (B) 𝑋𝑡 = ϴ (B)

Where ф (B) =

And ϴ (B) =

**Auto Regressive Integrated Moving Average (ARIMA) Process**

Let {𝑋𝑡, t ∈ I} denotes a non-stationary time series, non-stationary due to trend component. Let denotes the difference operator and let original time series {𝑋𝑡} is differenced‘d’ times so that the resulting series is stationary.

i.e., Let 𝑍𝑡 = 𝛻𝑑 𝑋𝑡

Suppose 𝑍𝑡 follows ARMA (p, q) process the original series {𝑋𝑡, t ∈ I} is said to be autoregressive integrated moving average process of order (p, d, q).

Ф (B) (1−𝐵) = ϴ (B)

This is the representation of ARIMA (p, d, q) process

**Seasonal Arima Model**

Seasonality in a time series is a regular pattern of changes that repeats over S time periods, where S defines the number of time periods until the pattern repeats again.

In a seasonal ARIMA model, seasonal AR and MA terms predict Xt using data values and errors at times with lags that are multiples of S(the span of the seasonality). With monthly data (S=12), a seasonal first order autoregressive model would use to predict. A seasonal second order autoregressive model would use and to predict . A seasonal first order MA (1) model would use and .

**Seasonal Time Series with Deterministic Component:**

Time series data with deterministic seasonal components can be eliminated simply by differencing the series with appropriate distance.

**Differencing:** it may be necessary to examine differenced data when we have seasonality. Seasonality usually causes the series to be non-stationary because the average values at some particular times with in the seasonal span may be different than the average values at other times. Seasonal differencing is defined as a difference between a value and a value with lag that is a multiple of S.

Seasonal difference is

**Non-seasonal differencing:** If trend is present in the data we may also need non-seasonal differencing. Often a first difference will “detrend” the data. That is, we use is in the presence of trend.

**Differencing for Trend and Seasonality:** When both Trend and Seasonality are present, we may need to apply both a non-seasonal first difference and a seasonal difference.

That is, we may need to examine the ACF and PACF of

Removing trend does not mean that we have removed the dependency. We may have removed the mean, part of which may include a periodic component. In some one ways we are breaking the dependency down into recent things that have happened and long-range things that have happened.

Non-seasonal behavior will still matter with seasonal data, it is likely that short run non-seasonal components will still contribute to the model. We will have to look at the ACF and PACF behavior over the first few lags to assess what non-seasonal terms might work in the model. The SARIMA model incorporates both non-seasonal and seasonal factors in a multiplicative model.

**Stochastic Seasonal Time series:**

Time series with seasonal effect do not necessarily recur with exact pattern. They often occur with random intensity, although they reappear at the same time during the year. Such seasonal influences are statistically distributed over time. Such seasonal influence on time series data is referred to as stochastic seasonal effect. In such case future seasonal effect can be realized from the past observed values. The stochastic seasonal models in general can be expressed by three different classes of models namely MA model, AR model, mixed ARMA model.

A seasonal ARIMA model is defined as ARIMA (p, d, q) (P, D, Q) s model.

where, P - Number of seasonal autoregressive terms (SAR)

D - Number of seasonal difference

Q - Number of seasonal moving average terms (SMA)

p - Non-seasonal AR order

q – Non-seasonal MA order

d - Non-seasonal differencing

The seasonal autoregressive integrated moving average model of Box and Jenkins (1970) is given by

The Non-seasonal components are,

The Seasonal components are,

⏀ () = 1 − ⏀1 − ⋯ − ⏀𝑝

() = 1 + 𝛩1 − ⋯ − 𝛩𝑞

**Determination of the orders d, D, p, P, q and Q**

Seasonal differencing is necessary to remove the seasonal trend. If there is secular trend non-seasonal differencing will be necessary. To avoid undue model complexity it has been advised that orders of differencing d and D should add up to at most 2 (i.e. d + D < 3). If the ACF of the differenced series has a positive spike at the seasonal lag then a seasonal AR component is suggestive. If it has a negative spike then a seasonal MA component is suggestive.

Box and Jenkins (1976) and Madsen (2008) have given a catalogue of seasonal models and their covariance structures for possible use for modeling. As already mentioned above, an AR (p) model has a PACF that truncates at lag p and an MA (q)) has an ACF that truncates at leg q. In practice ± 2/√n where n is the sample size are the non-significance limits for both functions.

**Auto Correlation Function (ACF)**

Autocorrelations referred to the observations in a time series are related to each other and is measured by the simple correlation between current observation () and observations from the p periods before the current one (). That is for a given series autocorrelation at lag p is the correlation between the pair () and is given by,

**Partial Autocorrelation Function (PACF)**

Partial autocorrelation are used to measure the degrees of association between and when y-effects at other time lags 1, 2, …, p-1 are removed.

The auto covariance coefficient at lag k measures the covariance between two values X (t) and X (t+k) separated by intervals of time. The set of autocovariance function of the process in an AR process of order p, all AC’s of order more than p are zero. This is because is not depending upon its past values for lag more than p.

We can write partial autocorrelation is a tool used for determining the order of the AR process. Therefore in time series analysis partial autocorrelations of lower order of important and it is the highest order of the significant partial autocorrelation which asserts the independent variable to be included in the multiple regression.

Partial autocorrelation function is a device which exploits the fact that whereas AR(p) process has an ACF which is infinite extent, it can be by its varying nature described in terms of p non zero components of autocorrelation.

**Definition:**

For a stationary process {, 𝑡 ∈ I} the partial autocorrelation function (PACF) is defined by,

Where

And is same as except last column of is replaced by the vector. Now the sample partial autocorrelation is defined as,

where

Where denotes the ACF =

is same as except that the last column of is replaced.

Accordingly we get sample PACF as ⏀11 =

Depends only on lag-k for all t. Where (k) is called auto covariance Coefficient.

**AKAIKE INFORMATION CRITERIA**

The Akaike Information Criteria (AIC) is a measure of the relative quantity of a statistical model for a given set of data. AIC provide a means for model selection. AIC is founded on information theory: it offers a relative estimate of the information lost when a given model is used to represent the process that generates the data. AIC penalizes a model with larger number of parameters and is defined as,

AIC = - 2ln (L) + 2p

Where ln (L) denotes the fitted log likelihood and p denotes the number of parameters. Given the set of candidate models for the data, the preferred model is one with the minimum AIC value.

**Ljung–Box Statistic:**

The Ljung–Box test may be defined as:

**H0:** The data are independently distributed (i.e. the correlations in the population from which the

sample is taken are 0, so that any observed correlations in the data result from randomness

of the sampling process).

**H1:** The data are not independently distributed; they exhibit serial correlation.

The test statistic is:

where *n* is the sample size, is the sample autocorrelation at lag *k*, and *h* is the number of lags being tested. Under the statistic Q follows.

For significance level α, the critical region for rejection of the hypothesis of randomness is:

Q >

Where is the 1-α quantile of the chi-squared distribution with *h* degrees of freedom.

The Ljung–Box test is uncommonly used in autoregressive integrated moving average (ARIMA) modeling. Note that it is applied to the residuals of a fitted ARIMA model, not the original series, and in such applications the hypothesis actually being tested is that the residuals from the ARIMA model have no autocorrelation. When testing the residuals of an estimated ARIMA model, the degrees of freedom need to be adjusted to reflect the parameter estimation. For example, for an ARIMA (p, 0, q) model, the degrees of freedom should be set to *h-p-q*

**Box-Pierce Statistic:**

The Box-Pierce test statistic is,

It uses the same critical region as defined above.

Simulation studies have shown that the Ljung–Box test statistics is better than Box-Pierce test.

**FORECASTING**

Forecasting the future values of an observed time series is an important problem in many areas, including economics, production planning, sales forecasting and stock control. Suppose we have an observed time series then problem is to estimate ,or more generally .the prediction .made at the time N of the value q steps ahead will be denoted by .The integer q is called the lead time. A wide variety of different forecasting procedures is available and it is important to realize that no single method is universally applicable rather, the analyst must choose the procedure that is most appropriate for a given set of conditions. It is also worth bearing in mind that forecasting is a form of extrapolation, with all the dangers that it entails. Forecasts are conditional statements about the future based on specific assumptions. Thus forecasts are not cared and the analyst should always be prepared to modify them as necessary in the light of any external information. For long term forecasting, it can be helpful to produce a range of forecasts based on different sets of assumptions to that “scenarios” can be explored.

* + 1. **Subjective forecast**

Subjective forecast can be made on a subjective basis using judgment, institutions, commercial knowledge and any other relevant information. Methods range widely from bold freehand extrapolation to the Delphi technique, in which a group of forecasters tries to obtain consensus forecast with controlled feedback of other ‘analysts’ predictions and opinions as well as other relevant information

* + 1. **Univariate forecast**

Univariate forecast can be based entirely on past observations in given time series, by fitting a model to the data and extra poling. This type of forecasting is called ‘native’ or ‘projection’ methods.

* + 1. **Multivariate forecast**

Multivariate forecast can be made by taking observations on other variables into account. Exponential smoothing may easily generalized to deal with the time series containing trend and seasonal variation. The resulting procedure is usually referred to as the Holt –winters procedures and is described by Winters (1960) and Chatfield (1978a).Trend and seasonal terms are introduced which are also updated by exponential smoothing.

**Features of Forecasting**

Peculiarities, characteristics or features of forecasting as follows:-

1. Forecasting in concerned with future events.
2. It shows the probability of happening of future events.
3. It analysis past and present data.
4. It uses personal observations.

**Importance Of Forecasting**

Merits, significance or importance of forecasting involves following points:-

* Forecasting provides relevant and reliable information about the past and present values and the likely future events .This is necessary for sound planning.
* It gives confidence to the managers for important decisions.
* It is the basis for making planning premises.
* It keeps managers active and alert to face the challenges of future events and the changes in the environment.

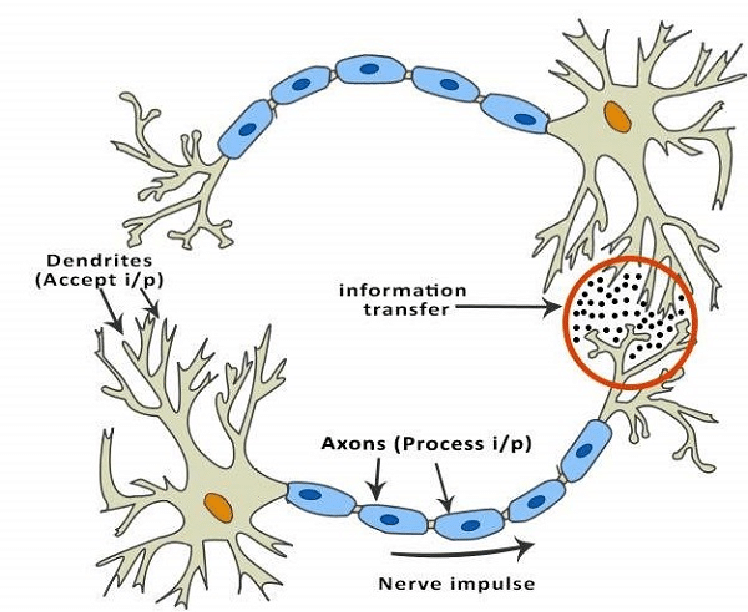
**Limitations of Forecasting**

Demerits, criticism or limitations of forecasting involves following steps:-

* The collection and analysis of data about the past, present and future involves a lot of time and money. Therefore, managers have to balance the cost of forecasting with its benefits. Many small firms don’t do forecasting because of high cost.
* Forecasting can only estimate the future events. It cannot guarantee that these events will take place in the future .Long term forecasts will be less accurate as compared to short term forecast.
* Forecasting is based on certain assumptions. If these assumptions are wrong, the forecasting will be wrong. Forecasting is based on past events. However, history may not repeat itself at all times.
* Forecasting requires proper judgments and skills on the part of managers. Forecasts may go wrong due to bad judgment and skills on the part of some of the managers. Therefore, forecasts are subject to human error.

**2.7 Multi-Layer Perceptions - Neural Networks(MLP-NN)**

The inventor of the first neurocomputer, Robert Hecht-Nielsen (2005), defines a neural network as: "...a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs. The idea of ANNs is based on the belief that working of human brain by making the right connections, can be imitated using silicon and wires as living neurons and dendrites. The human brain is composed of 100 billion nerve cells called neurons. They are connected to other thousand cells by Axons. Stimuli from external environment or inputs from sensory organs are accepted by dendrites. These inputs create electric impulses, which quickly travel through the neural network. A neuron can then send the message to other neuron to handle the issue or does not send it forward. ANNs are composed of multiple nodes, which imitate biological neurons of human brain. The neurons are connected by links and they interact with each other.



Biological neurons of human brain

The most widely used ANNs in forecasting problems are multi-layer perceptron’s (MLPs), which use a standard single hidden layer feed forward Network (SLFN) (Zang *et al*, 1998 ). The model is characterized by a network of three layers, viz. input, hidden and output layer, connected by acyclic links. There may be more than one hidden layer. processing elements. The nodes in various layers are also known as Processing elements. MLP-NN is the most popular ANN models for time series forecasting applications. Figure 3 shows a typical three-layer MLPNN used for forecasting purposes. The input nodes are the previous lagged observations, while the output provides the forecast for the future values. Hidden nodes with appropriate nonlinear transfer functions are used to process the information received by the input nodes.













Output Layer

 (Dependent Variable)

Input Layer

(Lag Dependent Variables)

Hidden Layer

(q unit neurons)

*Architecture of Multi-Layer Perception neural network model with a single hidden layer*

*(MLP-NN)*

The model of MLP-NN in figure can be written as,

,

where *p* is the number of input nodes, *q* is the number of hidden nodes,

*f* is a sigmoid transfer function such as the logistic:

is applied as the nonlinier activation function.

is a vector of weights from the hidden to output nodes and

are weights from the input to hidden nodes.

is the random shock; and are the bias term.

MLP-NN model in fact performs a non-linear functional mapping from the past observations of the time series to the future value that is,

,

where *W* is a vector of all parameters and *f* is a function determined by the network structure and connection weights (Zang et all, 1998 ). To estimate the connection weights, non-linear least square procedures are used, which are based on the minimization of the error function,

Here Ψ is the space of all connection weights.

**Input, hidden and output nodes:**

Notice that input nodes do not have activation functions. Thus, they are little more than placeholders. The input is simply weighted and summed. Furthermore, the size of input and output vectors will be the same if the neural network has nodes that are both input and output. Hidden nodes have two important characteristics. First, they only receive input from the other nodes, such as input or preceding hidden nodes. Second, they only output to other nodes, either as output or other, following hidden nodes.

Hidden nodes are not directly connected to the incoming data or to the eventual output. They are often grouped into fully connected hidden layers.

Another reason why additional hidden layers seemed to be a problem was that they would require a very extensive training set to be able to compute weights for the network. Before deep learning, the former situation was actually a problem, since deep learning means networks of several hidden layers. Although networks of one or two hidden layers are able to learn “everything” in theory, deep learning facilitates a more complex representation of patterns in the data.

**Back propagation:**

Back propagation is a method used in artificial neural networks to calculate a gradient that is needed in the calculation of the weights to be used in the network. It is commonly used to train deep neural networks, a term referring to neural networks with more than one hidden layer.

Back propagation is a special case of an older and more general technique

called automatic differentiation. In the context of learning, back propagation is commonly used by the gradient descent optimization algorithm to adjust the weight of neurons by calculating the gradient of the loss function. This technique is also sometimes called backward propagation of errors, because the error is calculated at the output and distributed back through the network layers.

***Back propagation algorithm***: Back propagation is a neural network learning algorithm. Back propagation learns by iteratively processing a data set of training tuples, comparing the network’s prediction for each tuple with the actual known target value. The target value may be the known class label of the training tuple (for classification problems) or a continuous value (for numeric prediction). For each training tuple, the weights are modified so as to minimize the mean-squared error between the network’s prediction and the actual target value. These modifications are made in the “backwards” direction (i.e., from the output layer) through each hidden layer down to the first hidden layer. In general the weights will eventually converge, and the learning process stops.

**2.8 Extreme Learning Machine (ELM)**

Extreme learning machines are [feed forward neural networks](https://en.wikipedia.org/wiki/Feedforward_neural_network) for [classification](https://en.wikipedia.org/wiki/Statistical_classification), [regression](https://en.wikipedia.org/wiki/Regression_analysis), [clustering](https://en.wikipedia.org/wiki/Cluster_analysis), [sparse approximation](https://en.wikipedia.org/wiki/Sparse_approximation), compression and [feature learning](https://en.wikipedia.org/wiki/Feature_learning) with a single layer or multiple layers of hidden nodes, where the parameters of hidden nodes (not just the weights connecting inputs to hidden nodes) need not be tuned. These hidden nodes can be randomly assigned and never updated (i.e. they are [random projection](https://en.wikipedia.org/wiki/Random_projection) but with nonlinear transforms), or can be inherited from their ancestors without being changed. In most cases, the output weights of hidden nodes are usually learned in a single step, which essentially amounts to learning a linear model. The name "extreme learning machine" (ELM) was given to such models by its main inventor Guang-Bin Huang.

According to their creators, these models are able to produce good generalization performance and learn thousands of times faster than networks trained using [back propagation](https://en.wikipedia.org/wiki/Backpropagation)

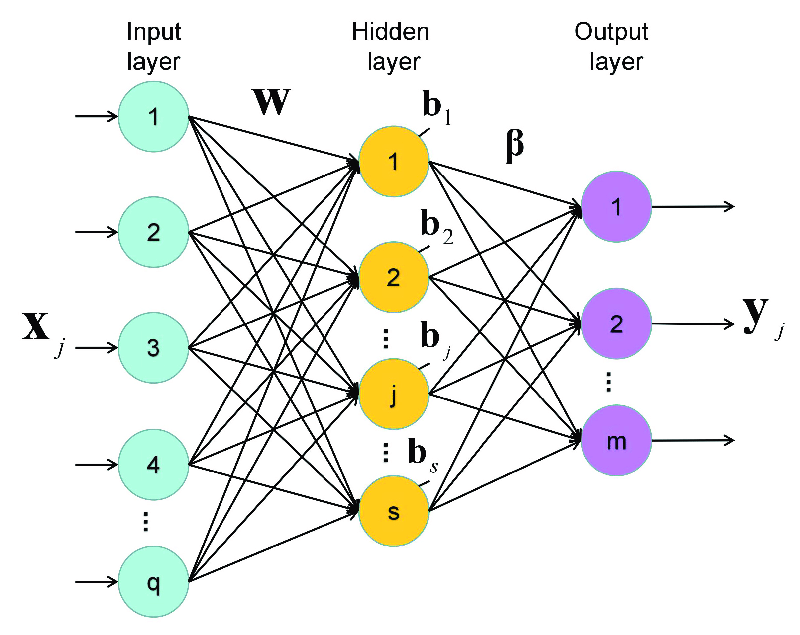


Figure a standard single hidden layer feed forward neural network (SLFN) in ELM

Given a single hidden layer of ELM, suppose that the output function of the i*-*th hidden node is, where and are the parameters of the *i-*th hidden node.

The output function of the ELM for SLFNs with L hidden nodes is:

, where is the output weight of the *i-*th hidden node.

H(x) = is the hidden layer output mapping of ELM. Given {\displaystyle N}N training samples, the hidden layer output matrix {\displaystyle \mathbf {H} }H of ELM is given as:

H= =

And T is the training data target matrix T=

Generally speaking, ELM is a kind of regularization neural networks but with non-tuned hidden layer mappings (formed by either random hidden nodes, kernels or other implementations), its objective function is:

Minimize:

Where

Different combinations of {\displaystyle \sigma \_{1}}, {\displaystyle \sigma \_{2}}, {\displaystyle p}  and {\displaystyle q} can be used and result in different learning algorithms for regression, classification, sparse coding, compression, feature learning and clustering

As a special case, a simplest ELM training algorithm learns a model of the form (for single hidden layer sigmoid neural networks):

Whereis the matrix of input-to-hidden-layer weights,is an activation function, and is the matrix of hidden-to-output-layer weights. The algorithm proceeds as follows:

1. Fillwith random values (e.g, [Gaussian random noise](https://en.wikipedia.org/wiki/Gaussian_noise))
2. Estimateby [least-squares fit](https://en.wikipedia.org/wiki/Least-squares_fit) to a matrix of response variables computed using the [pseudo inverse](https://en.wikipedia.org/wiki/Moore%E2%80%93Penrose_pseudoinverse) ⋅+, given a [design matrix](https://en.wikipedia.org/wiki/Design_matrix) X:

**CHAPTER 3**

**ANALYSIS AND DISCUSSIONS**

**Analysis and Interpretation of Precipitation in Udupi:**

The data used in the study are monthly average of different factors affecting Precipitation from January 1981 to May 2022. A 12 month observations of 2021 are considered for sample forecast accuracy purpose and remaining observations from 1981 to 2020 are used for model building purpose.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **n** | **Mean** | **Median** | **Minimum** | **Maximum** | **Variance** | **Std. Deviation** | **Skewness** | **Kurtosis** |
| PS | 497 | 96.88 | 96.87 | 96.53 | 97.27 | 0.024585 | 0.16 | 0.22 | -0.78 |
| TS | 497 | 26.15 | 25.29 | 22.5 | 31.96 | 4.32199 | 2.08 | 0.86 | -0.54 |
| T2M | 497 | 25.23 | 24.71 | 21.94 | 29.65 | 2.54571 | 1.6 | 0.73 | -0.51 |
| T2MDEW | 497 | 20.34 | 21.25 | 12.46 | 23.57 | 6.04466 | 2.46 | -0.72 | -0.64 |
| T2MWET | 497 | 22.79 | 23.21 | 17.99 | 25.76 | 2.40251 | 1.55 | -0.7 | -0.3 |
| QV2M | 497 | 15.77 | 16.54 | 9.83 | 18.98 | 4.89331 | 2.21 | -0.56 | -0.97 |
| RH2M | 497 | 77.09 | 76.81 | 49.69 | 92.38 | 119.2144 | 10.92 | -0.15 | -1.36 |
| PRECTOTCORR | 497 | 8.94 | 2.43 | 0 | 49.21 | 143.8879 | 12 | 1.3 | 0.58 |
| WS2M | 497 | 1.73 | 1.54 | 0.96 | 3.06 | 0.24235 | 0.49 | 0.83 | -0.48 |

**Table 1: Descriptive Statistics**

Mean, Median, Minimum, maximum, Variance, kurtosis and skewness of continues variables are given in the above table. From the above table, all variables under study are platykurtic. From skewness we can observe that Dew temperature, Wet temperature, Specific humidity and Relative humidity are negatively skewed and all other variables are positively skewed.

