International Journal of Current Research and Modern Education (IJCRME)
Impact Factor: 6.725, ISSN (Online): 2455 - 5428
(www.rdmodernresearch.com) Volume 2, Issue 2, 2017

RAINFALL FORECASTOF ANDHRA PRADESH USING ARTIFICIAL NEURAL NETWORKS

J. C. Ramesh Reddy*, T. Ganesh, M. Venkateswaran*** & P. R. S. Reddy***

* Department of Statistics, Sri Venkateswara University, Tirupati, Andhra Pradesh
** Department of Mathematics, NIT Hamirpur, Himachal Pradesh
*** Department of H&S, S.V Engineering College, Tirupati, Andhra Pradesh

Cite This Article: J. C. Ramesh Reddy, T. Ganesh, M. Venkateswaran & P. R. S. Reddy, "Rainfall Forecast of Andhra Pradesh Using Artificial Neural Networks", International Journal of Current Research and Modern Education, Volume 2, Issue 2, Page Number 223-234, 2017.

Abstract:

Forecasting monthly mean rainfall of Andhra Pradesh (India)using seasonal autoregressive integrated moving average (SARIMA) modeland artificial neural networks (ANN)has been discussed.In this paper, we have given the prediction values according to SARIMA and neural network models, in whichwe found that the ARIMA (1,0,0)(2,0,0)[12] for actual dataand ARIMA (3,0,0)(2,0,0)[12] for rainfall differenceshas been fitted.The significance test has been made by using lowest AIC and BIC values.

KeyWords: Box-Jenkins Methodology, SARIMA Model, Neural Network, AIC& BIC.

1. Introduction:

Andhra Pradesh is one of the states of India; we have taken the rainfall data. In this, we performed the forecasting of annual rainfall of Andhra Pradeshfor coming years. For the experiment, we have taken data of Mean Annual Rainfall from www.data.gov.in. The data is having the information of mean annual rainfall from year 1901 to 2016. In this experiment we have taken the help of R programming to fit the ARIMA models and SPSS for neural network. For the analysis, first column of the dataset is chosen to do analysis that is having annual mean rainfall information in mm unit.

2. Methodology:

ARIMA models are capable of modeling a wide range of seasonal data. A seasonal ARIMA model is formed by including additional seasonal terms in the ARIMA models we have seen so far. It is written as follows: ARIMA (p, d, q) $(P, D, Q)_m$: the first parenthesis represents the non-seasonal part of the model and second represents the seasonal part of the model, where m= number of periods per season. We use uppercase notation for the seasonal parts of the model, and lowercase notation for the non-seasonal parts of the model. The additional seasonal terms are simply multiplied with the non-seasonal terms.

2.1 ACF/PACF:

The seasonal part of an AR or MA model will be seen in the seasonal lags of the PACF and ACF. For example, an ARIMA $(0,0,0)(0,0,1)_{12}$ model will show: a spike at lag 12 in the ACF but no other significant spikes.The PACF will show exponential decay in the seasonal lags; that is, at lags 12, 24, 36,….etc. Similarly, an ARIMA$(0,0,0)(1,0,0)_{12}$ model will show:exponential decay in the seasonal lags of the ACFa single significant spike at lag 12 in the PACF.In considering the appropriate seasonal orders for an ARIMA model, restrict attention to the seasonal lags. The modeling procedure is almost the same as for non-seasonal data, except that we need to select seasonal AR and MA terms as well as the non-seasonal components of the model.
Seasonal autoregressive integrated moving average (SARIMA) model for any variable involves mainly four steps: Identification, Estimation, Diagnostic checking and Forecasting. The basic form of SARIMA model is denoted by

$$SARIMA(p,d,q)X(P,D,Q)_S$$

and the model is given by

$$\phi_p(B)\Phi_P(B^s)\nabla^d\nabla_s^D Z_t = \theta_q(B)\Theta_Q(B^s)a_t,$$

Where Z_t is the time series value at time t and $\phi, \Phi, \theta \, and \, \Theta$ are polynomials of order of p, P, q and Q respectively. B is the backward shift operator, $B^s Z_t = Z_{t-s}$ and $\nabla = (1-B)$. Order of seasonality is represented by s. Non-seasonal and seasonal difference orders are denoted by d and D respectively. White noise process is denoted by a_t . The Box-Jenkins methodology involves four steps (Box et al., 1994): (i) identification (ii) estimation (iii) diagnostic checking and (iv) forecasting. First, the original series must be transformed to become stationary around its mean and its variance. Second, the appropriate order of p and q must be specified using autocorrelation and partial autocorrelation functions. Third, the value of the parameters must be estimated using some non-linear optimization procedure that minimizes the sum of squares of the errors or some other appropriate loss function. Diagnostic checking of the model adequacy is required in the fourth step. This

procedure is continued until an adequate model is obtained. Finally, the future forecasts generate using minimum mean square error method (Box et al. 1994). SARIMA models are used as benchmark models to compare the performance of the other models developed on the same data set. The iterative procedure of SARIMA model building was explained by Kumari et al. (2013), Boiroju (2012), Rao (2011) and Box et al. (1994).