**Multiple Linear Regression**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Estimate | Std. Error | t value | Pr (> |t|) | Decision |
| Intercept | -93.2752 | 26.9135 | -3.466 | 0.000575 | Significant |
| TS | -5.3605 | 1.1106 | -4.827 | 1.86 10^-6 | Significant |
| T2MDEW | -9.4559 | 1.4468 | -6.536 | 1.59 10^-10 | Significant |
| QV2M | 6.9675 | 1.8895 | 3.688 | 0.000252 | Significant |
| RH2M | 0.998 | 0.2578 | 3.871 | 0.000123 | Significant |
| T2M | 8.8429 | 1.8724 | 4.723 | 3.04 10^-6 | Significant |
| WS2M | 14.3804 | 0.661 | 21.754 | 2 \* 10^-16 | Significant |

Residual standard error: 4.605 on 490 degree of freedom

Multiple R-squared: 0.8544 Adjusted R-squared: 0.8526

From the above table we can observe that Surface temperature, Dew temperature at 2 meters, Specific Humidity at 2 Meters, Relative Humidity at 2 Meters, Temperature at 2 Meters and Wind speed at 2 Meters are the significance variables for Precipitation in our dataset.

The value is 0.8544, indicates that about 85.44% of variation on precipitation can be explained by Surface temperature, Dew temperature at 2 meters, Specific Humidity at 2 Meters, Relative Humidity at 2 Meters, Temperature at 2 Meters and Wind speed at 2 Meters.

**Checking for Multicolinearity, Auto-correlation and Heteroscedasticity.**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | PS | TS | T2M | T2MDEW | T2MWET | QV2M | RH2M | PRECTOTCORR | WS2M |
| PS | 1 | -0.11 | -0.26 | -0.81 | -0.78 | -0.83 | -0.61 | -0.66 | -0.69 |
| TS | -0.11 | 1 | 0.98 | -0.06 | 0.46 | -0.05 | -0.58 | -0.37 | -0.17 |
| T2M | -0.26 | 0.98 | 1 | 0.13 | 0.62 | 0.13 | -0.43 | -0.24 | -0.07 |
| T2MDEW | -0.81 | -0.06 | 0.13 | 1 | 0.86 | 0.86 | 0.84 | 0.66 | 0.6 |
| T2MWET | -0.78 | 0.46 | 0.62 | 0.86 | 1 | 0.86 | 0.44 | 0.4 | 0.44 |
| QV2M | -0.83 | -0.05 | 0.13 | 0.86 | 0.86 | 1 | 0.84 | 0.68 | 0.62 |
| RH2M | -0.61 | -0.58 | -0.43 | 0.84 | 0.44 | 0.84 | 1 | 0.76 | 0.62 |
| PRECTOTCORR | -0.66 | -0.37 | -0.24 | 0.66 | 0.4 | 0.68 | 0.76 | 1 | 0.86 |
| WS2M | -0.69 | -0.17 | -0.07 | 0.6 | 0.44 | 0.62 | 0.62 | 0.86 | 1 |

Table 2: The Correlation between the variables

Since there is high correlation between the variables, we suspect that there is a presence of multicollinearity in the data. All the variables causes multicollinearity in the data.

The variance inflation factor for each variable are listed bellow

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TS | T2MDEW | QV2M | RH2M | T2M | WS2M |
| 124.6657 | 295.8851 | 408.55 | 185.31 | 208.719 | 2.47662 |

Observe that the variance inflation factor is more than 5 for Surface temperature, Dew temperature at 2 meters, Specific Humidity at 2 Meters, Relative Humidity at 2 Meters and Temperature at 2 Meters, which indicates that those variables are highly correlated with other explanatory variables.

**Durbin Watson test for Autocorrelation**

: The residuals are not autocorrelated.

: The residuals are correlated.

= 1.829 = 1.877

d = 1.722.

Since d < , we reject the null hypothesis of no autocorrelation and accept that there is a positive autocorrelation of first order.

Now since we have autocorrelation in our dataset, we used the Cochrane-Orcutt Procedure to remove autocorrelation from the dataset.

Now Durbin Watson statistic after the removal of autocorrelation is

d\* = 1.971

Since < d\* < (4-) we accept the null hypothesis of no autocorrelation.

The variance inflation factor for each variable after the removal of autocorrelation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TS | PS | T2MWET | QV2M | RH2M | T2M | WS2M |
| 108.0723 | 4.39748 | 434.4166 | 379.9197 | 158.2563 | 246.3568 | 2.53248 |

Observe that the variance inflation factor is more than 5 for Surface temperature, Wet temperature at 2 meters, Specific Humidity at 2 Meters, Relative Humidity at 2 Meters and Temperature at 2 Meters, which indicates that those variables are highly correlated with other explanatory variables.

**Gold-Feld Quant Test for Heteroscedasticity:**

: The variance for the errors are equal.

: The variance for the error are not equal.

If calculated F > Critical F value, we reject .

GQ = 1.7895 F-critical = 0.521

GQ Feld p value is 0.521 (>0.05), therefore we accept Ho and conclude that there is no Heteroscedasticity.

**Multiple Linear Regression (After removing Autocorrelation and Heteroscedasticity)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Estimate | Std. Error | t value | Pr (> |t|) | Decision |
| Intercept | 510.5847 | 295.6237 | 1.727 | 0.0851 | Significant |
| TS | -6.0824 | 1.3198 | -4.608 | 5.75E-06 | Significant |
| PS | -7.4273 | 3.5289 | -2.105 | 0.0361 | Significant |
| QV2M | 4.2692 | 2.4031 | 1.777 | 0.0765 | Significant |
| RH2M | 1.3921 | 0.322 | 4.323 | 2.03E-05 | Significant |
| T2M | 53.2607 | 26.7076 | 1.994 | 0.0469 | Significant |
| WS2M | 12.9753 | 0.8582 | 15.119 | 2.00E-16 | Significant |

Residual standard error: 4.426 on 339 degrees of freedom.

Multiple R-squared: 0.8385 Adjusted R-squared: 0.8347

From the above table we can observe that Surface temperature, Surface Temperature, Specific Humidity at 2 Meters, Relative Humidity at 2 Meters, Temperature at 2 Meters and Wind speed at 2 Meters are the significance variables for Precipitation in our dataset.

The value is 0.8385, indicates that about 83.85% of variation on precipitation can be explained by Surface temperature, Surface Temperature, Specific Humidity at 2 Meters, Relative Humidity at 2 Meters, Temperature at 2 Meters and Wind speed at 2 Meters.

**Ridge Regression**

Selection of turning parameter:-

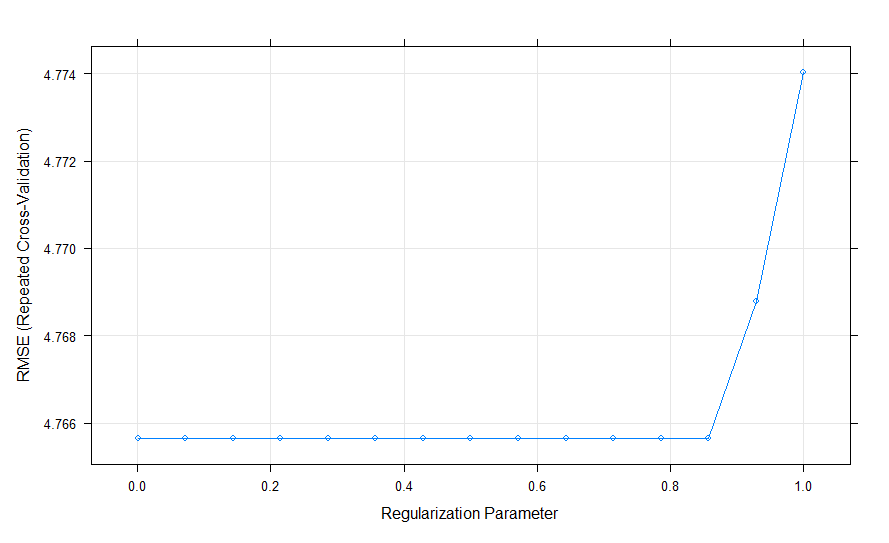


Fig. 1 Selection of Turning points in ridge regression

349 samples

8 predictor

No pre-processing

Resampling: Cross-Validated (10 fold, repeated 5 times)

Summary of sample sizes: 313, 313, 315, 313, 314, 313, ...

Resampling results across tuning parameters:

|  |  |  |  |
| --- | --- | --- | --- |
| lambda | RMSE | Rsquared | MAE |
| 0.001 | 4.765643 | 0.81652 | 3.601874 |
| 0.072357 | 4.765643 | 0.81652 | 3.601874 |
| 0.143714 | 4.765643 | 0.81652 | 3.601874 |
| 0.215071 | 4.765643 | 0.81652 | 3.601874 |
| 0.286429 | 4.765643 | 0.81652 | 3.601874 |
| 0.357786 | 4.765643 | 0.81652 | 3.601874 |
| 0.429143 | 4.765643 | 0.81652 | 3.601874 |
| 0.5005 | 4.765643 | 0.81652 | 3.601874 |
| 0.571857 | 4.765643 | 0.81652 | 3.601874 |
| 0.643214 | 4.765643 | 0.81652 | 3.601874 |
| 0.714571 | 4.765643 | 0.81652 | 3.601874 |
| 0.785929 | 4.765643 | 0.81652 | 3.601874 |
| 0.857286 | 4.765643 | 0.81652 | 3.601874 |
| 0.928643 | 4.768778 | 0.816403 | 3.603624 |
| 1 | 4.774033 | 0.816127 | 3.605817 |

Tuning parameter 'alpha' was held constant at a value of 0

RMSE was used to select the optimal model using the smallest value.

The final values used for the model were alpha = 0 and lambda = 0.8572857.

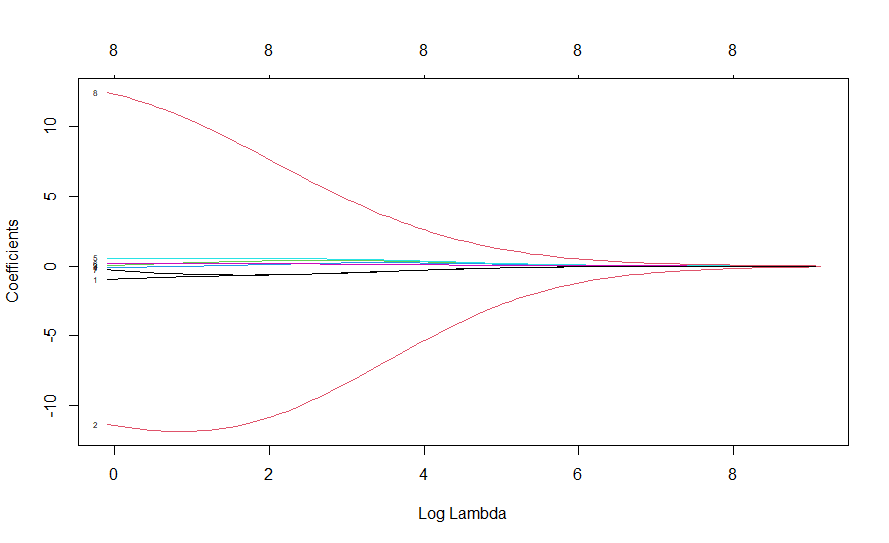


Fig. 2 Plot of Log Lambda v/s Coefficients

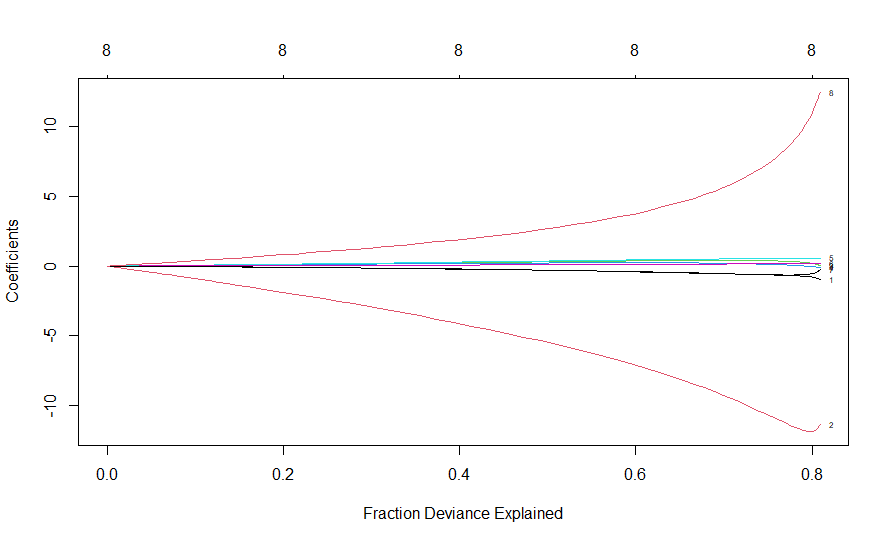


Fig. 3 Plot of Fraction Deviance Explained v/s Coefficients

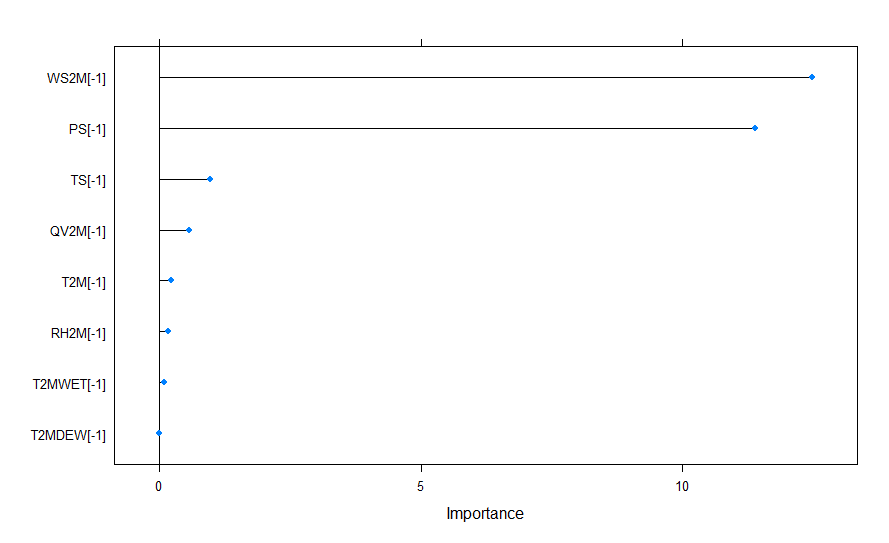


Fig. 4 Plot of Variable Importance

From the above graph, Wind speed at 2 meters is the most important variable for Precipitation followed by surface pressure, Surface Temperature, Specific Humidity Temperature at 2 meters, Temperature at 2 meters, Relative Humidity at 2 meters and Wet Bulb Temperature at 2 meters.

**Lasso Regression**

Selection of turning parameter:-

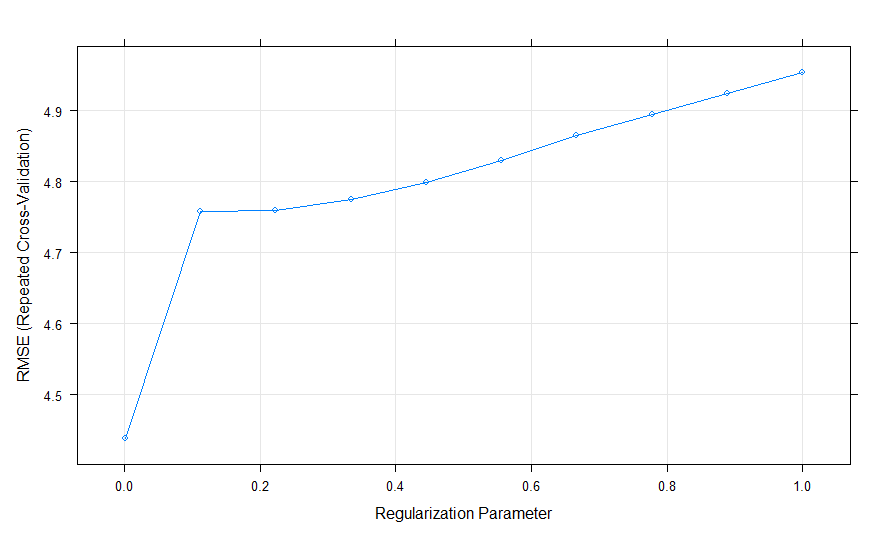


Fig. 5 Selection of Turning points in Lasso regression

349 samples

8 predictor

No pre-processing

Resampling: Cross-Validated (10 fold, repeated 5 times)

Summary of sample sizes: 313, 314, 313, 313, 312, 313, ...

Resampling results across tuning parameters:

Surface Temperature

|  |  |  |  |
| --- | --- | --- | --- |
| lambda | RMSE | Rsquared | MAE |
| 0.001 | 4.43825 | 0.838686 | 3.342969 |
| 0.112 | 4.757544 | 0.814424 | 3.599757 |
| 0.223 | 4.759133 | 0.814751 | 3.582878 |
| 0.334 | 4.775816 | 0.81408 | 3.577328 |
| 0.445 | 4.799747 | 0.813078 | 3.575761 |
| 0.556 | 4.830347 | 0.811777 | 3.577587 |
| 0.667 | 4.864944 | 0.810358 | 3.583485 |
| 0.778 | 4.895214 | 0.8095 | 3.587534 |
| 0.889 | 4.923862 | 0.80899 | 3.589124 |
| 1 | 4.954234 | 0.808506 | 3.592687 |

Tuning parameter 'alpha' was held constant at a value of 1

RMSE was used to select the optimal model using the smallest value.

The final values used for the model were alpha = 1 and lambda = 0.001.

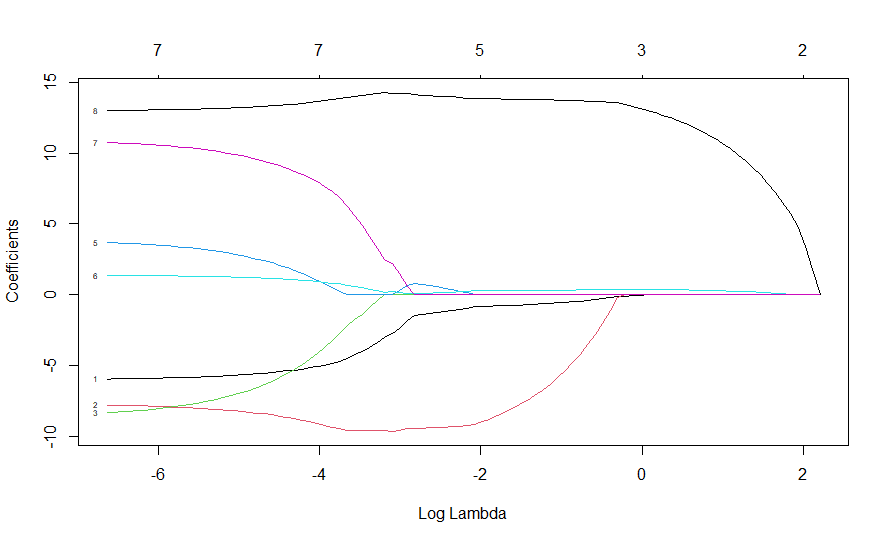


Fig. 6 Plot of Log Lambda v/s Coefficients

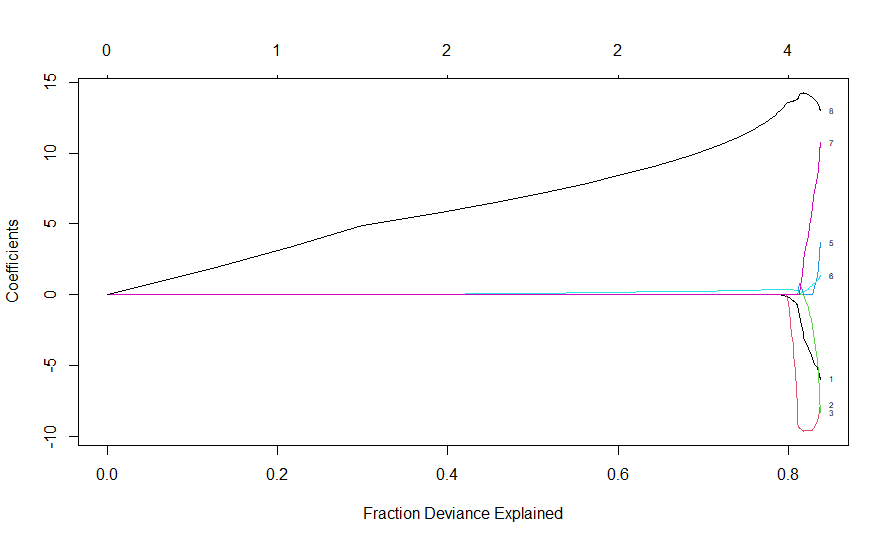


Fig. 7 Plot of Fraction Deviance Explained v/s Coefficients

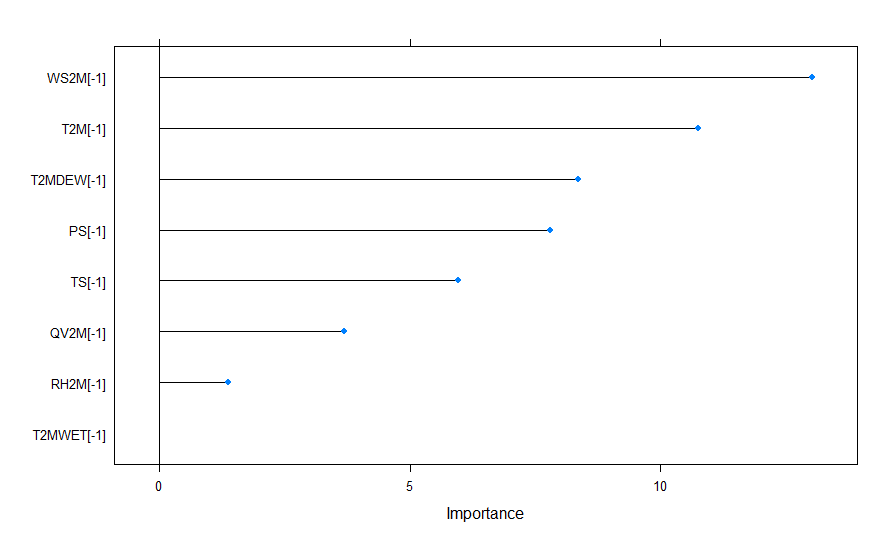


Fig. 8 Plot of Variable Importance

From the above graph, Wind speed at 2 meters is the most important variable for Precipitation followed by Temperature at 2 meters, Dew/Frost Point at 2 Meters, surface pressure, Surface Temperature, Specific Humidity Temperature at 2 meters and Relative Humidity at 2 meters.

**Elastic Net Regression**

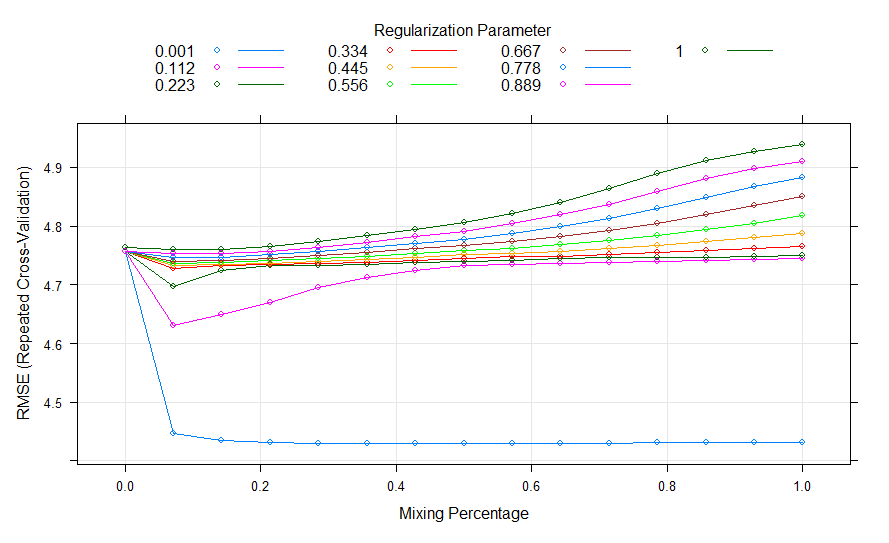


Fig. 9 Plot of Mixing Percentage v/s RMSE

349 samples

8 predictor

No pre-processing

Resampling: Cross-Validated (10 fold, repeated 5 times)

Summary of sample sizes: 313, 312, 313, 314, 315, 312, ...

Resampling results across tuning parameters:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| alpha | lambda | RMSE | Rsquared | MAE |
| 0 | 0.001 | 4.755989 | 0.81929 | 3.605471 |
| 0 | 0.112 | 4.755989 | 0.81929 | 3.605471 |
| 0 | 0.223 | 4.755989 | 0.81929 | 3.605471 |
| 0 | 0.334 | 4.755989 | 0.81929 | 3.605471 |
| 0 | 0.445 | 4.755989 | 0.81929 | 3.605471 |
| 0 | 0.556 | 4.755989 | 0.81929 | 3.605471 |
| 0 | 0.667 | 4.755989 | 0.81929 | 3.605471 |
| 0 | 0.778 | 4.755989 | 0.81929 | 3.605471 |
| 0 | 0.889 | 4.756153 | 0.819285 | 3.605533 |
| 0 | 1 | 4.763937 | 0.818887 | 3.608563 |
| 0.071429 | 0.001 | 4.446958 | 0.841771 | 3.340783 |
| 0.071429 | 0.112 | 4.63079 | 0.828146 | 3.498585 |
| 0.071429 | 0.223 | 4.697254 | 0.823148 | 3.564572 |
| 0.071429 | 0.334 | 4.727931 | 0.820872 | 3.594312 |
| 0.071429 | 0.445 | 4.733372 | 0.820579 | 3.595785 |
| 0.071429 | 0.556 | 4.736067 | 0.820531 | 3.592721 |
| 0.071429 | 0.667 | 4.73997 | 0.820424 | 3.590415 |
| 0.071429 | 0.778 | 4.746464 | 0.820141 | 3.59083 |
| 0.071429 | 0.889 | 4.7535 | 0.819841 | 3.591505 |
| 0.071429 | 1 | 4.760717 | 0.819554 | 3.591794 |
| 0.142857 | 0.001 | 4.434287 | 0.842713 | 3.341185 |
| 0.142857 | 0.112 | 4.649115 | 0.826769 | 3.5169 |
| 0.142857 | 0.223 | 4.724168 | 0.821093 | 3.592175 |
| 0.142857 | 0.334 | 4.732629 | 0.820554 | 3.596418 |
| 0.142857 | 0.445 | 4.734926 | 0.820535 | 3.59252 |
| 0.142857 | 0.556 | 4.738045 | 0.820501 | 3.588694 |
| 0.142857 | 0.667 | 4.742011 | 0.820448 | 3.585171 |
| 0.142857 | 0.778 | 4.746784 | 0.82038 | 3.581925 |
| 0.142857 | 0.889 | 4.752642 | 0.820272 | 3.579094 |
| 0.142857 | 1 | 4.760187 | 0.820064 | 3.577694 |
| 0.214286 | 0.001 | 4.431039 | 0.842943 | 3.342036 |
| 0.214286 | 0.112 | 4.670456 | 0.825142 | 3.53797 |
| 0.214286 | 0.223 | 4.732013 | 0.820512 | 3.598815 |
| 0.214286 | 0.334 | 4.73415 | 0.820487 | 3.594294 |
| 0.214286 | 0.445 | 4.736964 | 0.820467 | 3.589684 |
| 0.214286 | 0.556 | 4.74068 | 0.820439 | 3.58532 |
| 0.214286 | 0.667 | 4.74534 | 0.820398 | 3.581293 |
| 0.214286 | 0.778 | 4.750952 | 0.820344 | 3.577478 |
| 0.214286 | 0.889 | 4.757481 | 0.820275 | 3.573885 |
| 0.214286 | 1 | 4.764914 | 0.820193 | 3.570667 |
| 0.285714 | 0.001 | 4.430127 | 0.843018 | 3.343548 |
| 0.285714 | 0.112 | 4.69574 | 0.823206 | 3.563993 |
| 0.285714 | 0.223 | 4.733493 | 0.820425 | 3.597668 |
| 0.285714 | 0.334 | 4.736204 | 0.820387 | 3.592514 |
| 0.285714 | 0.445 | 4.739717 | 0.820357 | 3.587561 |
| 0.285714 | 0.556 | 4.744219 | 0.820326 | 3.582663 |
| 0.285714 | 0.667 | 4.749797 | 0.820287 | 3.578329 |
| 0.285714 | 0.778 | 4.756447 | 0.820239 | 3.574406 |
| 0.285714 | 0.889 | 4.764184 | 0.820178 | 3.570995 |
| 0.285714 | 1 | 4.772953 | 0.820108 | 3.567983 |
| 0.357143 | 0.001 | 4.429966 | 0.843048 | 3.344545 |
| 0.357143 | 0.112 | 4.712603 | 0.821912 | 3.581456 |
| 0.357143 | 0.223 | 4.735115 | 0.820334 | 3.596795 |
| 0.357143 | 0.334 | 4.738699 | 0.820263 | 3.591509 |
| 0.357143 | 0.445 | 4.743138 | 0.820211 | 3.586286 |
| 0.357143 | 0.556 | 4.748691 | 0.820162 | 3.581609 |
| 0.357143 | 0.667 | 4.755441 | 0.820111 | 3.577608 |
| 0.357143 | 0.778 | 4.763428 | 0.820053 | 3.574055 |
| 0.357143 | 0.889 | 4.772652 | 0.819987 | 3.57083 |
| 0.357143 | 1 | 4.783086 | 0.819912 | 3.567919 |
| 0.428571 | 0.001 | 4.4302 | 0.843024 | 3.346157 |
| 0.428571 | 0.112 | 4.724982 | 0.82097 | 3.593183 |
| 0.428571 | 0.223 | 4.737126 | 0.820218 | 3.596358 |
| 0.428571 | 0.334 | 4.741805 | 0.820101 | 3.591327 |
| 0.428571 | 0.445 | 4.747285 | 0.820021 | 3.586487 |
| 0.428571 | 0.556 | 4.75385 | 0.819964 | 3.582192 |
| 0.428571 | 0.667 | 4.761716 | 0.819913 | 3.578303 |
| 0.428571 | 0.778 | 4.770938 | 0.819865 | 3.574304 |
| 0.428571 | 0.889 | 4.781644 | 0.819805 | 3.570669 |
| 0.428571 | 1 | 4.79385 | 0.819727 | 3.567622 |
| 0.5 | 0.001 | 4.430266 | 0.843008 | 3.346948 |
| 0.5 | 0.112 | 4.73289 | 0.820383 | 3.60067 |
| 0.5 | 0.223 | 4.739484 | 0.820081 | 3.596569 |
| 0.5 | 0.334 | 4.745055 | 0.819933 | 3.591926 |
| 0.5 | 0.445 | 4.75079 | 0.819889 | 3.586395 |
| 0.5 | 0.556 | 4.757632 | 0.819885 | 3.580955 |
| 0.5 | 0.667 | 4.766666 | 0.819834 | 3.575984 |
| 0.5 | 0.778 | 4.77785 | 0.819737 | 3.572012 |
| 0.5 | 0.889 | 4.791025 | 0.819602 | 3.568757 |
| 0.5 | 1 | 4.806018 | 0.819439 | 3.566201 |
| 0.571429 | 0.001 | 4.4303 | 0.843005 | 3.347577 |
| 0.571429 | 0.112 | 4.734974 | 0.820245 | 3.601729 |
| 0.571429 | 0.223 | 4.742079 | 0.819927 | 3.597307 |
| 0.571429 | 0.334 | 4.747413 | 0.819836 | 3.591562 |
| 0.571429 | 0.445 | 4.753274 | 0.819838 | 3.584558 |
| 0.571429 | 0.556 | 4.762245 | 0.819743 | 3.579195 |
| 0.571429 | 0.667 | 4.773856 | 0.81958 | 3.574858 |
| 0.571429 | 0.778 | 4.787728 | 0.819376 | 3.571515 |
| 0.571429 | 0.889 | 4.803788 | 0.819134 | 3.568987 |
| 0.571429 | 1 | 4.82193 | 0.818857 | 3.567201 |
| 0.642857 | 0.001 | 4.430323 | 0.843003 | 3.348014 |
| 0.642857 | 0.112 | 4.736381 | 0.820157 | 3.601833 |
| 0.642857 | 0.223 | 4.744224 | 0.819806 | 3.597955 |
| 0.642857 | 0.334 | 4.748915 | 0.819802 | 3.589853 |
| 0.642857 | 0.445 | 4.756996 | 0.819695 | 3.583302 |
| 0.642857 | 0.556 | 4.768387 | 0.819486 | 3.5784 |
| 0.642857 | 0.667 | 4.782598 | 0.819214 | 3.574827 |
| 0.642857 | 0.778 | 4.799539 | 0.818877 | 3.572265 |
| 0.642857 | 0.889 | 4.819017 | 0.818484 | 3.570435 |
| 0.642857 | 1 | 4.840947 | 0.818037 | 3.570101 |
| 0.714286 | 0.001 | 4.430396 | 0.842998 | 3.34842 |
| 0.714286 | 0.112 | 4.738068 | 0.820051 | 3.602285 |
| 0.714286 | 0.223 | 4.745665 | 0.81974 | 3.597751 |
| 0.714286 | 0.334 | 4.751178 | 0.819715 | 3.58865 |
| 0.714286 | 0.445 | 4.761603 | 0.819486 | 3.582557 |
| 0.714286 | 0.556 | 4.775598 | 0.819155 | 3.578468 |
| 0.714286 | 0.667 | 4.792925 | 0.818731 | 3.57581 |
| 0.714286 | 0.778 | 4.813488 | 0.818214 | 3.573999 |
| 0.714286 | 0.889 | 4.837095 | 0.81761 | 3.57408 |
| 0.714286 | 1 | 4.863629 | 0.816917 | 3.576435 |
| 0.785714 | 0.001 | 4.430686 | 0.842977 | 3.348907 |
| 0.785714 | 0.112 | 4.739781 | 0.819943 | 3.602767 |
| 0.785714 | 0.223 | 4.746351 | 0.819726 | 3.596214 |
| 0.785714 | 0.334 | 4.754162 | 0.819572 | 3.587911 |
| 0.785714 | 0.445 | 4.766978 | 0.819224 | 3.582245 |
| 0.785714 | 0.556 | 4.78398 | 0.818738 | 3.579238 |
| 0.785714 | 0.667 | 4.805047 | 0.818112 | 3.577507 |
| 0.785714 | 0.778 | 4.829928 | 0.817353 | 3.577582 |
| 0.785714 | 0.889 | 4.858401 | 0.816466 | 3.580641 |
| 0.785714 | 1 | 4.889162 | 0.815542 | 3.585834 |
| 0.857143 | 0.001 | 4.430773 | 0.842972 | 3.349068 |
| 0.857143 | 0.112 | 4.741526 | 0.819831 | 3.603354 |
| 0.857143 | 0.223 | 4.747325 | 0.819695 | 3.594942 |
| 0.857143 | 0.334 | 4.757555 | 0.819401 | 3.587392 |
| 0.857143 | 0.445 | 4.773111 | 0.818907 | 3.582807 |
| 0.857143 | 0.556 | 4.793713 | 0.818219 | 3.580837 |
| 0.857143 | 0.667 | 4.819171 | 0.817334 | 3.580662 |
| 0.857143 | 0.778 | 4.848879 | 0.816282 | 3.583509 |
| 0.857143 | 0.889 | 4.880185 | 0.815264 | 3.588758 |
| 0.857143 | 1 | 4.911997 | 0.814401 | 3.594245 |
| 0.928571 | 0.001 | 4.430887 | 0.842964 | 3.349288 |
| 0.928571 | 0.112 | 4.743134 | 0.819727 | 3.603927 |
| 0.928571 | 0.223 | 4.748875 | 0.81962 | 3.594064 |
| 0.928571 | 0.334 | 4.761419 | 0.819198 | 3.587164 |
| 0.928571 | 0.445 | 4.780152 | 0.818523 | 3.583951 |
| 0.928571 | 0.556 | 4.80495 | 0.817582 | 3.583232 |
| 0.928571 | 0.667 | 4.835104 | 0.816405 | 3.585366 |
| 0.928571 | 0.778 | 4.867294 | 0.81524 | 3.590224 |
| 0.928571 | 0.889 | 4.898823 | 0.814352 | 3.595134 |
| 0.928571 | 1 | 4.926976 | 0.81393 | 3.597565 |
| 1 | 0.001 | 4.431001 | 0.842959 | 3.349504 |
| 1 | 0.112 | 4.744592 | 0.819635 | 3.604636 |
| 1 | 0.223 | 4.750717 | 0.819523 | 3.593469 |
| 1 | 0.334 | 4.765728 | 0.818963 | 3.587407 |
| 1 | 0.445 | 4.78813 | 0.818069 | 3.585585 |
| 1 | 0.556 | 4.817687 | 0.816824 | 3.58661 |
| 1 | 0.667 | 4.850635 | 0.815498 | 3.590453 |
| 1 | 0.778 | 4.882345 | 0.814516 | 3.595152 |
| 1 | 0.889 | 4.910658 | 0.813998 | 3.597162 |
| 1 | 1 | 4.939495 | 0.813623 | 3.599685 |

RMSE was used to select the optimal model using the smallest value.

The final values used for the model were alpha = 0.3571429 and lambda = 0.001.

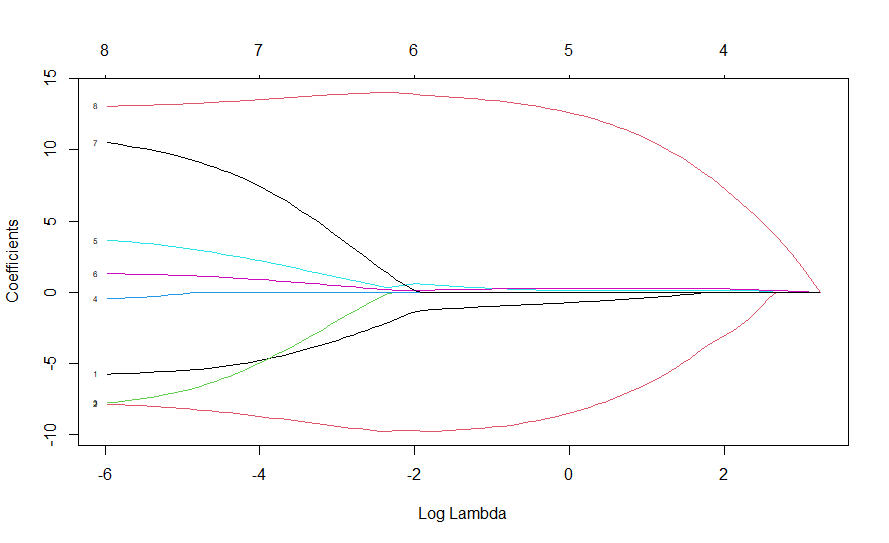


Fig. 10 Plot of Log Lambda v/s Coefficients

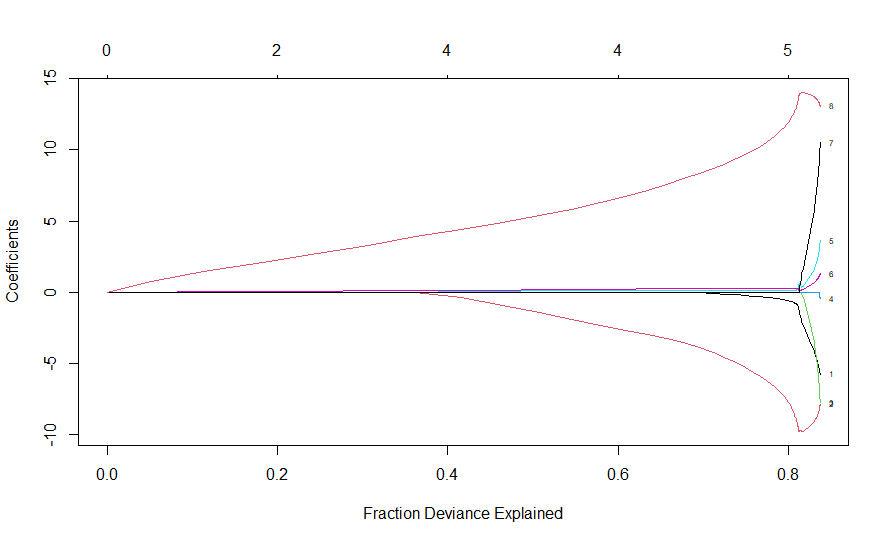


Fig. 11 Plot of Fraction Deviance Explained v/s Coefficients

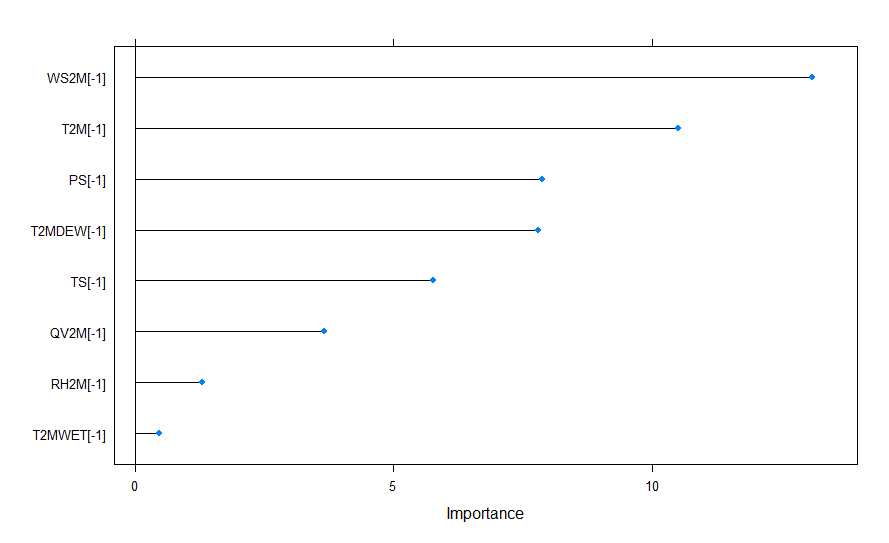


Fig. 12 Plot of Variable Importance

From the above graph, Wind speed at 2 meters is the most important variable for Precipitation followed by Temperature at 2 meters, surface pressure, Dew/Frost Point at 2 Meters, Surface Temperature, Specific Humidity Temperature at 2 meters, Relative Humidity at 2 meters and Wet Bulb Temperature at 2 Meters.