2.1.1 Arima:

By default, the arima command in R sets c=μ=0 when d>0 and provides an estimate of μ when d=0. The parameter μ is called the "intercept" in the R output. It will be close to the sample mean of the time series, but usually not identical to it as the sample mean is not the maximum likelihood estimate when p+q>0. The arima command has an argument include.mean which only has an effect when d=0 and is TRUE by default. Setting include.mean=FALSE will force μ=0.

The Arima command from the forecast package provides more flexibility on the inclusion of a constant. It has an argument include.mean which has identical functionality to the corresponding argument for arima. It also has an argument include.drift which allows μ≠0μ≠0 when d=1. For d>1, no constant is allowed as a quadratic or higher order trend is particularly dangerous when forecasting. The parameter μμ is called the "drift" in the R output when d=1.

This is also an argument include.constant which, if TRUE will see include.mean=TRUE if d=0 and include.drift=TRUE when d=1. If include.constant=FALSE. Both include.mean and include.drift will be set to FALSE. If include.constant is used, the values of include.mean=TRUE and include.drift=TRUE are ignored.

2.1.2 Auto.Arima

The auto.arima function automates the inclusion of a constant. By default, for d=0 or d=1, a constant will be included if it improves the AIC value; for d>1, the constant is always omitted. If allow drift=FALSE is specified, then the constant is only allowed when d=0.

There is another function arima in R which also fits an ARIMA model. However, it does not allow for the constant cc unless d=0, and it does not return everything required for the forecast function. Finally, it does not allow the estimated model to be applied to new data (which is useful for checking forecast accuracy). Consequently, it is recommended that you use Arima instead.

2.1.3 Modeling Procedure:

When fitting an ARIMA model to a set of time series data, the following procedure provides a useful general approach.

- ✓ Plot the data. Identify any unusual observations.
- ✓ If necessary, transform the data (using a Box-Cox transformation) to stabilize the variance.
- ✓ If the data are non-stationary: take first differences of the data until the data are stationary.
- ✓ Examine the ACF/PACF: Is an AR(pp) or MA(qq) model appropriate?
- ✓ Try your chosen model(s), and use the AICc to search for a better model.
- ✓ Check the residuals from your chosen model by plotting the ACF of the residuals, and doing a portmanteau test of the residuals. If they do not look like white noise, try a modified model.
- ✓ Once the residuals look like white noise, calculate forecasts.

AIC and BIC are both penalized-likelihood criteria. They are sometimes used for choosing best predictor subsets in regression and often used for comparing non-nested models, which ordinary statistical tests cannot do. The AIC or BIC for a model is usually written in the form [-2log$L$ + kp], where L is the likelihood function, p is the number of parameters in the model, and k is 2 for AIC and log(n) for BIC.

AIC is an estimate of a constant plus the relative distance between the unknown true likelihood function of the data and the fitted likelihood function of the model, so that a lower AIC means a model is considered to be closer to the truth. BIC is an estimate of a function of the posterior probability of a model being true, under a certain Bayesian setup, so that a lower BIC means that a model is considered to be more likely to be the true model. Both criteria are based on various assumptions and asymptotic approximations. Each, despite its heuristic usefulness, has therefore been criticized as having questionable validity for real world data. But despite various subtle theoretical differences, their only difference in practice is the size of the penalty; BIC penalizes model complexity more heavily. The only way they should disagree is when AIC chooses a larger model than BIC.

AIC and BIC are both approximately correct according to a different goal and a different set of asymptotic assumptions. Both sets of assumptions have been criticized as unrealistic. Understanding the difference in their practical behavior is easiest if we consider the simple case of comparing two nested models. In such a case, several authors have pointed out that IC's become equivalent to likelihood ratio tests with different alpha levels. Checking a chi-squared table, we see that AIC becomes like a significance test at alpha=.16, and BIC becomes like a significance test with alpha depending on sample size, e.g., .13 for n = 10, .032 for n = 100, .0086 for n = 1000, .0024 for n = 10000. Remember that power for any given alpha is increasing in n. Thus, AIC always has a chance of choosing too big a model, regardless of n. BIC has very little

chance of choosing too big a model if *n* is sufficient, but it has a larger chance than AIC, for any given *n*, of choosing too small a model.

So what's the bottom line? In general, it might be best to use AIC and BIC together in model selection. For example, in selecting the number of latent classes in a model, if BIC points to a three-class model and AIC points to a five-class model, it makes sense to select from models with 3, 4 and 5 latent classes. AIC is better in situations when a false negative finding would be considered more misleading than a false positive, and BIC is better in situations where a false positive is as misleading as, or more misleading than, a false negative.

2.2 Artificial Neural Networks:

First, Neural Networks are the algorithms (computer applications) that you use when you employ a techniquecalled Neural Computing. Neural Networks are very good at detecting patterns in data that may be too complexfor us to easily recognize. Using the recognized patterns they can tell us what data is important to the patternsand how to use the patterns to interpret new data, including often how to predict future patterns. You canconsider a regression model as just a specialized case of a Neural Network because, like regression, theknowledge of relationships between the data is stored in the weights (coefficients) that exist inside the NeuralNetwork.

Neural Networks consists of two stages: the training and testing stages. The data is divided into two sets– a training (or model building) set and a testing set (as done in prior lessons for multiple regression analysis).The majority of the data is in the training set and the testing set is used to see how well themodel performed.

Neural Network architecture, which is comprised of many nodes (perceptrons) arranged in three layers

- ✓ Input layer – the entrances to the network,
- ✓ Hidden layer – the turns inside the network, and
- ✓ Output layer – the center of the network.

The input layer is comprised of all the inputs that you are planning to use in the data analysis, such as square feetof living area and number of bedrooms.

The output layer is comprised of the expected output or desired results and often a bias node, which is used toaccumulate the error from all of the nodes in order to adjust the weights between nodes for the next trial.

The hidden layer is where all the work happens and is comprised of layers of interconnected perceptrons, which use mathematical concepts to determine the turns the data should take. These concepts range from simplesummations using weights, which is very similar to linear regression, to the application of vector calculus andclustering to determine the path that is the closest to the reality expressed by the data (more on that in thesupplement for those interested). Usually there is only one layer of perceptrons in the hidden layer; however,some models use two or more layers. In the simplest model, each perceptron in the hidden layer is connected toall of the perceptrons in the input layer and all outputs from the final perceptrons of the hidden layer feed in to the output layer perceptrons.

2.2.1 Feed Forward Neural Networks:

An artificial neural network, usually called neural networks, is a mathematical model or computational model that is inspired by the structure and/or functional aspects of biological neural networks. A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation. In a feed forward neural network (FFNN) structure, the only appropriate connections are between the outputs of each layer and the inputs of the next layer. Therefore, no connections exist between the outputs of a layer and the inputs of either the same layer or previous layers. In this topology, the inputs of each neuron are the weighted sum of the outputs from the previous layer. There are weighted connections between the outputs of each layer and the inputs of the next layer. If the weight of a branch is assigned a zero, it is equivalent to no connection between correspondence nodes. The inputs are connected to each neuron in hidden layer via their correspondence weights. Outputs of the last layer are considered the outputs of the network. Selecting the best number of hidden neurons involves experimentation. The forward selection method involves adding hidden neurons until network performance starts deteriorating. A neural network is required to go through training before it is actually being applied. Training involves feeding the network with data so that it would be able to learn the knowledge among inputs through its learning rule. Back propagation algorithm is used in supervised learning of the network. The main idea of the back propagation algorithm is to minimize the error, which is the difference between the expected value and the output of the model. Weights between neurons are adjusted until the error reaches an acceptable value. In order to train the network successfully, the output of the network is made to approach the desired output by continually reducing the error between the network's output and the desired output. This is achieved by adjusting the weights between layers by calculating the approximation error and back propagating this error from the final layer to the first layer. The weights are then adjusted in such a way to reduce the approximation error. The approximation error is minimized using the gradient descent optimization technique (Rojas 1996).

Faraway and Chatfield (1998) compared FFNN models with a SARIMA model on their accuracy for forecasting airline data in which the FFNN model also reduces the mean square errors (MSEs) of out-of-sample

prediction. Rao(2011), Naveen Kumar Boiroju (2012) and Zhang et.al. (1998) provided a comprehensive review of the current status of research in this area.

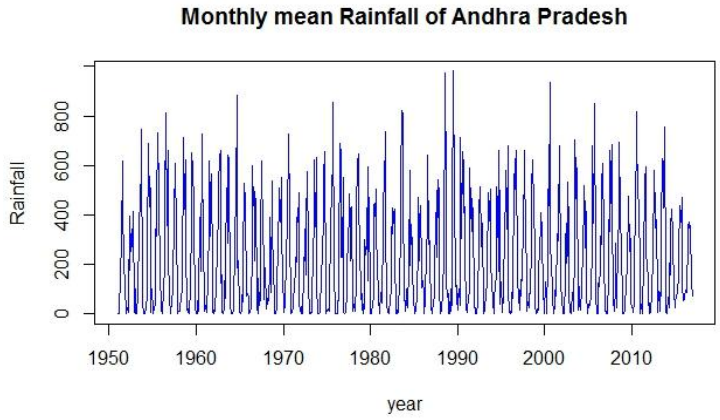
1. Rainfall Forecasting in Andhra Pradesh:

The data has been taken to predict the rain fall ofAndhra Pradesh area of Andhra Pradesh from the year 1951 to 2016. The data has monthly rainfall for each year. In this section, we have to check forecasting model to this data using one of statistical tool R software. In R software majorly we need packages for forecasting model. Using these packages is predicting the model for the Andhra Pradeshdata. The packages are '**ggplot2'**, '**forecast'** and '**tseries'**. Install the above mentioned packages using *install.packages()* function and call that packages using library function as below:

```
R> x=read.csv(file.choose(), header = T)
R> AP=ts (x [, 3], start=c (1951, 1), end = c (2016, 12), frequency = 12)
R>View(AP)
R>summary(AP)#summery details of the rainfall
Min.  1st Qu. Median  Mean 3rd Qu. Max.
  0.0   32.2   135.9   220.4   370.2   981.0
```

The minimum and maximum rainfall is 0.00 and 981.0, first and third quartiles are 32.2and 370.2. it shows the data has much scatter. The deviation is 169. Mean and median rainfall of Andhra Pradesh is 220.4 an d 135.9 respectively.