**Comparing the performance of Multiple Linear, Ridge, Lasso and Elastic net regression models:-**

Table 3. Performance comparison table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Alpha | Lambda | RMSE | Rsquared | MAE |
| Multiple Linear Regression | (-) | (-) | 4.446103 | 0.83285 | 3.38456 |
| Ridge Regression | 0 | 0.8572857 | 4.765643 | 0.81652 | 3.601874 |
| Lasso Regression | 1 | 0.001 | 4.43825 | 0.838686 | 3.342969 |
| Elastic Net Regression | 0.3571429 | 0.001 | 4.755989 | 0.81929 | 3.605471 |

From the above table we can observe that Lasso regression is better model for estimating Precipitation as it has low Root mean square value and Mean absolute error.

**Time Series Analysis**

**Surface Temperature**

Time profile of Surface Temperature variable is given below.

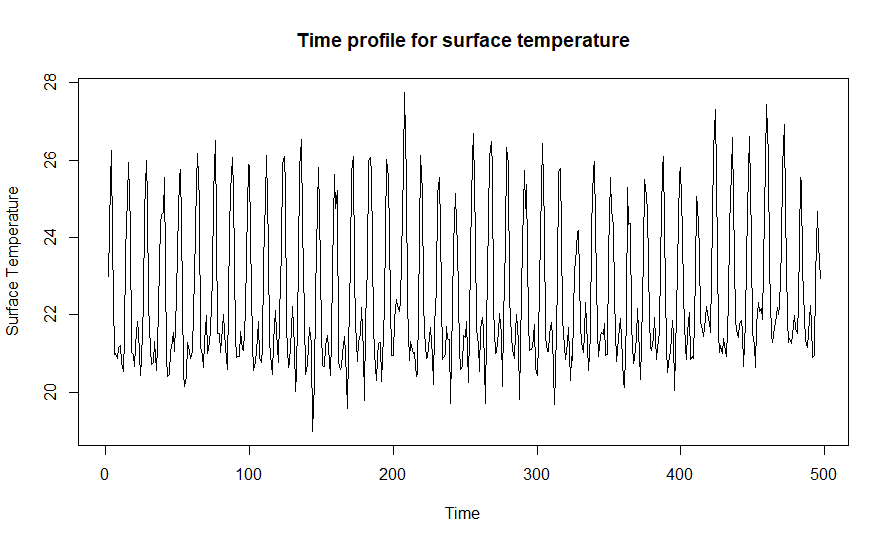


Fig.13 Time profile for surface temperature

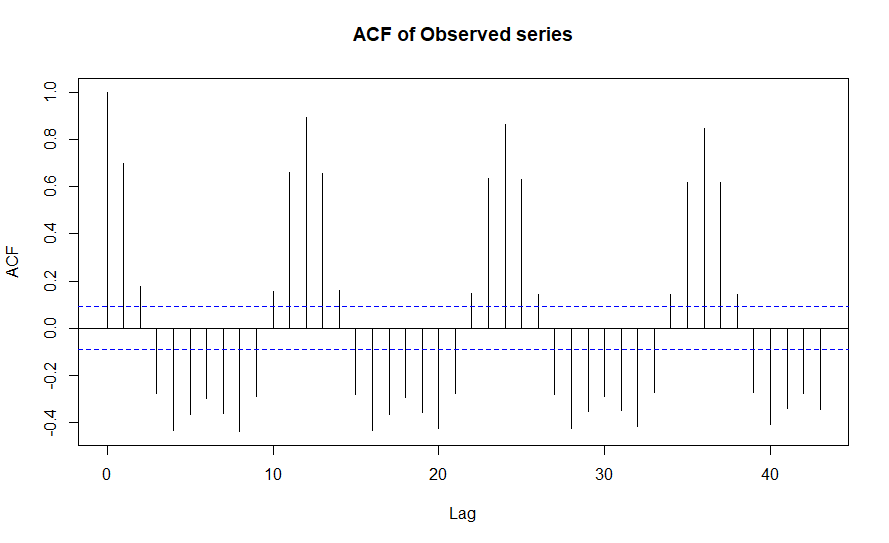


Fig.14 ACF plot of Observed series

From Fig.14 we observe that most of the autocorrelation lies outside the 2σ limits. This implies that the series is non-stationary. Non-stationary is removed by differencing method.

**Rank-sum test for seasonality:**

H0: There is no seasonal variation in the data.

H1: There is seasonal variation in the data.

Test statistic is =~

Chi-square calculated value is 6.01775.

Chi-square critical value is 19.67514.

Since calculated value of chi square is less than critical value of chi square we accept H0 and conclude that there is no seasonal variation in the data.

**Mann Kendall test – test for trend**

H0: There is no monotonic trend in the series.

H1: There is a trend in the series.

2-sided p-value = 0.025007.

As the computed p-value is less than the significance level alpha=0.05, we reject the null hypothesis and conclude that there is trend in the given series.

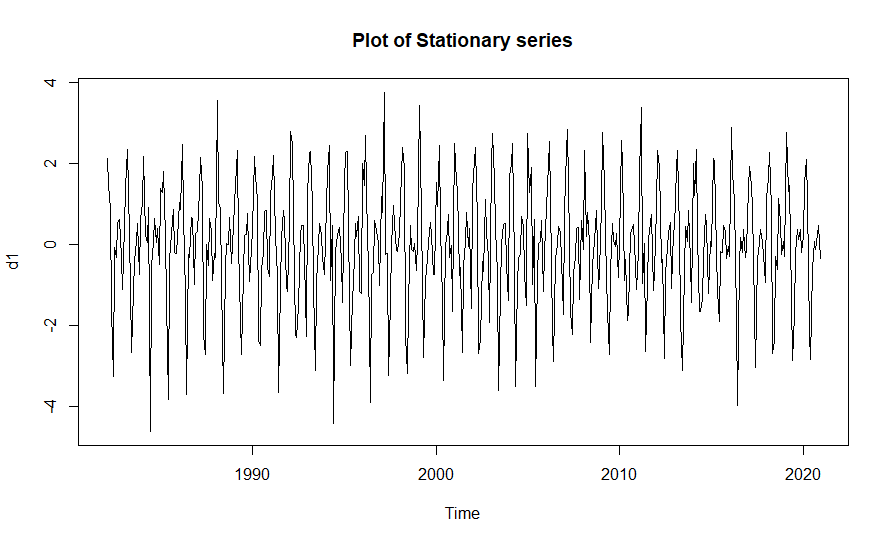


Fig.15 Plot of Stationary Series

Fig.15 suggested that the data is free from trend and seasonality. The trend is removed using differencing method.

**Augmented Dickey-Fuller test:**

H0: The series is not stationary

H1: The series is stationary

Dickey-Fuller= -21.203, Lag order = 7, p-value = 0.01.

As p value is less than 0.05, we reject H0 and conclude that the series is stationary.

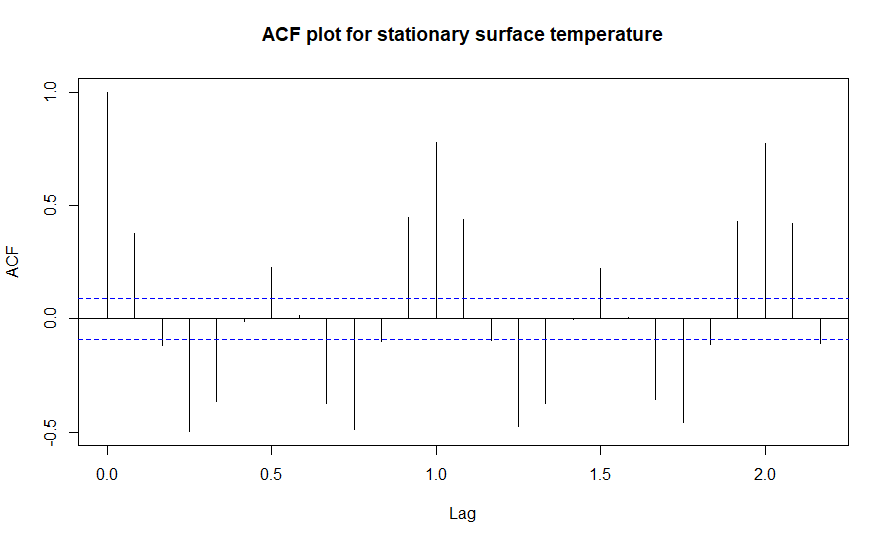


Fig.16 ACF of Stationary series

From the Fig.16, the value of non-seasonal MA order q = 1 is obtained.

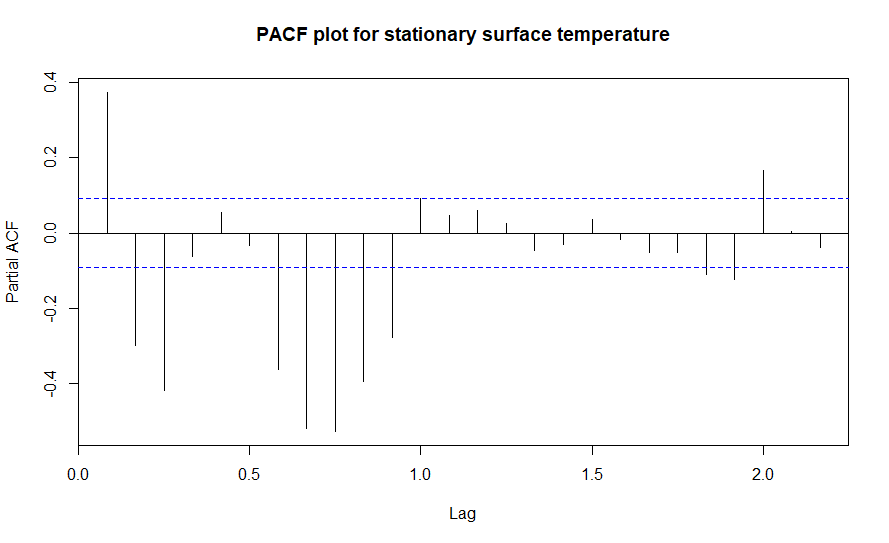


Fig.17 PACF of Stationary Series

From Fig.17 we identified that the value of non-seasonal AR order p = 1 is obtained

Since the first trend difference series turns out to be stationary we put d=1.This gives only an initial idea for order of the model to be fitted. We choose the model with minimum AIC; however the accuracy of fitted model depends not only on AIC value but also on the assumptions of residuals. Usually the residuals are assumed to be uncorrelated. In the classical time series set up it is common to assume that the white noise sequence **Ɛt** is iid Gaussian (Box and Jenkins, 1976). To check the validity of these assumptions we use Ljung Box test and Box-pierce test.

**Ljung–Box test**

The null and alternative hypothesis are

H0: Series is uncorrelated

H1: Series is not uncorrelated

**Box-pierce test**

The null and alternative hypothesis are

H0: Series is uncorrelated

H1: Series is not uncorrelated

Based on the ACF and PACF plot different ARIMA model with different order were chosen.

Table 4. Represents the summary of the fitted models

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | p | d | q | LB | BP | AIC |
| 1 | 3 | 1 | 4 | 0.642287 | 0.469577 | 1365.873 |
| 2 | 3 | 1 | 3 | 0.297968 | 0.161021 | 1319.827 |
| 3 | 3 | 1 | 2 | 0.727982 | 0.563259 | 1380.891 |
| 4 | 3 | 1 | 1 | 0.11089 | 0.040167 | 1484.203 |
| 5 | 2 | 1 | 4 | 0.687006 | 0.519145 | 1341.454 |
| 6 | 2 | 1 | 3 | 0.323595 | 0.18057 | 1368.125 |
| **7** | **2** | **1** | **2** | **0.772295** | **0.624649** | **1325.823** |
| 8 | 2 | 1 | 1 | 0.177903 | 0.075075 | 1394.579 |
| 9 | 1 | 1 | 4 | 0.694492 | 0.533999 | 1424.304 |
| 10 | 1 | 1 | 3 | 0.373043 | 0.225519 | 1437.458 |
| 11 | 1 | 1 | 2 | 0.219572 | 0.102012 | 1580.458 |
| 12 | 1 | 1 | 1 | 0.565272 | 0.424919 | 1601.832 |

Here the model ARIMA (2, 1, 2) has minimum AIC value and maximum p-value of Box- pierce method. Thus ARIMA (2, 1, 2) is the best fitted model for monthly averaged Surface Temperature.

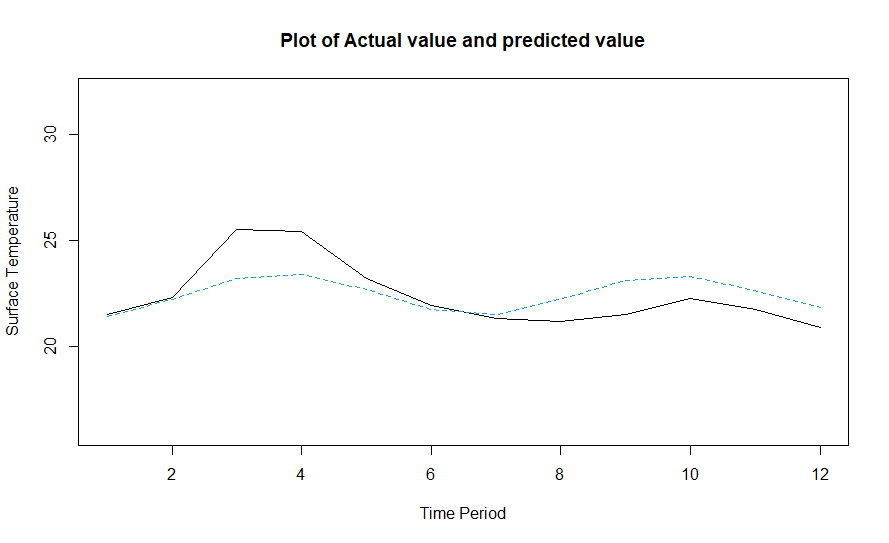


Fig.18 Plot of Actual value and Predicted value for Surface Temperature

Table 5. Test values and ARIMA forecasted values:

|  |  |  |  |
| --- | --- | --- | --- |
| Month | Year | Actual Value | Predicted value |
| January | 2021 | 21.51419 | 21.42105 |
| February | 2021 | 22.30826 | 22.21294 |
| March | 2021 | 25.54926 | 23.20877 |
| April | 2021 | 25.43503 | 23.42752 |
| May | 2021 | 23.19407 | 22.6866 |
| June | 2021 | 21.96409 | 21.74829 |
| July | 2021 | 21.34209 | 21.53721 |
| August | 2021 | 21.171 | 22.23053 |
| September | 2021 | 21.54205 | 23.11472 |
| October | 2021 | 22.25534 | 23.3183 |
| November | 2021 | 21.75775 | 22.66961 |
| December | 2021 | 20.90296 | 21.83649 |

**NEURAL NETWORKING**

**Forecasting from Multi-Layer Perceptions- Neural Networking**

MLP fit with 5 hidden nodes and 20 repetitions.

Series modelled in differences: D1.

Univariate lags: (1,2,3,4,5,6,7,8,9,10,11,12)

Forecast combined using the median operator.

MSE: 0.1137.

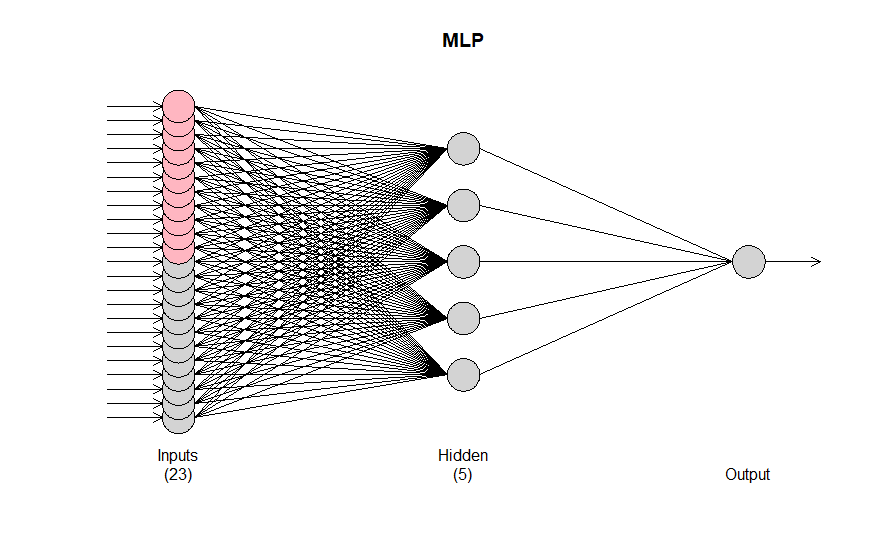


Fig.19 Architecture of Multi-Layer Perception neural network model (MLP-NN).

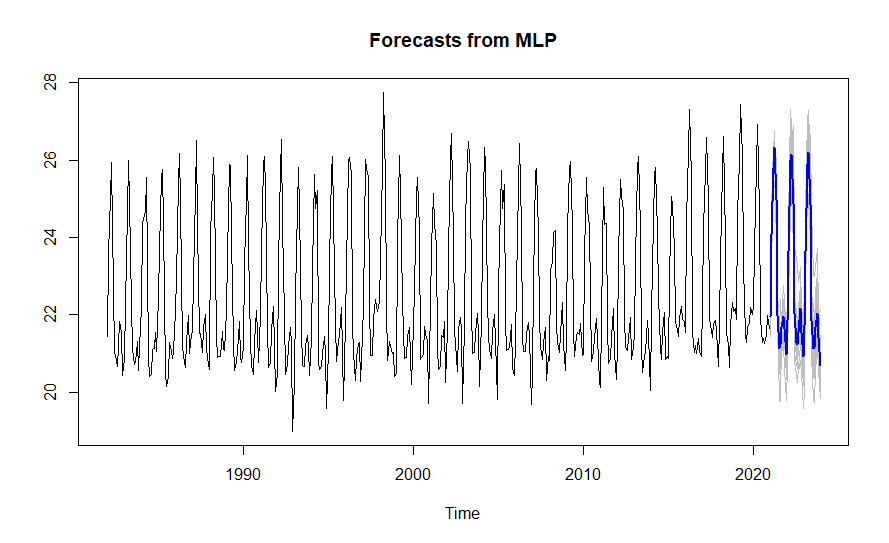


Fig. 20 Forecast from MLP-NN

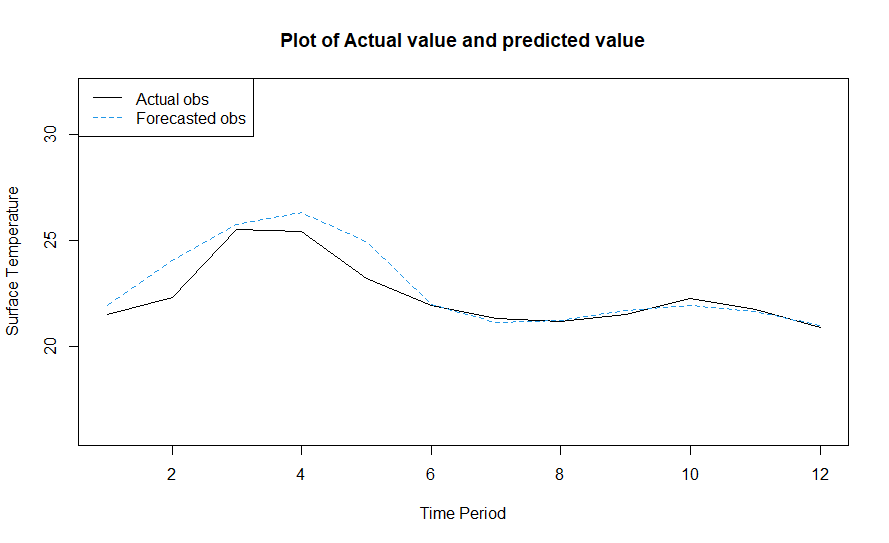


Fig. 21 Plot of Actual value and Predicted value

Table 6. Test values and MLP-NN forecasted values:

|  |  |  |  |
| --- | --- | --- | --- |
| Month | Year | Actual Value | Predicted value |
| January | 2021 | 21.51419 | 21.96645 |
| February | 2021 | 22.30826 | 24.08574 |
| March | 2021 | 25.54926 | 25.77982 |
| April | 2021 | 25.43503 | 26.31691 |
| May | 2021 | 23.19407 | 24.97835 |
| June | 2021 | 21.96409 | 21.98333 |
| July | 2021 | 21.34209 | 21.1456 |
| August | 2021 | 21.171 | 21.22844 |
| September | 2021 | 21.54205 | 21.68719 |
| October | 2021 | 22.25534 | 21.96634 |
| November | 2021 | 21.75775 | 21.64671 |
| December | 2021 | 20.90296 | 20.98591 |

**Forecasting from Extreme learning machine-Neural Networking**

ELM fit with 100 hidden nodes and 20 repetitions.

Series modelled in differences: D1.

Univariate lags: (1,2,3,4,5,6,7,8,9,10,11,12)

Forecast combined using the median operator. Output weight estimation using: lasso.

MSE: 0.3062.