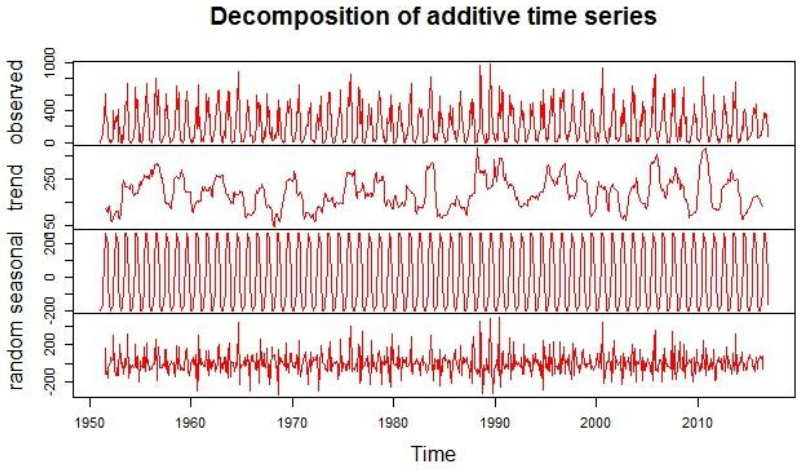
```
R>plot (AP, xlab='year', ylab = 'Rainfall', main="Monthly mean Rainfall of Andhra Pradesh", col="blue")
```




```
R>library(tseries)
R>library(forecast)
R>adf.test(AP, alternative= c("stationary", "explosive"))
Augmented Dickey-Fuller Test
data:  AP
Dickey-Fuller = -12.788, Lag order = 9, p-value = 0.01
alternative hypothesis: stationary
```

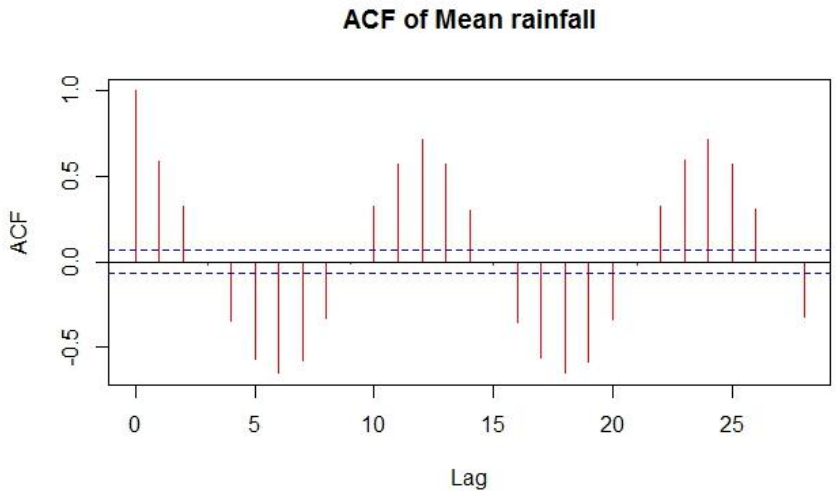
Based on the graph, we cannot identify whether the data is in stationary or not. To check the stationary of the data we have applied Augmented Dickey-Fuller test. The test is significant (p=0.01), so the data is stationary and we can observe in the graph of the data and its differences also.

```
R>Decom = decompose (AP)
R>plot (Decom, col='red')
```

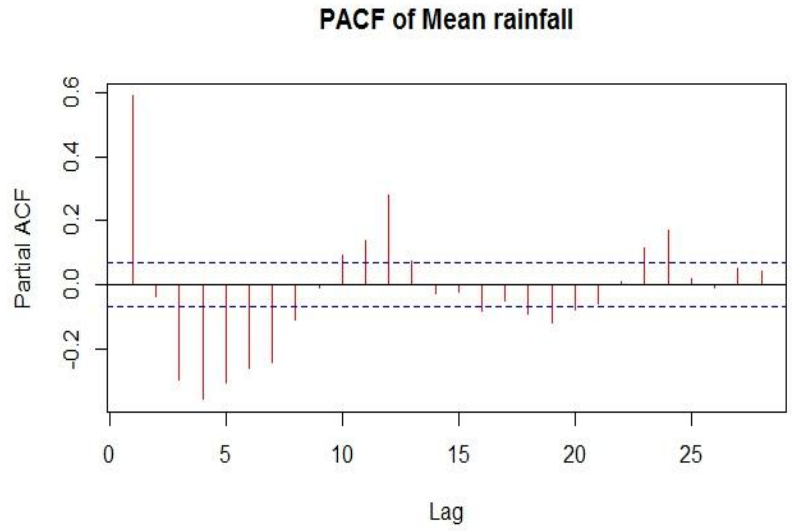


The above graph of decomposition, it has observed (which is drawn for original values), seasonal component (finding the seasonality), trend component and random. It shows the periodic seasonal pattern extracted out from the original data and the trend.