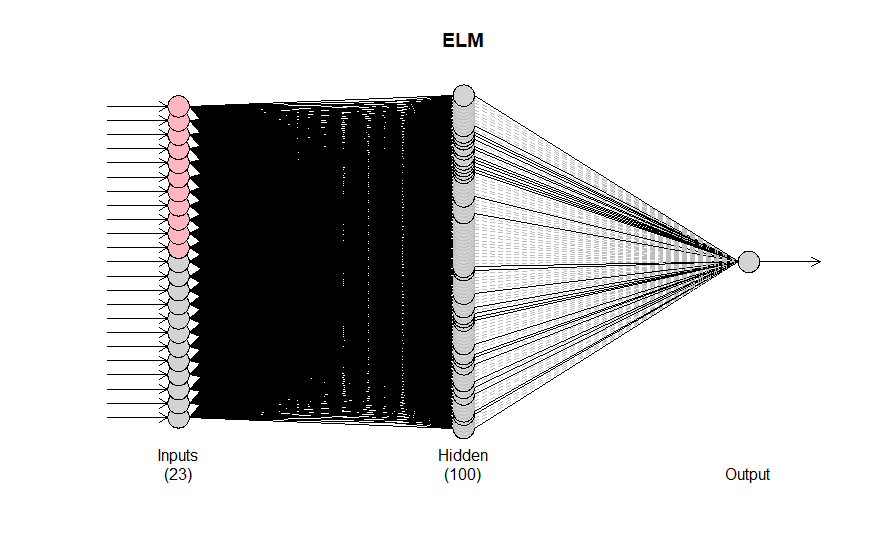


Fig. 22 Architecture of Extreme Learning Machine neural network model (ELM-NN).

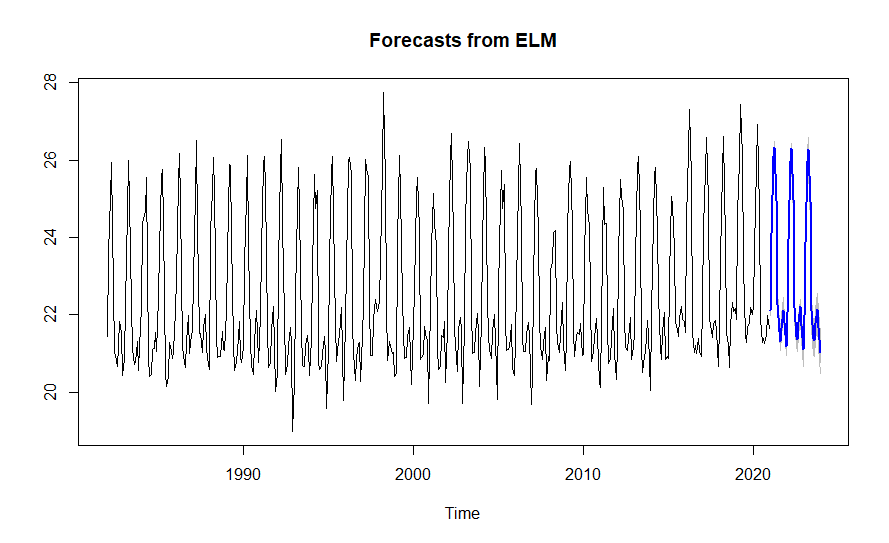


Fig. 23 Forecast from ELM-NN

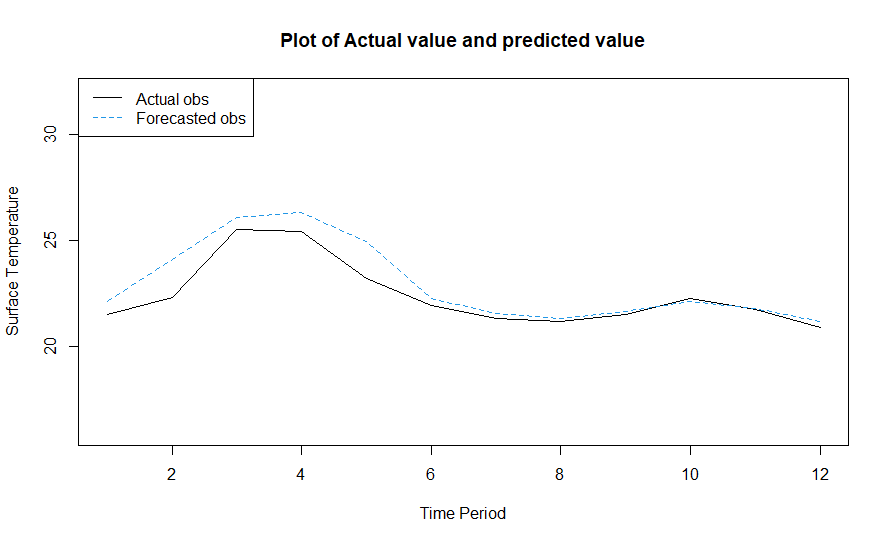


Fig. 24 Plot of Actual value and Predicted value

Table 7. Test values and ELM-NN forecasted values:

|  |  |  |  |
| --- | --- | --- | --- |
| Month | Year | Actual Value | Predicted value |
| January | 2021 | 21.51419 | 22.12061 |
| February | 2021 | 22.30826 | 24.10431 |
| March | 2021 | 25.54926 | 26.10878 |
| April | 2021 | 25.43503 | 26.33618 |
| May | 2021 | 23.19407 | 24.96007 |
| June | 2021 | 21.96409 | 22.27836 |
| July | 2021 | 21.34209 | 21.56454 |
| August | 2021 | 21.171 | 21.31888 |
| September | 2021 | 21.54205 | 21.65778 |
| October | 2021 | 22.25534 | 22.11419 |
| November | 2021 | 21.75775 | 21.81213 |
| December | 2021 | 20.90296 | 21.19474 |

Table 8. Represents the Actual value and forecasted value of all models.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Month | Year | Actual Value | ARIMA | MLP-NN | ELM-NN |
| January | 2021 | 21.51419 | 21.42105 | 21.96645 | 22.12061 |
| February | 2021 | 22.30826 | 22.21294 | 24.08574 | 24.10431 |
| March | 2021 | 25.54926 | 23.20877 | 25.77982 | 26.10878 |
| April | 2021 | 25.43503 | 23.42752 | 26.31691 | 26.33618 |
| May | 2021 | 23.19407 | 22.6866 | 24.97835 | 24.96007 |
| June | 2021 | 21.96409 | 21.74829 | 21.98333 | 22.27836 |
| July | 2021 | 21.34209 | 21.53721 | 21.1456 | 21.56454 |
| August | 2021 | 21.171 | 22.23053 | 21.22844 | 21.31888 |
| September | 2021 | 21.54205 | 23.11472 | 21.68719 | 21.65778 |
| October | 2021 | 22.25534 | 23.3183 | 21.96634 | 22.11419 |
| November | 2021 | 21.75775 | 22.66961 | 21.64671 | 21.81213 |
| December | 2021 | 20.90296 | 21.83649 | 20.98591 | 21.19474 |

Fig. 25 Actual value and forecasted value of all fitted models.

Table 9: Represents the Accuracy measures RMSE, MAE and MAPE for all the models.

|  |  |  |  |
| --- | --- | --- | --- |
| Model | RMSE | MAE | MAPE |
| ARIMA | 1.165356 | 0.916284 | 3.978052 |
| MLP-NN | 0.792915 | 0.502313 | 2.190954 |
| ELM-NN | 0.823052 | 0.576398 | 2.517131 |

In terms of RMSE, MAE and MAPE MLP-NN model is better forecasting model compare to all other fitted models. Because MLP-NN has minimum error statistics compared to all other model.

Table 10: Represents the forecasted Surface Temperature for next 19 months using MLP-NN model.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Month | Year | Forecasted value | Month | Year | Forecasted value |
| June | 2022 | 22.03297 | April | 2023 | 26.18757 |
| July | 2022 | 21.28415 | May | 2023 | 24.84985 |
| August | 2022 | 21.22931 | June | 2023 | 21.85286 |
| September | 2022 | 21.55525 | July | 2023 | 21.12826 |
| October | 2022 | 22.15646 | August | 2023 | 21.15424 |
| November | 2022 | 21.55119 | September | 2023 | 21.66972 |
| December | 2022 | 20.91909 | October | 2023 | 22.03522 |
| January | 2023 | 21.71873 | November | 2023 | 21.41066 |
| February | 2023 | 23.63629 | December | 2023 | 20.68305 |
| March | 2023 | 25.89924 |  |  |  |

**Specific Humidity**

Time profile of Surface Humidity variable is given below.

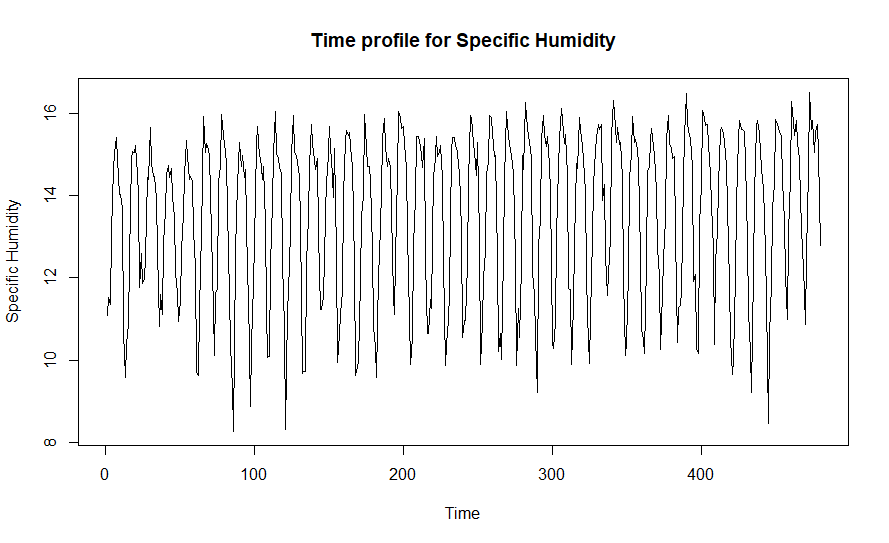


Fig. 26 Time profile for Specific Humidity

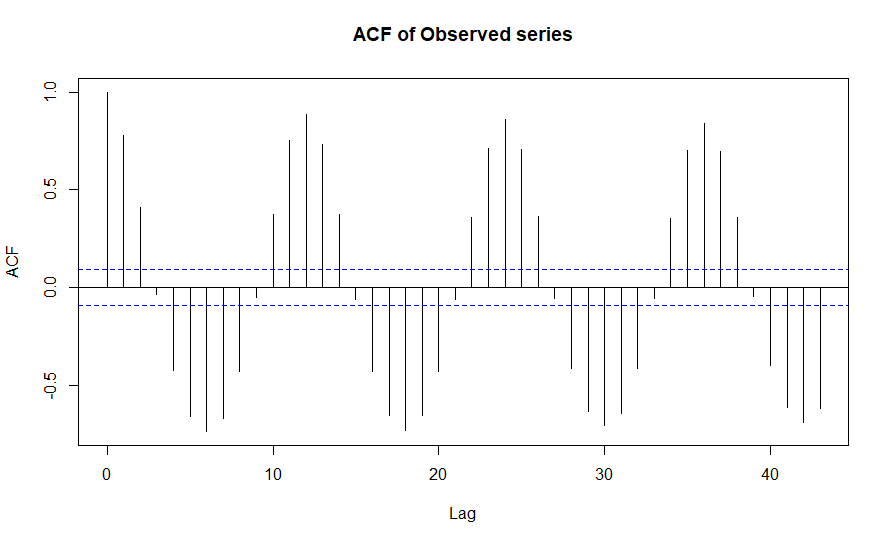


Fig. 27 ACF plot for Original Specific Humidity Observations

From Fig.27 we observe that most of the autocorrelation lies outside the 2σ limits. This implies that the series is non-stationary. Non-stationary is removed by differencing method.

**Rank-sum test for seasonality:**

H0: There is no seasonal variation in the data.

H1: There is seasonal variation in the data.

Test statistic is =~ χ2(D-1)

Chi-square calculated value is 555.3728.

Chi-square critical value is 19.67514.

Since calculated value of chi square is more than critical value of chi square we reject H0 and conclude that there is seasonal variation in the data.

**Man Kendall test – test for trend**

H0: There is no monotonic trend in the series.

H1: There is a trend in the series.

2-sided p-value = 0.76825.

As the computed p-value is greater than the significance level alpha=0.05, we accept the null hypothesis and conclude that there is no trend in seasonal differenced series.

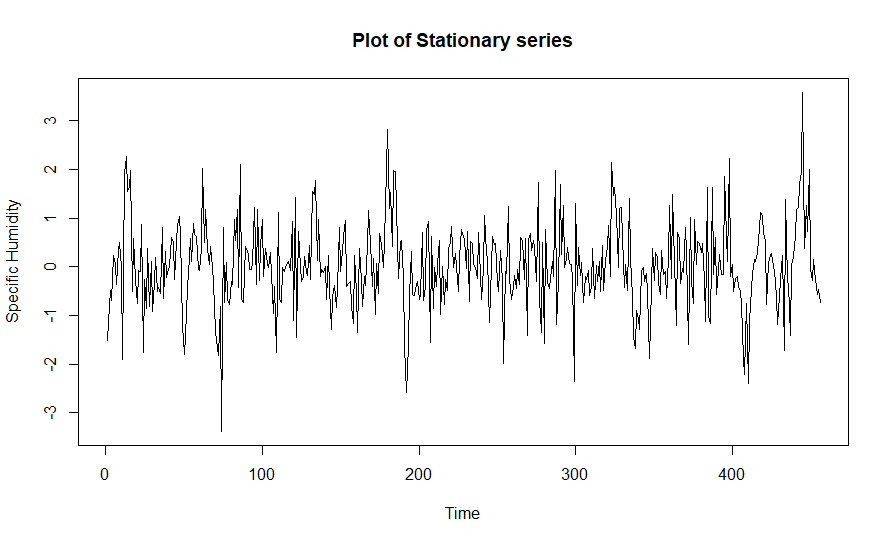


Fig. 28 Plot of Stationary series of Specific Humidity

Fig.28 suggested that the data is free from trend and seasonality. The seasonality is removed using seasonal differencing method.

**Augmented Dickey-Fuller test:**

H0: The series is not stationary

H1: The series is stationary

Dickey-Fuller= -7.2918, Lag order = 7, p-value = 0.01.

As p value is less than 0.05, we reject H0 and conclude that the series is stationary.

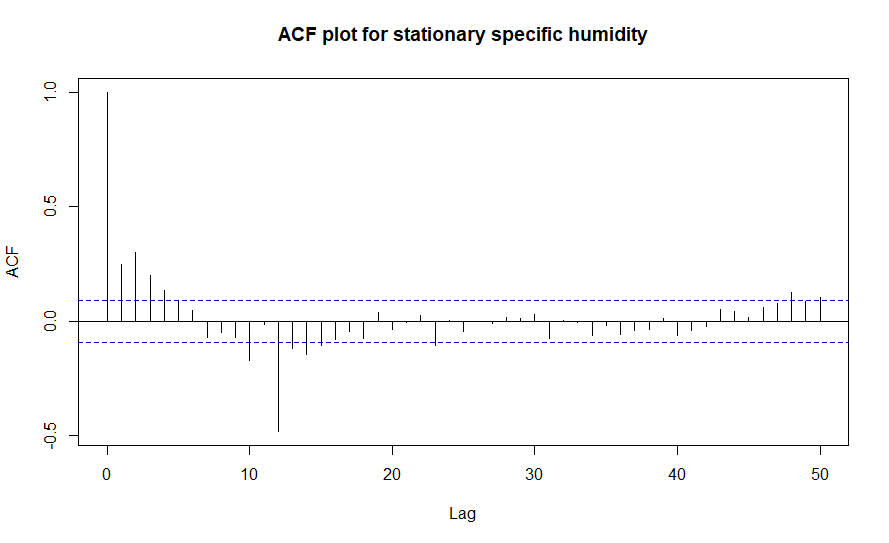


Fig. 29 ACF plot for Stationary specific humidity

From the Fig.29, the value of seasonal MA order Q = 1 and non-seasonal MA order q = 4 are obtained.

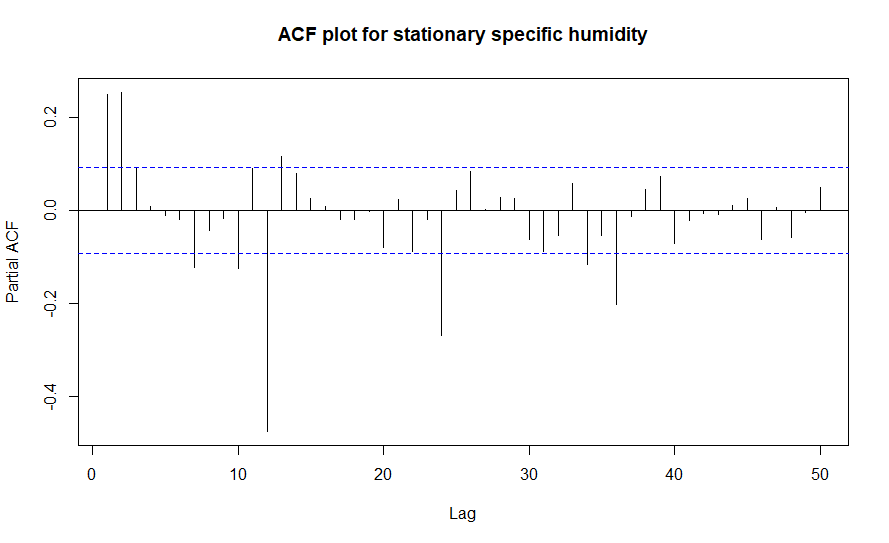


Fig.30 PACF plot for Stationary specific humidity

From Fig.30 we identified that the value of seasonal AR order P = 3 and non-seasonal AR order p = 3 are obtained.

And since the first seasonal difference series turns out to be stationary we put D=1 and here d=0.This gives only an initial idea for order of the model to be fitted. We choose the model with minimum AIC; however the accuracy of fitted model depends not only on AIC value but also on the assumptions of residuals. Usually the residuals are assumed to be uncorrelated. In the classical time series set up it is common to assume that the white noise sequence **Ɛt** is iid Gaussian (Box and Jenkins, 1976). To check the validity of these assumptions we use Ljung Box test and Box-pierce test.

Based on the ACF and PACF plot different SARIMA model with different order were chosen.

Table. 11 Represents the summary of the fitted models

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **p** | **d** | **q** | **P** | **D** | **Q** | **LB** | **BP** | **IC** |
| **1** | **3** | **0** | **4** | **3** | **1** | **1** | **0.562437** | **0.643198** | **885.43** |
| 2 | 3 | 0 | 4 | 2 | 1 | 1 | 0.275322 | 0.351156 | 887.8434 |
| 3 | 3 | 0 | 4 | 1 | 1 | 1 | 0.204048 | 0.270134 | 889.2681 |
| 4 | 3 | 0 | 3 | 3 | 1 | 1 | 0.507668 | 0.589195 | 886.0636 |
| 5 | 3 | 0 | 3 | 2 | 1 | 1 | 0.504908 | 0.586753 | 884.1273 |
| 6 | 3 | 0 | 3 | 1 | 1 | 1 | 0.142142 | 0.196182 | 888.0832 |
| 7 | 3 | 0 | 2 | 3 | 1 | 1 | 0.392 | 0.472216 | 888.945 |
| 8 | 3 | 0 | 2 | 2 | 1 | 1 | 0.394168 | 0.474508 | 886.9861 |
| 9 | 3 | 0 | 2 | 1 | 1 | 1 | 0.119376 | 0.16983 | 885.0115 |
| 10 | 3 | 0 | 1 | 3 | 1 | 1 | 0.377488 | 0.457292 | 887.0187 |
| 11 | 3 | 0 | 1 | 2 | 1 | 1 | 0.377981 | 0.457908 | 885.0754 |
| 12 | 3 | 0 | 1 | 1 | 1 | 1 | 0.263046 | 0.337136 | 884.4901 |
| 13 | 2 | 0 | 4 | 3 | 1 | 1 | 0.554485 | 0.63329 | 886.4481 |
| 14 | 2 | 0 | 4 | 2 | 1 | 1 | 0.550503 | 0.629724 | 884.5608 |
| 15 | 2 | 0 | 4 | 1 | 1 | 1 | 0.438071 | 0.522608 | 883.9495 |
| 16 | 2 | 0 | 3 | 3 | 1 | 1 | 0.55512 | 0.634061 | 884.504 |
| 17 | 2 | 0 | 3 | 2 | 1 | 1 | 0.549258 | 0.628788 | 882.6395 |
| 18 | 2 | 0 | 3 | 1 | 1 | 1 | 0.43377 | 0.518657 | 882.0707 |
| 19 | 2 | 0 | 2 | 3 | 1 | 1 | 0.283571 | 0.359503 | 887.2032 |
| 20 | 2 | 0 | 2 | 2 | 1 | 1 | 0.280521 | 0.356368 | 885.2602 |
| 21 | 2 | 0 | 2 | 1 | 1 | 1 | 0.192683 | 0.25902 | 884.3988 |
| 22 | 2 | 0 | 1 | 3 | 1 | 1 | 0.231796 | 0.301055 | 886.7419 |
| 23 | 2 | 0 | 1 | 2 | 1 | 1 | 0.237576 | 0.307209 | 884.6617 |
| 24 | 2 | 0 | 1 | 1 | 1 | 1 | 0.127782 | 0.179837 | 884.3523 |
| 25 | 1 | 0 | 4 | 3 | 1 | 1 | 0.329379 | 0.407038 | 889.1601 |
| 26 | 1 | 0 | 4 | 2 | 1 | 1 | 0.329327 | 0.407109 | 887.2104 |
| 27 | 1 | 0 | 4 | 1 | 1 | 1 | 0.220692 | 0.289688 | 886.5945 |
| 28 | 1 | 0 | 3 | 3 | 1 | 1 | 0.318215 | 0.395182 | 887.1765 |
| 29 | 1 | 0 | 3 | 2 | 1 | 1 | 0.317672 | 0.394731 | 885.228 |
| 30 | 1 | 0 | 3 | 1 | 1 | 1 | 0.212965 | 0.280866 | 884.6074 |
| 31 | 1 | 0 | 2 | 3 | 1 | 1 | 0.270613 | 0.344024 | 886.2591 |
| 32 | 1 | 0 | 2 | 2 | 1 | 1 | 0.262659 | 0.335538 | 884.3936 |
| 33 | 1 | 0 | 2 | 1 | 1 | 1 | 0.15401 | 0.212198 | 883.9953 |
| 34 | 1 | 0 | 1 | 3 | 1 | 1 | 0.180059 | 0.239436 | 885.4035 |
| 35 | 1 | 0 | 1 | 2 | 1 | 1 | 0.172539 | 0.230872 | 883.5603 |
| 36 | 1 | 0 | 1 | 1 | 1 | 1 | 0.074408 | 0.110527 | 883.4477 |

Here the model SARIMA (3, 0, 4) (3, 1, 1)12 has maximum p-value of Box- pierce method and L-jung Box method. Thus SARIMA (3, 0, 4) (3, 1, 1)12 is the best fitted model for monthly averaged Specific humidity.