R>acf(ts(AP), main= 'ACF of Mean rainfall', col = "red")



>pacf(ts(AP), main= 'PACF of Mean rainfall', col = "red")



The ACF and PACF plots of the residuals from the ARIMA (1,0,0)(2,0,0)[12] model shows all correlations within the threshold limits indicating that the residuals are behaving like white noise.

R> y=auto.arima(AP, approximation = F, trace = F)
R>y
Series: AP
ARIMA(1,0,0)(2,0,0)[12] with non-zero mean
Coefficients:
ar1 sar1 sar2 mean
Estimates0.0717 0.4047 0.4089 215.497
Standard error 0.0350 0.0269 0.0330 25.759
sigma^2 estimated as 18949: log likelihood=-5028.2
AIC=10066.41 AICc=10066.48 BIC=10089.78

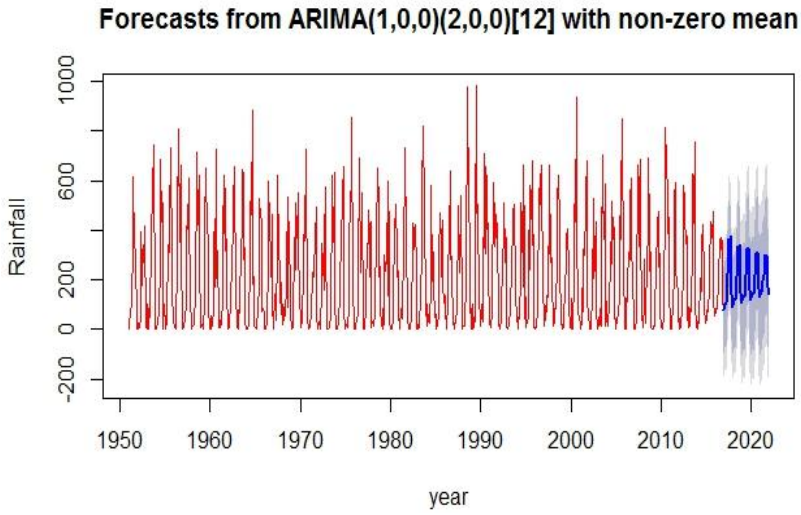
The fitted ARIMA model for the data is ARIMA (1,0,0)(2,0,0)[12] with non-zero mean. The predicted values for coast area rain fall details using ARIMA method of (1, 0, 0) and (2, 0, 0) is given and forecasts from the ARIMA(1,0,0)(2,0,0)[12] model are shown in the graph.

R> forecast(y, h=60)

Point	**Forecast**	**Lo 80**	**Hi 80**	**Lo 95**	**Hi 95**
Jan 2017	79.48285	-96.931	255.8976	-190.320	349.286
Feb 2017	101.93115	-74.936186	278.7985	-168.564097	372.426
Mar 2017	109.35396	-67.515695	286.2236	-161.144836	379.852