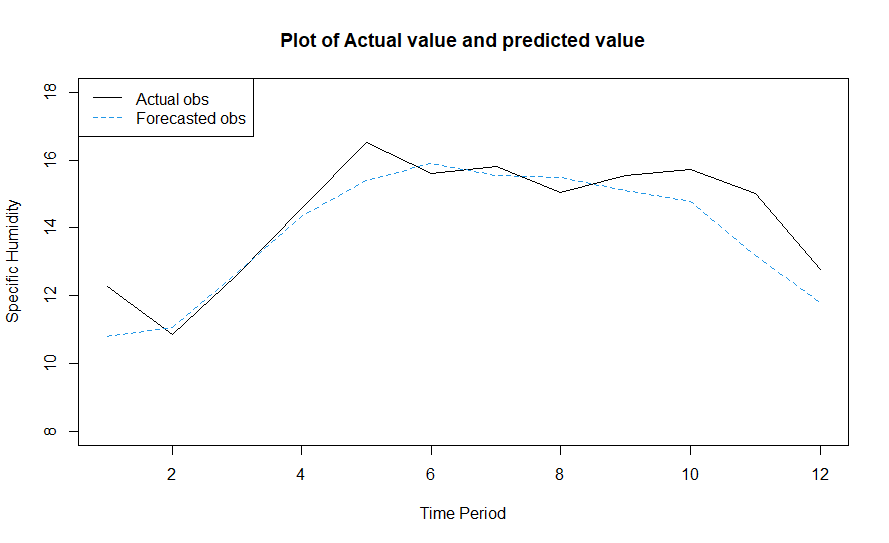


Fig. 31 Plot of Actual value and Predicted value for Specific Humidity.

Table 12. Test values and ARIMA forecasted values:

|  |  |  |  |
| --- | --- | --- | --- |
| Month | Year | Actual Value | Predicted Value |
| January | 2021 | 12.26368 | 10.80231 |
| February | 2021 | 10.85437 | 11.08015 |
| March | 2021 | 12.59529 | 12.66648 |
| April | 2021 | 14.5787 | 14.32641 |
| May | 2021 | 16.50727 | 15.39604 |
| June | 2021 | 15.61018 | 15.9143 |
| July | 2021 | 15.81228 | 15.53916 |
| August | 2021 | 15.04529 | 15.47935 |
| September | 2021 | 15.54146 | 15.10035 |
| October | 2021 | 15.73626 | 14.76737 |
| November | 2021 | 15.01087 | 13.1947 |
| December | 2021 | 12.77854 | 11.80644 |

**NEURAL NETWORKING**

**Forecasting from Multi-Layer Perceptions- Neural Networking**

MLP fit with 5 hidden nodes and 20 repetitions.

Series modelled in differences: D1.

Univariate lags: (1,2,3,4,5,6,7,8,9,10)

Forecast combined using the median operator.

MSE: 0.1446.

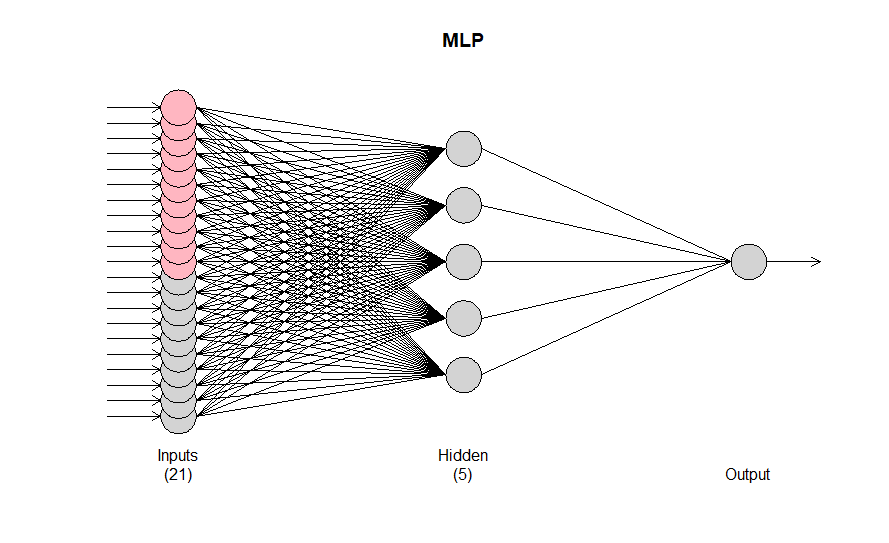


Fig. 32 Architecture of Multi-Layer Perception neural network model (MLP-NN).

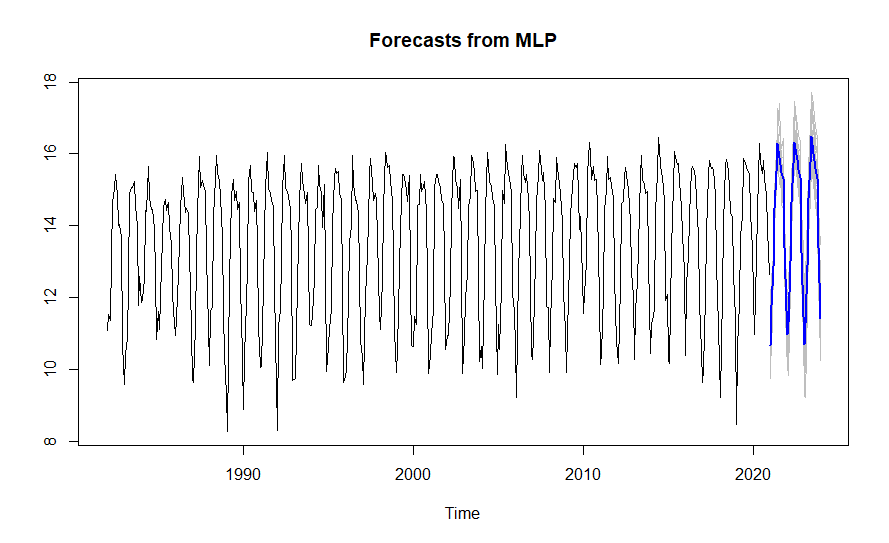


Fig. 33 Forecast of Specific Humidity using MLP-NN.

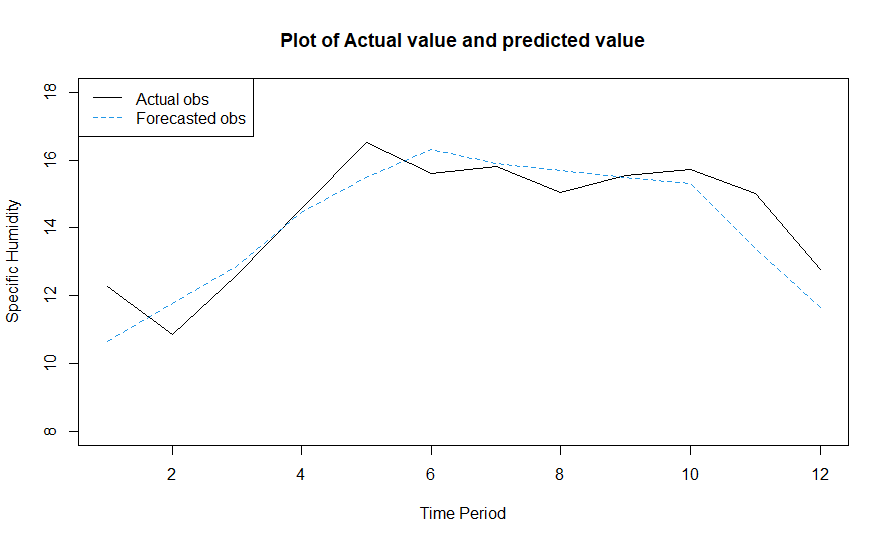


Fig. 34 Plot of Actual and Predicted value using MLP-NN.

Table 13. Test values and MLP-NN forecasted values:

|  |  |  |  |
| --- | --- | --- | --- |
| Month | Year | Actual Value | Predicted Value |
| January | 2021 | 12.26368 | 10.65462 |
| February | 2021 | 10.85437 | 11.77079 |
| March | 2021 | 12.59529 | 12.86055 |
| April | 2021 | 14.5787 | 14.4681 |
| May | 2021 | 16.50727 | 15.50117 |
| June | 2021 | 15.61018 | 16.30055 |
| July | 2021 | 15.81228 | 15.89175 |
| August | 2021 | 15.04529 | 15.69645 |
| September | 2021 | 15.54146 | 15.47502 |
| October | 2021 | 15.73626 | 15.29884 |
| November | 2021 | 15.01087 | 13.39083 |
| December | 2021 | 12.77854 | 11.64593 |

**Forecasting from Extreme learning machine-Neural Networking**

ELM fit with 100 hidden nodes and 20 repetitions.

Series modelled in differences: D1.

Univariate lags: (1,2,3,4,5,6,7,8,9,10)

Forecast combined using the median operator.

Output weight estimation using: lasso.

MSE: 0.3981.

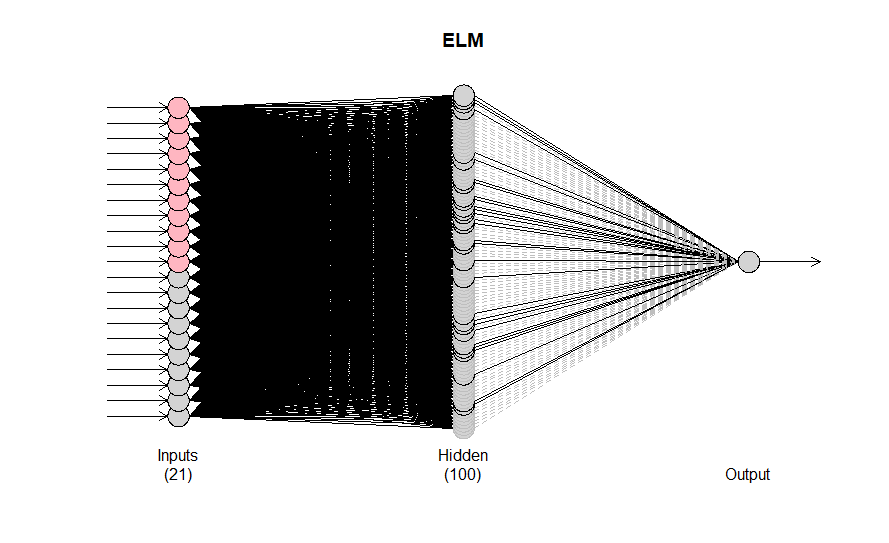


Fig. 35 Architecture of Extreme Learning Machine neural network model (ELM-NN).

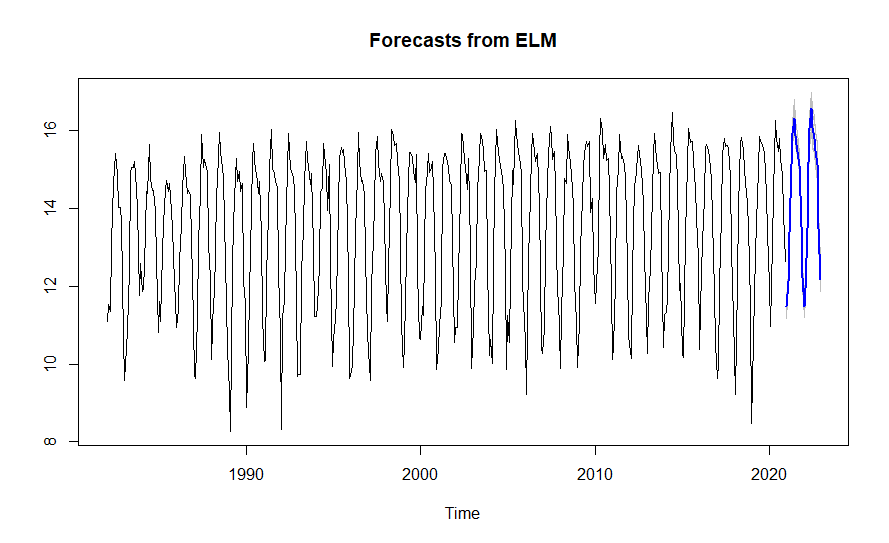


Fig. 36 Forecast of Specific Humidity using ELM-NN.

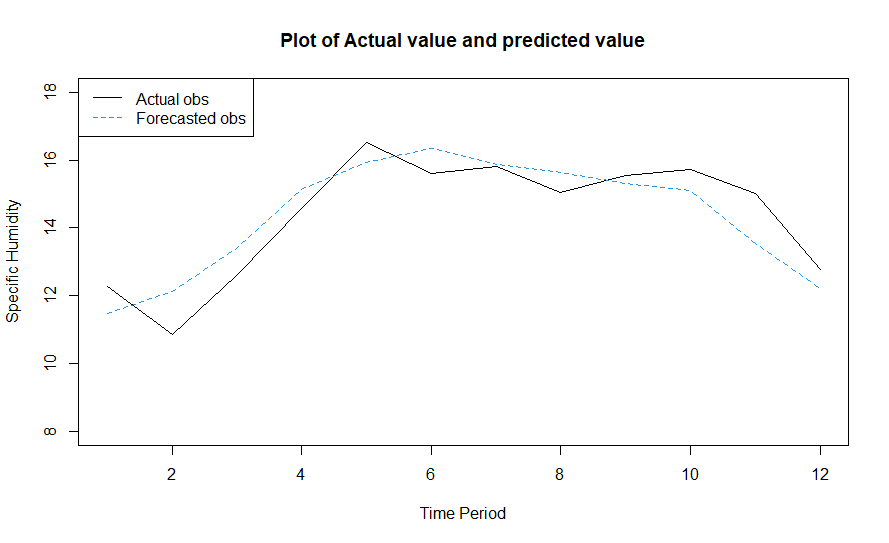


Fig. 37 Plot of Actual and Predicted value using ELM-NN.

Table 14. Test values and ELM-NN forecasted values:

|  |  |  |  |
| --- | --- | --- | --- |
| Month | Year | Actual Value | Predicted Value |
| January | 2021 | 12.26368 | 11.47991 |
| February | 2021 | 10.85437 | 12.11769 |
| March | 2021 | 12.59529 | 13.39147 |
| April | 2021 | 14.5787 | 15.14426 |
| May | 2021 | 16.50727 | 15.93888 |
| June | 2021 | 15.61018 | 16.3297 |
| July | 2021 | 15.81228 | 15.88313 |
| August | 2021 | 15.04529 | 15.63914 |
| September | 2021 | 15.54146 | 15.30064 |
| October | 2021 | 15.73626 | 15.09897 |
| November | 2021 | 15.01087 | 13.55195 |
| December | 2021 | 12.77854 | 12.22116 |

Table 15. Represents the Actual value and forecasted value of all models.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Month | Year | Actual Value | SARIMA | MLP | ELM |
| January | 2021 | 12.26368 | 10.80231 | 10.65462 | 11.47991 |
| February | 2021 | 10.85437 | 11.08015 | 11.77079 | 12.11769 |
| March | 2021 | 12.59529 | 12.66648 | 12.86055 | 13.39147 |
| April | 2021 | 14.5787 | 14.32641 | 14.4681 | 15.14426 |
| May | 2021 | 16.50727 | 15.39604 | 15.50117 | 15.93888 |
| June | 2021 | 15.61018 | 15.9143 | 16.30055 | 16.3297 |
| July | 2021 | 15.81228 | 15.53916 | 15.89175 | 15.88313 |
| August | 2021 | 15.04529 | 15.47935 | 15.69645 | 15.63914 |
| September | 2021 | 15.54146 | 15.10035 | 15.47502 | 15.30064 |
| October | 2021 | 15.73626 | 14.76737 | 15.29884 | 15.09897 |
| November | 2021 | 15.01087 | 13.1947 | 13.39083 | 13.55195 |
| December | 2021 | 12.77854 | 11.80644 | 11.64593 | 12.22116 |

Fig. 38 Comparison between Actual value and Forecasted value of all model

Table 16: Represents the Accuracy measures RMSE, MAE and MAPE for all the models.

|  |  |  |  |
| --- | --- | --- | --- |
| Model | RMSE | MAE | MAPE |
| SARIMA | 0.876659 | 0.694286 | 4.857166 |
| MLP-NN | 0.891335 | 0.715413 | 5.219929 |
| ELM-NN | 0.778125 | 0.687987 | 4.524891 |

In terms of RMSE, MAE and MAPE ELM-NN model is better forecasting model compare to all other fitted models. Because ELM-NN has minimum error statistics compared to all other model.

Table 17: Represents the forecasted Specific Humidity for next 19 months using ELM-NN model.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Month | Year | Forecasted value | Month | Year | Forecasted value |
| June | 2022 | 16.57539 | April | 2023 | 15.14691 |
| July | 2022 | 16.07875 | May | 2023 | 16.04551 |
| August | 2022 | 15.80884 | June | 2023 | 16.63242 |
| September | 2022 | 15.4911 | July | 2023 | 16.13267 |
| October | 2022 | 15.12162 | August | 2023 | 15.86861 |
| November | 2022 | 13.48512 | September | 2023 | 15.58821 |
| December | 2022 | 12.17914 | October | 2023 | 15.17392 |
| January | 2023 | 11.34161 | November | 2023 | 13.53707 |
| February | 2023 | 11.88724 | December | 2023 | 12.17789 |
| March | 2023 | 13.44667 |  |  |  |

**Precipitation**

Time profile of Precipitation variable is given below.

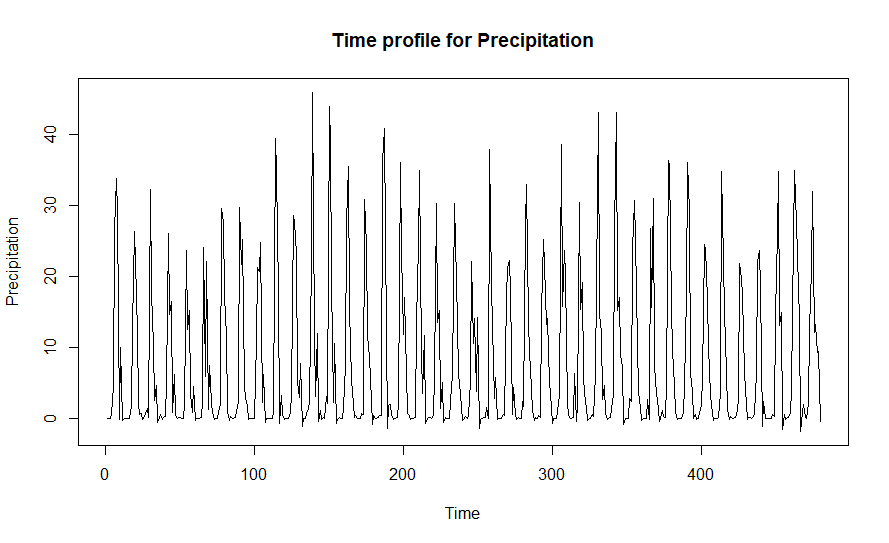


Fig. 39 Time profile for Precipitation Corrected.

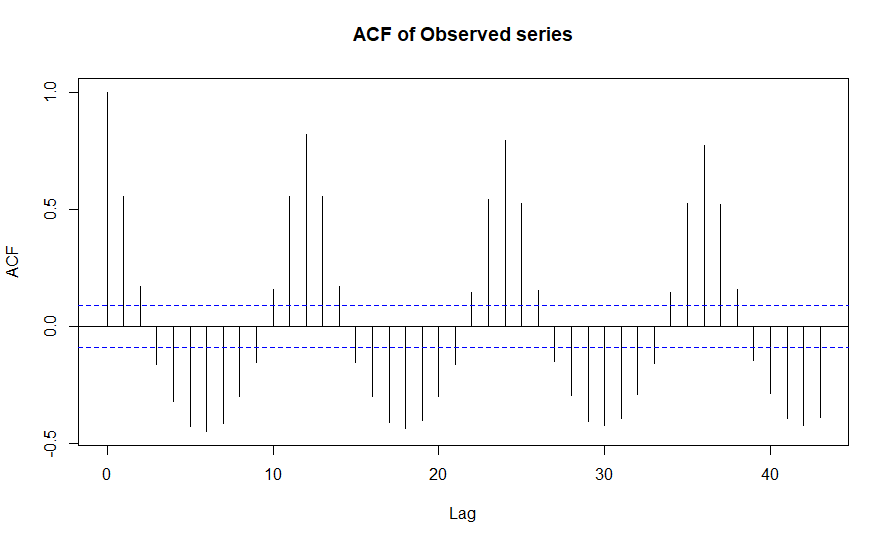


Fig.40 ACF plot of Observed Precipitation series.

From Fig.40 we observe that most of the autocorrelation lies outside the 2σ limits. This implies that the series is non-stationary. Non-stationary is removed by differencing method.

**Rank-sum test for seasonality:**

H0: There is no seasonal variation in the data.

H1: There is seasonal variation in the data.

Test statistic is =~ χ2(D-1)

Chi-square calculated value is 258.9053.

Chi-square critical value is 19.67514.

Since calculated value of chi square is more than critical value of chi square we reject H0 and conclude that there is seasonal variation in the data.

**Man Kendall test – test for trend**

H0: There is no monotonic trend in the series.

H1: There is a trend in the series.

2-sided p-value = 0.18372.

As the computed p-value is greater than the significance level alpha=0.05, we accept the null hypothesis and conclude that there is no trend in the seasonal differenced series.

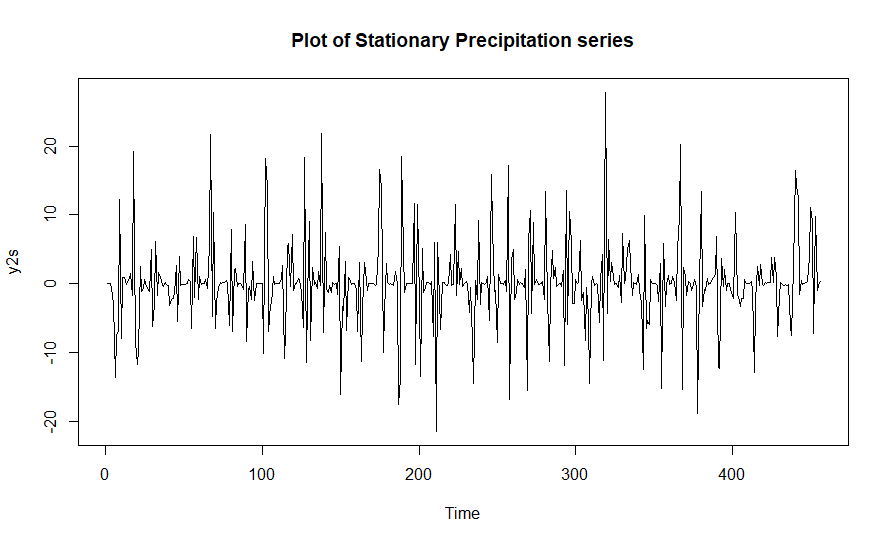


Fig. 41 Plot of Stationary Precipitation series

Fig.41 suggested that the data is free from trend and seasonality. The seasonality41s removed using seasonal differencing method.

**Augmented Dickey-Fuller test:**

H0: The series is not stationary

H1: The series is stationary

Dickey-Fuller= -7.8152, Lag order = 7, p-value = 0.01.

As p value is less than 0.05, we reject H0 and conclude that the series is stationary.

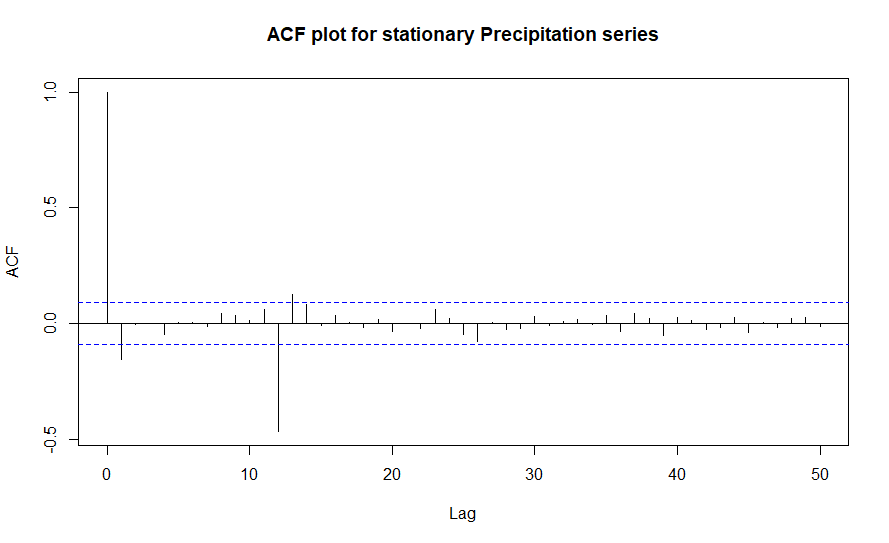


Fig. 42 ACF plot of Stationary Precipitation series

From the Fig.42, the value of seasonal MA order Q = 1 and non-seasonal MA order q = 1 are obtained.

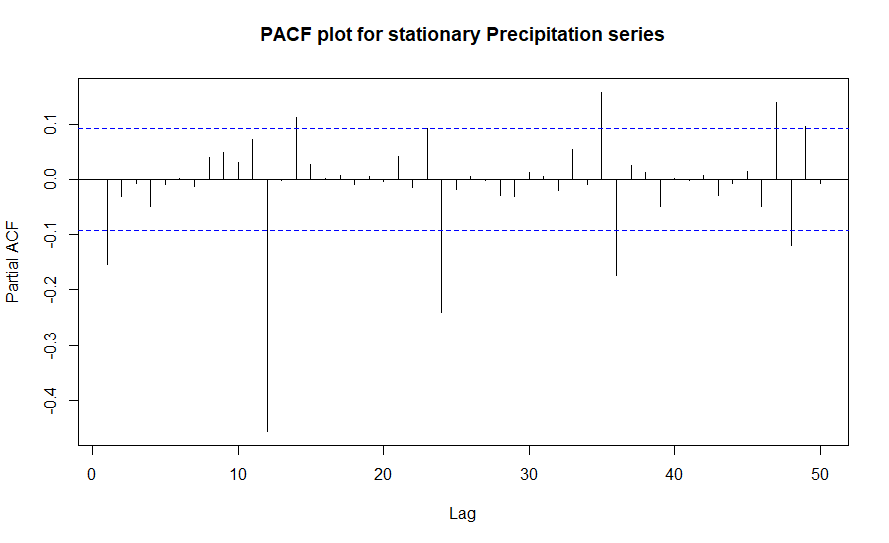


Fig. 43 PACF plot of Stationary Precipitation series

From Fig.43 we identified that the value of seasonal AR order P = 4 and non-seasonal AR order p = 1 are obtained.

And since the first seasonal difference series turns out to be stationary we put D=1 and here d=0.This gives only an initial idea for order of the model to be fitted. We choose the model with minimum AIC; however the accuracy of fitted model depends not only on AIC value but also on the assumptions of residuals. Usually the residuals are assumed to be uncorrelated. In the classical time series set up it is common to assume that the white noise sequence **Ɛt** is iid Gaussian (Box and Jenkins, 1976). To check the validity of these assumptions we use Ljung Box test and Box-pierce test.

Table. 18 Represents the summary of the fitted models

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **p** | **d** | **q** | **P** | **D** | **Q** | **LB** | **BP** | **IC** |
| 1 | 1 | 0 | 1 | 4 | 1 | 1 | 0.930919 | 0.952997 | 2757.01 |
| **2** | **1** | **0** | **1** | **3** | **1** | **1** | **0.945353** | **0.964549** | **2731.168** |
| 3 | 1 | 0 | 1 | 2 | 1 | 1 | 0.935467 | 0.958098 | 2733.851 |
| 4 | 1 | 0 | 1 | 1 | 1 | 1 | 0.821999 | 0.872562 | 2735.408 |

Here the model SARIMA (1, 0, 1) (3, 1, 1)12 has maximum p-value of Box- pierce method and L-jung Box method. Thus SARIMA (1, 0, 1) (3, 1, 1)12 is the best fitted model for monthly averaged Precipitation Corrected.

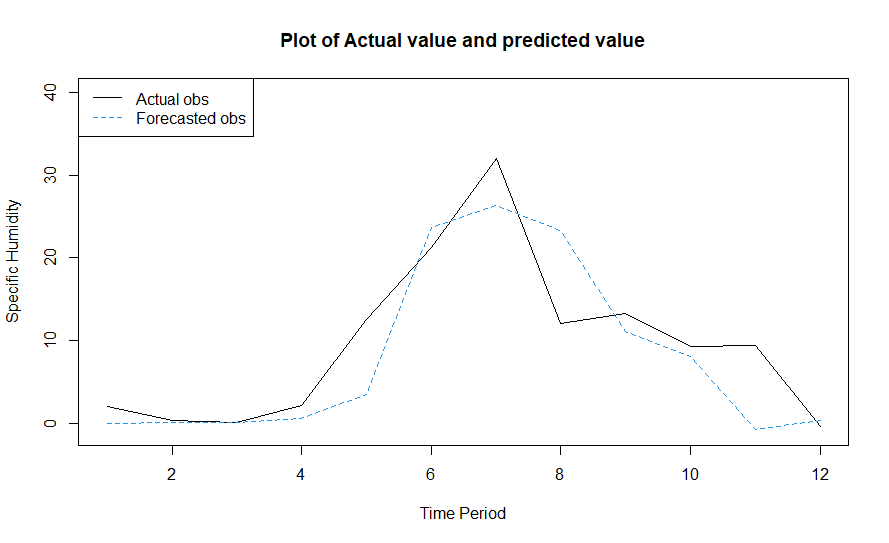


Fig. 44 Plot of Actual value and Predicted value

Table 19. Test values and SARIMA forecasted values:

|  |  |  |  |
| --- | --- | --- | --- |
| Month | Year | Actual Value | Predicted value |
| January | 2021 | 1.96401154 | -0.04971891 |
| February | 2021 | 0.27413946 | 0.04848261 |
| March | 2021 | 0.06931076 | 0.13675107 |
| April | 2021 | 2.13876599 | 0.62455439 |
| May | 2021 | 12.5742303 | 3.44234315 |
| June | 2021 | 21.2267063 | 23.62781814 |
| July | 2021 | 31.9570405 | 26.34872935 |
| August | 2021 | 12.0742544 | 23.3468005 |
| September | 2021 | 13.2249853 | 11.08723389 |
| October | 2021 | 9.28600055 | 8.04188538 |
| November | 2021 | 9.40205923 | -0.7665062 |
| December | 2021 | -0.3714075 | 0.37300022 |

**NEURAL NETWORKING**

**Forecasting from Multi-Layer Perceptions- Neural Networking**

MLP fit with 5 hidden nodes and 20 repetitions.

Series modelled in differences: D1.

Univariate lags: (1,2,3,4,5,6,7,8,9,10,11,12)

Forecast combined using the median operator.

MSE: 8.0256.

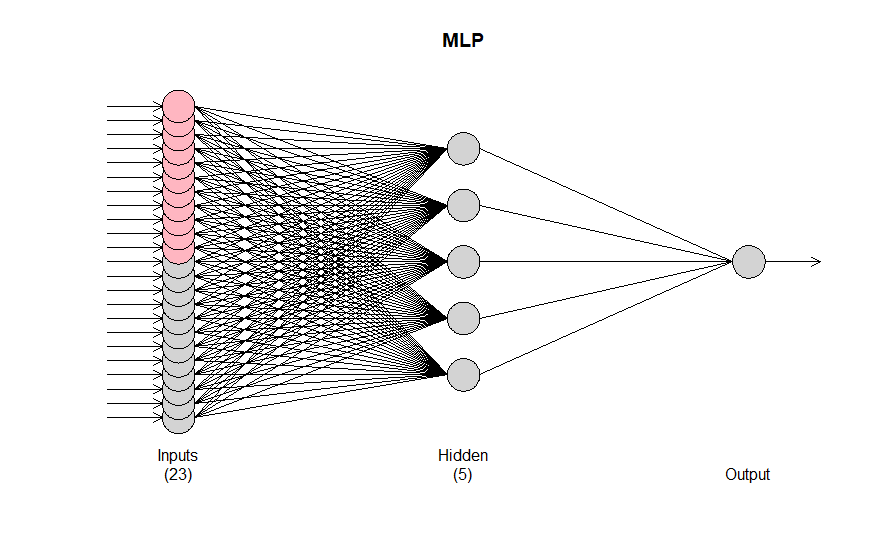


Fig. 45 Architecture of Multi-Layer Perception neural network model (MLP-NN).

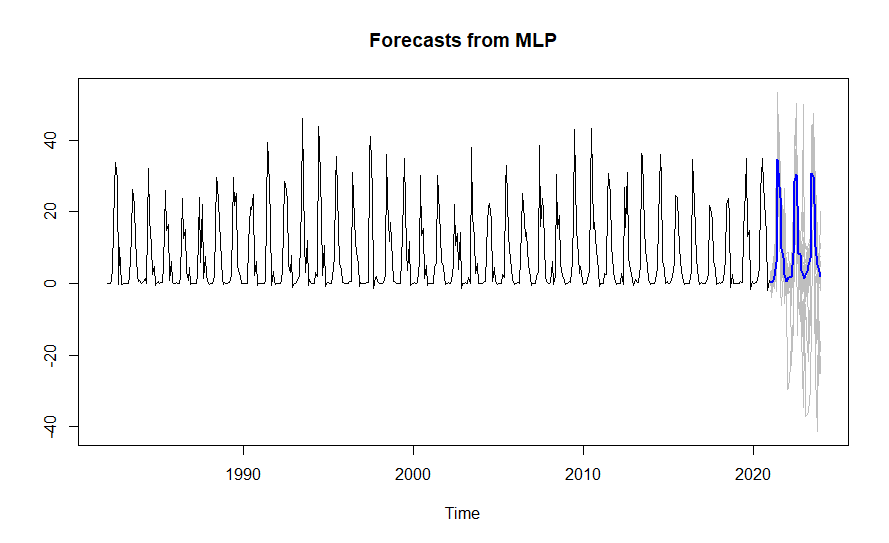


Fig. 46 Forecast plot for Precipitation using MLP

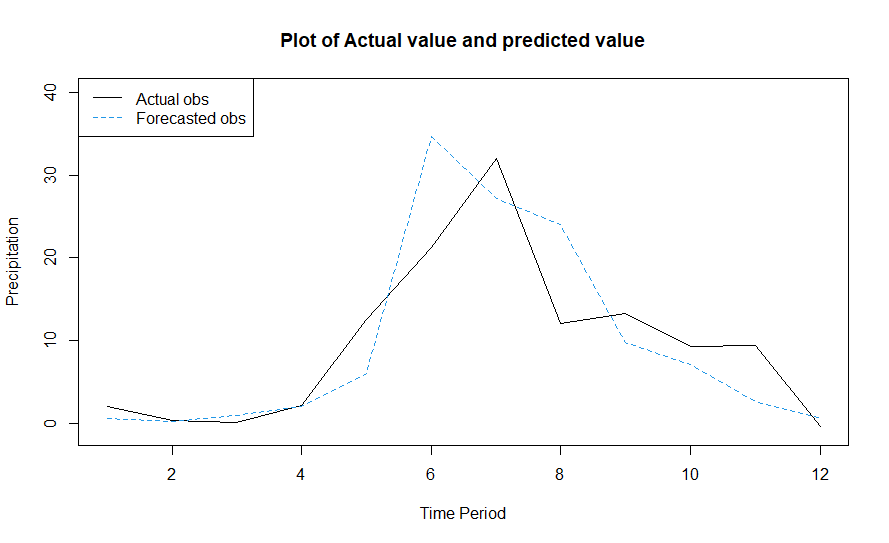


Fig. 47 Plot of Actual value and Predicted value using MLP-NN.

Table 20. Actual values and MLP-NN forecasted values:

|  |  |  |  |
| --- | --- | --- | --- |
| Month | Year | Actual Value | Predicted value |
| January | 2021 | 1.96401154 | 0.5952244 |
| February | 2021 | 0.27413946 | 0.2142818 |
| March | 2021 | 0.06931076 | 0.8888923 |
| April | 2021 | 2.13876599 | 2.0040372 |
| May | 2021 | 12.5742303 | 5.9864097 |
| June | 2021 | 21.2267063 | 34.6390186 |
| July | 2021 | 31.9570405 | 27.1315754 |
| August | 2021 | 12.0742544 | 24.017683 |
| September | 2021 | 13.2249853 | 9.7056058 |
| October | 2021 | 9.28600055 | 7.1337958 |
| November | 2021 | 9.40205923 | 2.6145659 |
| December | 2021 | -0.3714075 | 0.5624107 |

**Forecasting from Extreme learning machine-Neural Networking**

ELM fit with 100 hidden nodes and 20 repetitions.

Series modelled in differences: D1.

Univariate lags: (1,2,3,4,5,6,7,8,9,10,11,12)

Forecast combined using the median operator.

Output weight estimation using: lasso.

MSE: 24.804.

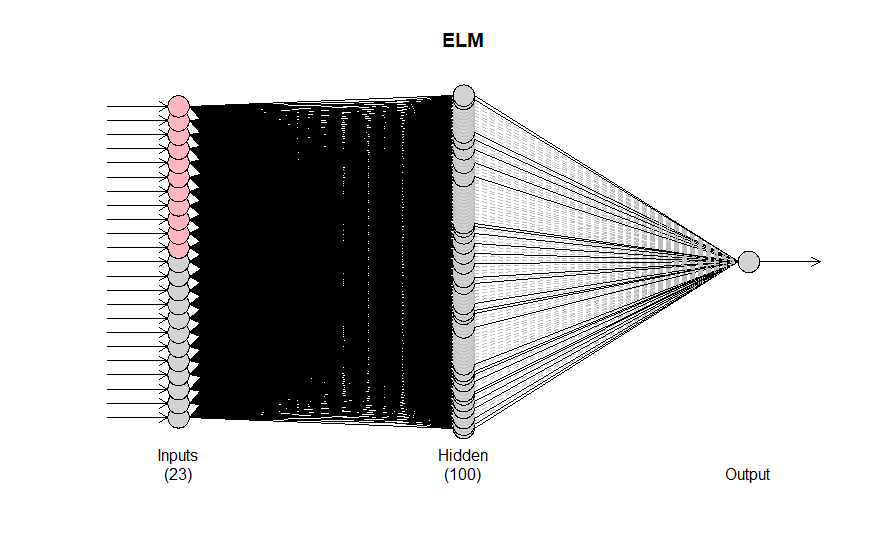


Fig.48 Architecture of Extreme Learning Machine neural network model (ELM-NN).

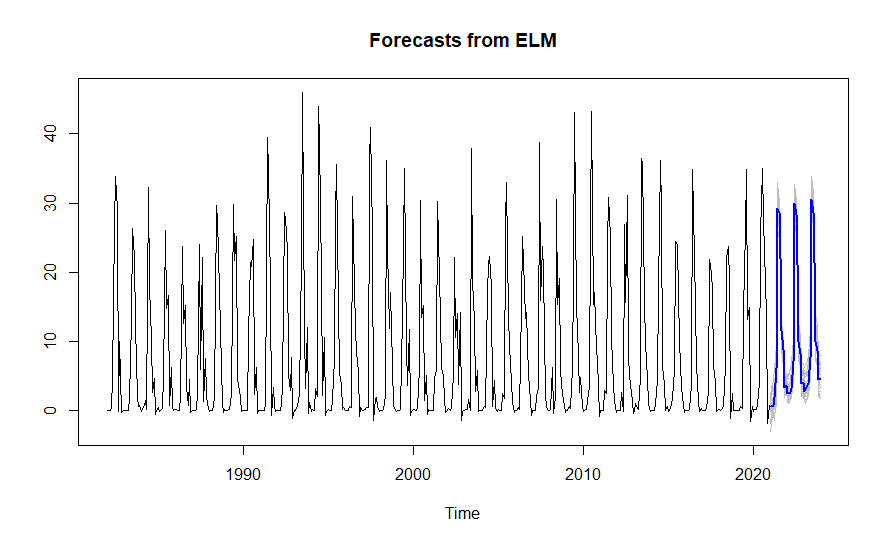


Fig. 49 Forecast of Precipitation using ELM-NN.

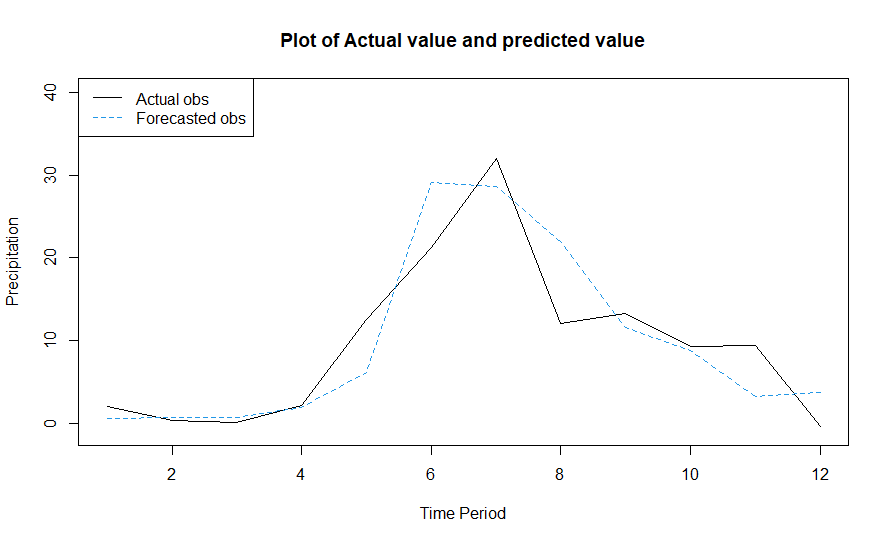


Fig. 50 Plot of Actual value and predicted value using ELM-NN

Table 21. Actual values and ELM-NN forecasted values:

|  |  |  |  |
| --- | --- | --- | --- |
| Month | Year | Actual Value | Predicted value |
| January | 2021 | 1.96401154 | 0.5766456 |
| February | 2021 | 0.27413946 | 0.633796 |
| March | 2021 | 0.06931076 | 0.7337911 |
| April | 2021 | 2.13876599 | 1.8472192 |
| May | 2021 | 12.57423025 | 6.1193175 |
| June | 2021 | 21.22670633 | 29.1522661 |
| July | 2021 | 31.95704045 | 28.6063144 |
| August | 2021 | 12.07425443 | 21.9933106 |
| September | 2021 | 13.22498525 | 11.5395178 |
| October | 2021 | 9.28600055 | 8.8240939 |
| November | 2021 | 9.40205923 | 3.2723475 |
| December | 2021 | -0.37140754 | 3.6561016 |

Table 22. Represents the Actual value and forecasted value of all models.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Month | Year | Actual Value | SARIMA | MLP-NN | ELM-NN |
| January | 2021 | 1.96401154 | -0.04971891 | 0.5952244 | 0.5766456 |
| February | 2021 | 0.27413946 | 0.04848261 | 0.2142818 | 0.633796 |
| March | 2021 | 0.06931076 | 0.13675107 | 0.8888923 | 0.7337911 |
| April | 2021 | 2.13876599 | 0.62455439 | 2.0040372 | 1.8472192 |
| May | 2021 | 12.57423025 | 3.44234315 | 5.9864097 | 6.1193175 |
| June | 2021 | 21.22670633 | 23.62781814 | 34.6390186 | 29.1522661 |
| July | 2021 | 31.95704045 | 26.34872935 | 27.1315754 | 28.6063144 |
| August | 2021 | 12.07425443 | 23.3468005 | 24.017683 | 21.9933106 |
| September | 2021 | 13.22498525 | 11.08723389 | 9.7056058 | 11.5395178 |
| October | 2021 | 9.28600055 | 8.04188538 | 7.1337958 | 8.8240939 |
| November | 2021 | 9.40205923 | -0.7665062 | 2.6145659 | 3.2723475 |
| December | 2021 | -0.37140754 | 0.37300022 | 0.5624107 | 3.6561016 |

Fig. 51 Comparison between Actual value and Forecasted value of all model

Table 23: Represents the Accuracy measures RMSE, MAE and MAPE for all the models.

|  |  |  |  |
| --- | --- | --- | --- |
| Model | RMSE | MAE | MAPE |
| SARIMA | 5.508745 | 3.877478 | 252.8279 |
| MLP-NN | 5.769758 | 3.882255 | 264.2885 |
| ELM-NN | 4.774275 | 3.554825 | 210.2313 |

In terms of RMSE, MAE and MAPE ELM-NN model is better forecasting model compare to all other fitted models. Because ELM-NN has minimum error statistics compared to all other model.