Apr 2017	109.89402	-66.975647	286.7637	-160.604795	380.392
May 2017	178.85831	1.988641	355.7280	-91.640506	449.357
Jun 2017	223.80622	46.936548	400.6759	-46.692599	494.305
Jul 2017	364.46937	187.599705 5	41.3390	93.970558	634.968
Aug 2017	348.31543	171.445757	525.1851	77.816610	618.814
Sep 2017	323.52112	146.651452	500.3908	53.022305	594.019
Oct 2017	378.09632	201.226647	554.9660	107.597500	648.595
Nov 2017	140.43343	-36.436239	317.3031	-130.065386	410.932
Dec 2017	91.99488	-84.874794	268.8645	-178.503941	362.493
Jan 2018	95.56073	-95.176623	286.2981	-196.146877	387.268
Feb 2018	111.47439	-79.331595	302.2804	-180.338185	403.287
Mar 2018	124.29171	-66.514628	315.0981	-167.521404	416.104
Apr 2018	119.64461	-71.161732	310.4510	-172.168509	411.457
May 2018	189.13925	-1.667093	379.9456	-102.673871	480.952
Jun 2018	214.19994	23.393591	405.0063	-77.613186	506.013
Jul 2018	334.66970	143.863357	525.4760	42.856579	626.482
Aug 2018	332.46594	141.659600	523.2723	40.652823	624.279
Sep 2018	303.90881	113.102468	494.7152	12.095690	595.721
Oct 2018	339.08087	148.274528	529.8872	47.267751	630.894
Nov 2018	148.48252	-42.323821	339.2889	-143.330598	440.295
Dec 2018	107.41206	-83.394286	298.2184	-184.401063	399.225
Jan 2019	111.34240	-104.560424	327.2452	-218.852475	441.537
Feb 2019	126.96171	-89.062510	342.9859	-203.418828	457.342
Mar 2019	135.18423	-80.840615	351.2091	-195.197262	465.565
Apr 2019	133.52426	-82.500592	349.5491	-196.857241	463.905
May 2019	189.84855	-26.176300	405.8734	-140.532949	520.230
Jun 2019	218.36953	2.344678	434.3944	-112.011971	548.750
Jul 2019	324.64094	108.616088	540.6658	-5.740561	655.022
Aug 2019	317.14399	101.119141	533.1688	-13.237508	647.525
Sep 2019	295.44833	79.423480	511.4732	-34.933168	625.829
Oct 2019	331.99804	115.973185	548.0229	1.616536	662.379
Nov 2019	157.68262	-58.342232	373.7075	-172.698881	488.061
Dec 2019	121.25484	-94.770010	337.2797	-209.126659	451.636
Jan 2020	124.30355	-102.805103	351.4122	-223.029163	471.636
Feb 2020	137.13185	-90.032350	364.2960	-210.285813	484.549
Mar 2020	145.70045	-81.464033	372.8649	-201.717648	493.118
Apr 2020	143.12851	-84.035975	370.2930	-204.289590	490.546
May 2020	194.33930	-32.825188	421.5038	-153.078804	541.757
Jun 2020	216.12924	-11.035242	443.2937	-131.288858	563.547
Jul 2020	308.39756	81.233070	535.5620	-39.020545	655.815
Aug 2020	304.46229	77.297804	531.6268	-42.955812	651.880
Sep 2020	284.00508	56.840598	511.1696	-63.413017	631.423
Oct 2020	313.17879	86.014305	540.3433	-34.239310	660.596
Nov 2020	164.69724	-62.467251	391.8617	-182.720866	512.113
Dec 2020	133.16113	-94.003354	360.3256	-214.256969	480.579
Jan 2021	136.00206	-101.606989	373.6111	-227.389621	499.393
Feb 2021	147.58041	-90.081110	385.2419	-215.891519	511.052
Mar 2021	154.41035	-83.251434	392.0721	-209.061986	517.882
Apr 2021	152.69070	-84.971092	390.3525	-210.781645	516.163
May 2021	196.44683	-41.214961	434.1086	-167.025513	559.912
Jun 2021	216.92741	-20.734378	454.5892	-146.544930	580.399
Jul 2021	297.72291	60.061116	535.3847	-65.749436	661.195
Aug 2021	293.06486	55.403065	530.7266	-70.407487	656.537
Sep 2021	275.91440	38.252613	513.5762	-87.557939	639.386
Oct 2021	302.66616	65.004369	540.3280	-60.806183	666.138
Nov 2021	171.29796	-66.363833	408.9597	-192.174385	534.770
Dec 2021	143.63994	-94.021854	381.3017	-219.832407	507.112

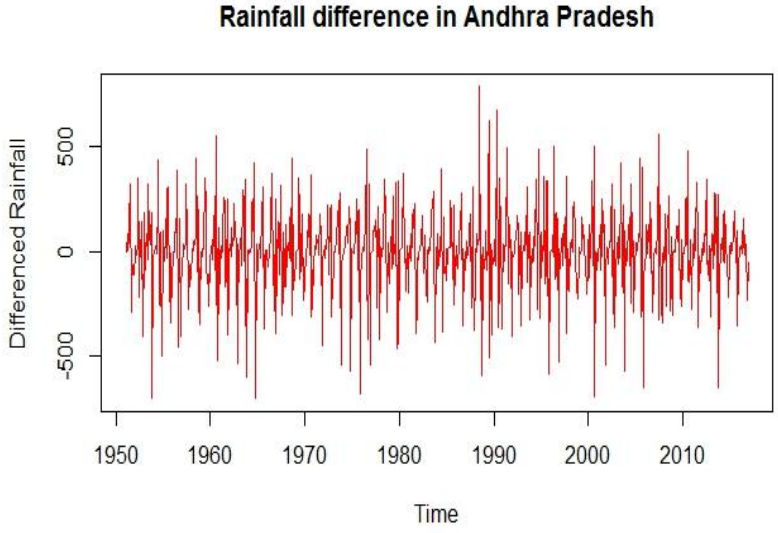
R>plot (forecast(y, h=60), xlab='year', ylab = 'Rainfall', col="red")



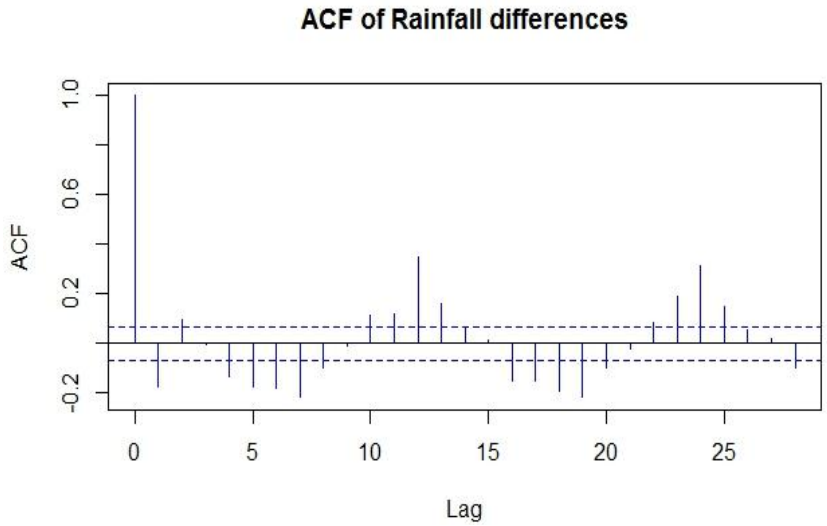
3.1 Forecasting for Seasonal Differences:

In this section, we have considered the rainfall data with differences. The same interpretation has been carried out for the below mentioned model.

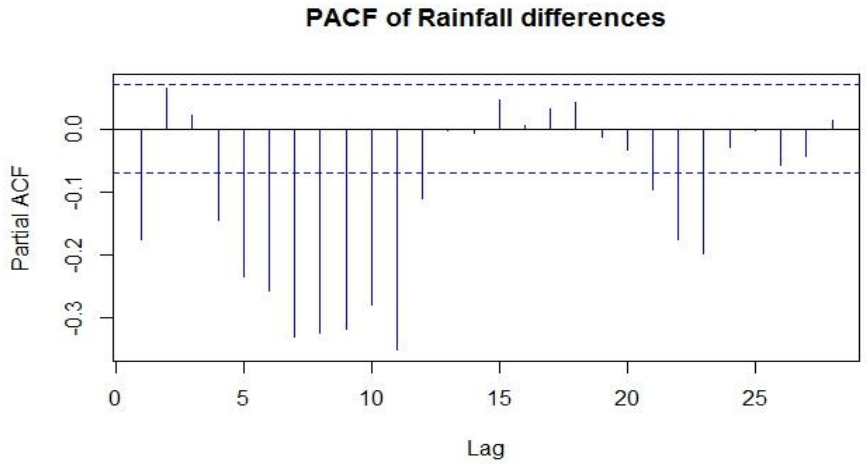
R>plot (diff (AP), main="Rainfall difference in Andhra Pradesh", ylab='Differenced Rainfall', col="red")



R>acf(ts(diff(AP)), main = 'ACF of Rainfall differences', col = "blue")



R>pacf(ts(diff(AP)), main = 'PACF of Rainfall differences', col = "blue")



R> y = auto.arima(diff(AP), approximation = F, trace = F)
R>y
Series: diff(AP)
ARIMA(3,0,0)(2,0,0)[12] with zero mean
Coefficients:

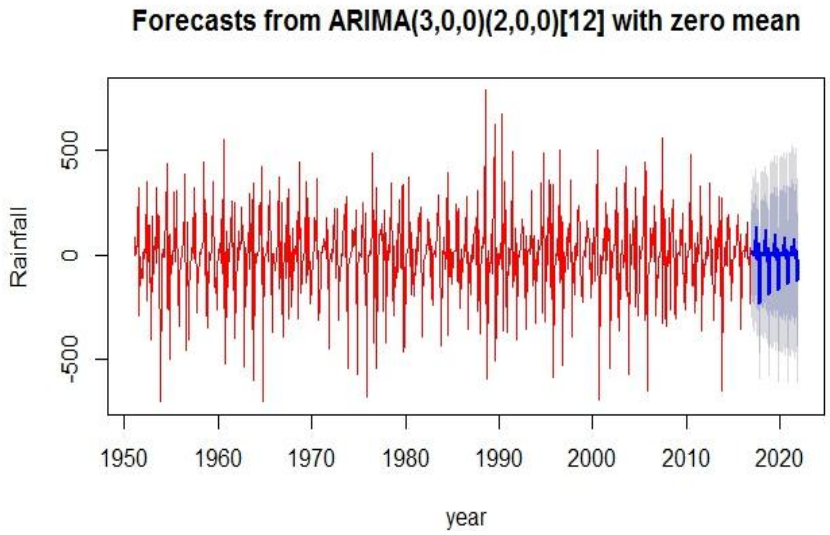
ar1	ar2	ar3	sar1	sar2
Estimates	-0.6703	-0.3878	-0.1804	0.4140	0.3891
Standard error	0.0375	0.0430	0.0362	0.0328	0.0330

sigma^2 estimated as 24043: log likelihood=-5115.44
AIC=10242.88 AICc=10242.99 BIC=10270.92
R> forecast(y, h=60)