Table 24: Represents the forecasted Precipitation for next 19 months using ELM-NN model.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Month | Year | Forecasted value | Month | Year | Forecasted value |
| June | 2022 | 29.9778553 | April | 2023 | 3.9474547 |
| July | 2022 | 28.3183698 | May | 2023 | 7.6645837 |
| August | 2022 | 21.16918 | June | 2023 | 30.4593903 |
| September | 2022 | 9.9756587 | July | 2023 | 28.5957906 |
| October | 2022 | 8.202889 | August | 2023 | 21.3898633 |
| November | 2022 | 3.8872712 | September | 2023 | 10.0039469 |
| December | 2022 | 4.127471 | October | 2023 | 8.7116675 |
| January | 2023 | 2.7419113 | November | 2023 | 4.4305232 |
| February | 2023 | 3.1898548 | December | 2023 | 4.6980035 |
| March | 2023 | 3.5818031 |  |  |  |

**CHAPTER 4**

**Conclusions**

* Studied the variables under study using measure of central tendency and measure of dispersion.
* From skewness we observed that Dew temperature, Wet temperature, Specific humidity and Relative humidity are negatively skewed and all other variables are positively skewed.
* All variables under study are platykurtic.
* We observed multicollinearity, autocorrelation and heteroscedasticity in the study dataset and we removed autocorrelation for further analysis.
* The factor affecting Precipitation are Surface Temperature, Surface Pressure, Specific humidity at 2 meter, Relative humidity at 2 meter, Temperature at 2 meter and Wind speed at 2 meter.
* Among all predictive models under study, Lasso is better for estimating Precipitation.
* We developed appropriate traditional time series model and artificial neural network for Surface Temperature, Surface Pressure and Precipitation.
* From the analysis we found that,
* MLP is better model for forecasting Surface Temperature.
* ELM is better model for forecasting Surface Pressure.
* ELM is better model for forecasting Precipitation.
* Forecasted the values for Surface Temperature, Surface Pressure and Precipitation for 19 months.

Reference:-

* Azad Abdulolhafedh (2017) “Detecting and Removal of Temporal Autocorrelation in Vehicular Crash Data”: Article in Journal of Transportation Technologies 2017.
* Dash.Y. et al. (2018) “Rainfall Prediction for the Kerala state using Artificial Intelligence approach”: Computers and Electrical Engineering,70,66-73.
* Kashiwao, T., et al. (2017) “A neural-network based local rainfall prediction system using meteorological data on internet”: Applied soft computing, 56, 317-330.
* Dutta, P. S., & Tahbilder, H. (2014). “Prediction of rainfall using data mining technique over Assam”. Indian Journal of Computer Science and Engineering (IJCSE), 5(2), 85-90.
* Sultana et al. (2017, July) “An application of data mining and machine learning for weather forecasting”. In International Conference on Computing and Information Technology (pp. 169-178). Springer, Cham.

**APPENDIX:**

x=read.csv("Data.csv",header=T)

#Descriptive Statistics

#install.packages("psych")

library(psych)

psych::describe(x)

attach(X)

y=PRECTOTCORR

x=cbind(TS,PS,T2MDEW,T2MWET,QV2M,RH2M,T2M,WS2M)

n=nrow(x)

### CHECK FOR MULTICOLINEARITY

model1=step(lm(PRECTOTCORR~TS+PS+T2MDEW+T2MWET+QV2M+RH2M+T2M+WS2M,data=X))

summary(model1)

#install.packages("car")

library(car)

a=vif(model1)

### CHECK FOR AUTOCORRELATION

DW=durbinWatsonTest(model1)

round(cor(X),digits=2)

###Removal of Auto Correlation

beta=solve(t(x)%\*%x)%\*%t(x)%\*%y

ycap=x%\*%beta

e=y-ycap

d1=0

for(i in 2:n){

d1[i]=((e[i]-e[i-1])^2)

}

d2=sum(d1)

d3=sum(e^2)

dw=(d2/d3)

SSR=sum((ycap-mean(y))^2)

SSE=sum(e^2)

SST=SSE+SSR

Rsq=SSR/SST

library(Metrics)

RMSE=rmse(y,ycap)

##finding rho

d4=0

for(i in 2:n){

d4[i]=(e[i]\*e[i-1])

}

d5=sum(d4)

d6=sum(e^2)

rho=(d5/d6)

##Ystar and xstar after

Ytstar={}

for(i in 2:n){

Ytstar[i]=y[i]-(rho\*y[i-1])

}

head(Ytstar)

Xtstar1={}

for(i in 2:n){

Xtstar1[i]=x[i,1]-(rho\*x[i-1,1])

}

Xtstar2={}

for(i in 2:n){

Xtstar2[i]=x[i,2]-(rho\*x[i-1,2])

}

Xtstar3={}

for(i in 2:n){

Xtstar3[i]=x[i,3]-(rho\*x[i-1,3])

}

Xtstar4={}

for(i in 2:n){

Xtstar4[i]=x[i,4]-(rho\*x[i-1,4])

}

Xtstar5={}

for(i in 2:n){

Xtstar5[i]=x[i,5]-(rho\*x[i-1,5])

}

Xtstar6={}

for(i in 2:n){

Xtstar6[i]=x[i,6]-(rho\*x[i-1,6])

}

Xtstar7={}

for(i in 2:n){

Xtstar7[i]=x[i,7]-(rho\*x[i-1,7])

}

Xtstar8={}

for(i in 2:n){

Xtstar8[i]=x[i,8]-(rho\*x[i-1,8])

}

Xtstar=cbind(Xtstar1,Xtstar2,Xtstar3,Xtstar4,Xtstar5,Xtstar6,Xtstar7,Xtstar8);head(Xtstar)

beta1=solve(t(Xtstar[-1,])%\*%Xtstar[-1,])%\*%t(Xtstar[-1,])%\*%Ytstar[-1]

ycap1=Xtstar[-1,]%\*%beta1

e1=Ytstar[-1]-ycap1

d11=0

for(i in 2:n){

d11[i]=((e1[i]-e1[i-1])^2)

}

d21=sum(d11,na.rm=T)

d41=sum(e1^2,na.rm=T)

dw1=(d21/d41);dw1

SSR1=sum((ycap1-mean(Ytstar[-1]))^2);SSR1

SSE1=sum(e1^2);SSE1

SST1=SSE1+SSR1;SST1

Rsq1=SSR1/SST1;Rsq1

RMSE1=rmse(Ytstar[-1],ycap1);RMSE1

## CHECK FOR AUTOCORRELATION AFTER REMOVAL OF AUTOCORRELATION

n1=nrow(Xtstar[-1,]);n1

model2=step(lm(Ytstar[-1]~Xtstar1[-1]+Xtstar2[-1]+Xtstar3[-1]+Xtstar4[-1]+Xtstar5[-1]+Xtstar6[-1]+Xtstar7[-1]+Xtstar8[-1]))

summary(model1)

durbinWatsonTest(model2)

a1=vif(model2)

D=cbind(Ytstar,Xtstar1,Xtstar2,Xtstar3,Xtstar4,Xtstar5,Xtstar6,Xtstar7,Xtstar8);head(D)

#write.csv(D,"DataCorr.csv")

### CHECK FOR HETEROSCEDASTICITY

library(lmtest)

gqtest(model2,order.by=~Xtstar1[-1]+Xtstar2[-1]+Xtstar3[-1]+Xtstar4[-1]+Xtstar5[-1]+Xtstar6[-1]+Xtstar7[-1]+Xtstar8[-1],fraction=7)

data=read.csv("DataCorr.csv",header=T)

d=data[,2:10]

#install.packages("caret") #Need to install lava and stringi packages along with caret

#install.packages("mlbench")

#install.packages("glmnet")

library(caret)

library(mlbench)

library(psych)

library(glmnet)

library(MASS)

attach(d)

y=d$PRECTOTCORR[-1] #PRECTOTCORR

x1=d$TS[-1] #TS

x2=d$PS[-1] #PS

x3=d$T2MDEW[-1] #T2MDEW

x4=d$T2MWET[-1] #T2MWET

x5=d$QV2M[-1] #QV2M

x6=d$RH2M[-1] #RH2M

x7=d$T2M[-1] #T2M

x8=d$WS2M[-1] #WS2M

x=cbind(x1,x2,x3,x4,x5,x6,x7,x8)

ind=sample(2,nrow(d),replace=TRUE,prob=c(0.7,0.3))

train=d[ind==1,]

nrow(train)

test=d[ind==2,]

ytrain=train$PRECTOTCORR

custom=trainControl(method="repeatedcv",number=10,repeats=5,verboseIter=T)

lm=train(PRECTOTCORR[-1]~TS[-1]+PS[-1]+T2MDEW[-1]+T2MWET[-1]+QV2M[-1]+RH2M[-1]+T2M[-1]+WS2M[-1],train,method="lm",trControl=custom,na.rm=TRUE)

lm$results

lm

summary(lm)

ft=fitted(lm)

plot(lm$finalModel)

# Ridge regression

ridge=train(PRECTOTCORR[-1]~TS[-1]+PS[-1]+T2MDEW[-1]+T2MWET[-1]+QV2M[-1]+RH2M[-1]+T2M[-1]+WS2M[-1],train,method="glmnet",tuneGrid=expand.grid(alpha=0,lambda=seq(0.001,1,length=15)),trControl=custom)

plot(ridge)

ridge

plot(ridge$finalModel,xvar="lambda",label=T)

plot(ridge$finalModel,xvar="dev",label=T)

plot(varImp(ridge,scale=F))

# Lasso regression

lasso=train(PRECTOTCORR[-1]~TS[-1]+PS[-1]+T2MDEW[-1]+T2MWET[-1]+QV2M[-1]+RH2M[-1]+T2M[-1]+WS2M[-1],train,method="glmnet",tuneGrid=expand.grid(alpha=1,lambda=seq(0.001,1,length=10)),trControl=custom)

plot(lasso)

lasso

plot(lasso$finalModel,xvar="lambda",label=T)

plot(lasso$finalModel,xvar="dev",label=T)

plot(varImp(lasso,scale=F))

#Elastic net regression

ELN=train(PRECTOTCORR[-1]~TS[-1]+PS[-1]+T2MDEW[-1]+T2MWET[-1]+QV2M[-1]+RH2M[-1]+T2M[-1]+WS2M[-1],train,method="glmnet",tuneGrid=expand.grid(alpha=seq(0,1,length=15),lambda=seq(0.001,1,length=10)),trControl=custom)

plot(ELN)

ELN

plot(ELN$finalModel,xvar="lambda",label=T)

plot(ELN$finalModel,xvar="dev",label=T)

plot(varImp(ELN,scale=F))

### Surface Temperature

y=TS[13:492];head(y)

plot.ts(TS,main='Time profile for surface temperature',ylab='Surface Temperature')

n=length(y);n

train=y[1:468]

n1=length(train);n1

test=y[469:480];test

ys=ts(train,frequency=12,start=c(1982,1));ys

plot.ts(ys,main="Time profile of Surface temperature",ylab="Surface Temperature")

m=round(2\*sqrt(length(ys)));m

acf(c(ys),m,main="ACF of Observed series",na.rm=T)

##Test for trend

#install.packages("Kendall")

library(Kendall)

MannKendall(ys)

##Test for seasonality

s=12

c=39

mx=matrix(ys,s,c)

rn={}

for(j in 1:c){

rn[j]=rank(mx[,j])

}

M=matrix(rn,s,c)

m=rowSums(M)

i={}

for(l in 1:s){

i[l]=(m[l]-(c\*(s-1)/2))^2

}

sum(i)

chi=12\*sum(i)/(c\*s\*(s+1));chi

chit=qchisq(0.95,(s-1));chit

if(chi>chit)

cat("We reject Ho and conclude that data contains seasonality")else

cat("We accept Ho and conclude that data doesn't contains seasonality")

##Variate differencing

v=var(ys);v

d1=diff(ys);head(d1)

v1=var(d1);v1

if(v<v1)

cat("ys is stationary")else

cat("ys is not stationary")

d2=diff(d1)

v2=var(d2)

if(v1<v2)

cat("d1 is stationary")else

cat("d1 is not stationary")

plot.ts(d1,main="Plot of Stationary series")

##Augmented Dickey-Fuller Test

library(tseries)

adf.test(d1)

###ARIMA model building

d=1

acf(d1,main="ACF plot for stationary surface temperature")

q=4

pacf(d1,main="PACF plot for stationary surface temperature")

p=3

fit1=arima(train,order=c(3,1,4))

res1=fit1$residuals

k=round(2\*sqrt(n1));k

LB1=Box.test(res1,lag=k,type="Ljung-Box")

BP1=Box.test(res1,lag=k,type="Box-Pierce")

SP1=shapiro.test(res1)

IC1=fit1$aic

fit2=arima(train,order=c(3,1,3))

res2=fit2$residuals

k=round(2\*sqrt(n1));k

LB2=Box.test(res2,lag=k,type="Ljung-Box")

BP2=Box.test(res2,lag=k,type="Box-Pierce")

SP2=shapiro.test(res2)

IC2=fit2$aic

fit3=arima(train,order=c(3,1,2))

res3=fit3$residuals

k=round(2\*sqrt(n1));k

LB3=Box.test(res3,lag=k,type="Ljung-Box")

BP3=Box.test(res3,lag=k,type="Box-Pierce")

SP3=shapiro.test(res3)

IC3=fit3$aic

fit4=arima(train,order=c(3,1,1))

res4=fit4$residuals

k=round(2\*sqrt(n1));k

LB4=Box.test(res4,lag=k,type="Ljung-Box")

BP4=Box.test(res4,lag=k,type="Box-Pierce")

SP4=shapiro.test(res4)

IC4=fit4$aic

fit5=arima(train,order=c(2,1,4))

res5=fit5$residuals

k=round(2\*sqrt(n1));k

LB5=Box.test(res5,lag=k,type="Ljung-Box")

BP5=Box.test(res5,lag=k,type="Box-Pierce")

SP5=shapiro.test(res5)

IC5=fit5$aic

fit6=arima(train,order=c(2,1,3))

res6=fit6$residuals

k=round(2\*sqrt(n1));k

LB6=Box.test(res6,lag=k,type="Ljung-Box")

BP6=Box.test(res6,lag=k,type="Box-Pierce")

SP6=shapiro.test(res6)

IC6=fit6$aic

fit7=arima(train,order=c(2,1,2))

res7=fit7$residuals

k=round(2\*sqrt(n1));k

LB7=Box.test(res7,lag=k,type="Ljung-Box")

BP7=Box.test(res7,lag=k,type="Box-Pierce")

SP7=shapiro.test(res7)

IC7=fit7$aic

fit8=arima(train,order=c(2,1,1))

res8=fit8$residuals

k=round(2\*sqrt(n1));k

LB8=Box.test(res8,lag=k,type="Ljung-Box")

BP8=Box.test(res8,lag=k,type="Box-Pierce")

SP8=shapiro.test(res8)

IC8=fit8$aic

fit9=arima(train,order=c(1,1,4))

res9=fit9$residuals

k=round(2\*sqrt(n1));k

LB9=Box.test(res9,lag=k,type="Ljung-Box")

BP9=Box.test(res9,lag=k,type="Box-Pierce")

SP9=shapiro.test(res9)

IC9=fit9$aic

fit10=arima(train,order=c(1,1,3))

res10=fit10$residuals

k=round(2\*sqrt(n1));k

LB10=Box.test(res10,lag=k,type="Ljung-Box")

BP10=Box.test(res10,lag=k,type="Box-Pierce")

SP10=shapiro.test(res10)

IC10=fit10$aic

fit11=arima(train,order=c(1,1,2))

res11=fit11$residuals

k=round(2\*sqrt(n1));k

LB11=Box.test(res11,lag=k,type="Ljung-Box")

BP11=Box.test(res11,lag=k,type="Box-Pierce")

SP11=shapiro.test(res11)

IC11=fit11$aic

fit12=arima(train,order=c(1,1,1))

res12=fit12$residuals

k=round(2\*sqrt(n1));k

LB12=Box.test(res12,lag=k,type="Ljung-Box")

BP12=Box.test(res12,lag=k,type="Box-Pierce")

SP12=shapiro.test(res12)

IC12=fit12$aic

Model=c(1,2,3,4,5,6,7,8,9,10,11,12)

p=c(3,3,3,3,2,2,2,2,1,1,1,1)

d=c(rep(1,12))

q=c(4,3,2,1,4,3,2,1,4,3,2,1)

LB=c(LB1$p.value,LB2$p.value,LB3$p.value,LB4$p.value,LB5$p.value,LB6$p.value,LB7$p.value,LB8$p.value,LB9$p.value,LB10$p.value,LB11$p.value,LB12$p.value)

BP=c(BP1$p.value,BP2$p.value,BP3$p.value,BP4$p.value,BP5$p.value,BP6$p.value,BP7$p.value,BP8$p.value,BP9$p.value,BP10$p.value,BP11$p.value,BP12$p.value)

SP=c(SP1$p.value,SP2$p.value,SP3$p.value,SP4$p.value,SP5$p.value,SP6$p.value,SP7$p.value,SP8$p.value,SP9$p.value,SP10$p.value,SP11$p.value,SP12$p.value)

IC=c(IC1,IC2,IC3,IC4,IC5,IC6,IC7,IC8,IC9,IC10,IC11,IC12)

df=data.frame("Model"=Model,"p"=p,"d"=d,"q"=q,"LB"=LB,"BP"=BP,"SP"=SP,"AIC"=IC)

cat("\n Residual analysis table is\n")

print(df)

max(SP)

min(IC)

plot(res2,main="Plot of residual series of ARIMA(3,0,3)")

##Model perfomance and Forecast

pre=predict(fit2,n.ahead=36)

prev=pre$pred

cat("\n Predicted values of data are ",prev)

data.frame(test,prev[1:12])

z=as.vector(test);z

z1=as.vector(prev[1:12])

plot(z,type="l",lty=1,col=1,ylim=c(16,32),xlab="TimePeriod",ylab="Surface Temperature",main="Plot of Actual value and predicted value")

lines(z1,lty=2,col=4)

#install.packages("nnfor")

library(nnfor)

tab=accuracy(prev[1:12],test);tab

forecast=prev[13:36];forecast

### MLP

#install.packages("e1071")

library(e1071)

mlp.fit<-mlp(ys)

mlp.fit

plot(mlp.fit)

print(mlp.fit)

mlp.frc1=forecast(mlp.fit,h=12);mlp.frc1

mlp.frc=forecast(mlp.fit,h=36);mlp.frc

plot(mlp.frc,ylab='Temperature',xlab='Date',main='Forecast using MLP')

tab1=accuracy(mlp.frc1,test);tab1

z2=as.vector(c(mlp.frc1$mean));z2

plot(z,type="l",lty=1,col=1,ylim=c(16,32),xlab="TimePeriod",ylab="Surface Temperature",main="Plot of Actual value and predicted value")

lines(z2,lty=2,col=4)

legend("topleft",legend=c("Actual obs","Forecasted obs"),lty=c(1,2),col=c(1,4))

data.frame(test,mlp.frc1$mean)

###ELM

elm.fit=elm(ys)

print(elm.fit)

plot(elm.fit)

elm.frc=forecast(elm.fit,h=36);elm.frc

elm.frc1=forecast(elm.fit,h=12);elm.frc1

tab2=accuracy(elm.frc1,test);tab2

plot(elm.frc,ylab='Temperature',xlab='Date',main='Forecast using ELM')

z3=as.vector(c(elm.frc1$mean));z2

plot(z,type="l",lty=1,col=1,ylim=c(16,32),xlab="TimePeriod",ylab="Surface Temperature",main="Plot of Actual value and predicted value")

lines(z3,lty=2,col=4)

legend("topleft",legend=c("Actual obs","Forecasted obs"),lty=c(1,2),col=c(1,4))

data.frame(test,elm.frc1$mean)