Point	**Forecast**	**Lo 80**	**Hi 80**	**Lo 95**	**Hi 95**
Jan 2017	6.9803553	-191.7346	205.695322	-296.9280	310.888
Feb 2017	21.5878978	-217.6390	260.814792	-344.2781	387.453
Mar 2017	7.6660553	-231.8725	247.204596	-358.6765	374.008
Apr 2017	0.1613994	-239.4985	239.821318	-366.3668	366.689
May 2017	68.6469904	-171.4319	308.725866	-298.5220	435.819
Jun 2017	43.2166482	-197.3097	283.742949	-324.6366	411.069
Jul 2017	138.3278071	-102.2166	378.872263	-229.5532	506.208
Aug 2017	-15.0630551	-255.6103	225.484144	-382.9483	352.821
Sep 2017	-24.8926593	-265.4410	215.655644	-392.7795	342.994
Oct 2017	52.8509127	-187.7019	293.403691	-315.0428	420.744
Nov 2017	-232.814973	-473.3682	7.738357	-600.7094	135.079
Dec 2017	-47.606542	-288.1599	192.946724	-415.5011	320.288
Jan 2018	3.9405839	-250.2893	258.170427	-384.8704	392.751
Feb 2018	15.433744	-244.7347	275.602116	-382.4595	413.320
Mar 2018	12.511657	-247.7072	272.730546	-385.4588	410.481
Apr 2018	-4.563277	-264.8014	255.674816	-402.5631	393.436
May 2018	67.987428	-192.3167	328.291547	-330.1134	466.082
Jun 2018	24.426597	-235.9486	284.801646	-373.7829	422.635
Jul 2018	117.726478	-142.6516	378.104518	-280.4874	515.940
Aug 2018	-2.1111048	-262.4896	258.267368	-400.3256	396.103
Sep 2018	-27.930296	-288.3089	232.448356	-426.1451	370.284
Oct 2018	34.328920	-226.0504	294.708276	-363.8869	432.544
Nov 2018	-186.216436	-446.5959	74.163007	-584.4324	211.996
Dec 2018	-40.134342	-300.5138	220.245100	-438.3503	358.086
Jan 2019	4.347227	-278.8503	287.544791	-428.7661	437.460
Feb 2019	14.788537	-278.0968	307.673867	-433.1409	462.710
Mar 2019	8.162089	-284.8039	301.128077	-439.8907	456.219
Apr 2019	-1.826203	-294.8234	291.170995	-449.9268	446.273
May 2019	54.853788	-238.2511	347.958628	-393.4114	503.190
Jun 2019	26.926715	-266.2934	320.146877	-421.5148	475.363
Jul 2019	102.55579	-190.6691	395.780619	-345.8930	551.005
Aug 2019	-6.734795	-299.9604	286.490772	-455.1846	441.710
Sep 2019	-21.247449	-314.4733	271.978403	-469.6977	427.208

Oct 2019	34.774487	-258.4525	328.001490	-413.6775	483.225
Nov 2019	-167.671759	-460.8988	125.555428	-616.1239	280.705
Dec 2019	-35.137166	-328.3643	258.089970	-483.5894	413.310
Jan 2020	3.332808	-300.1195	306.785096	-460.7574	467.423
Feb 2020	12.126933	-295.8215	320.075313	-458.8395	483.093
Mar 2020	8.246928	-299.7395	316.233333	-462.7776	479.275
Apr 2020	-2.531496	-310.5325	305.469529	-473.5784	468.514
May 2020	49.160381	-258.8910	357.211751	-421.9635	520.243
Jun 2020	20.650637	-287.4548	328.756037	-450.5559	491.852
Jul 2020	88.259952	-219.8477	396.367559	-382.9500	559.469
Aug 2020	-3.609329	-311.7173	304.498607	-474.8198	467.611
Sep 2020	-19.662943	-327.7710	288.445126	-490.8736	451.577
Oct 2020	27.752209	-280.3564	335.860819	-443.4592	498.937
Nov 2020	-141.863962	-449.9726	166.244710	-613.0755	329.376
Dec 2020	-30.161176	-338.2699	277.947498	-501.3727	441.054
Jan 2021	3.071102	-314.1930	320.335189	-482.1424	488.287
Feb 2021	10.774109	-310.5267	332.074914	-480.6131	502.163
Mar 2021	6.589664	-314.7452	327.924524	-484.8496	498.029
Apr 2021	-1.758491	-323.1065	319.589509	-493.2178	489.709
May 2021	41.693381	-279.6999	363.086680	-449.8353	533.220
Jun 2021	19.025486	-302.4165	340.467279	-472.5775	510.623
Jul 2021	76.439371	-245.0045	397.883169	-415.1666	568.043
Aug 2021	-4.114552	-325.5587	317.329598	-495.7210	487.419
Sep 2021	-16.40679	-337.8511	305.037471	-508.0134	475.198
Oct 2021	25.018649	-296.4261	346.463404	-466.5887	516.620
Nov 2021	-123.96507	-445.4099	197.479732	-615.5725	367.643
Dec 2021	-26.156976	-347.6018	295.287837	-517.7644	465.454

R>plot(forecast(y, h=60), xlab='year', ylab = 'Rainfall', col="red")

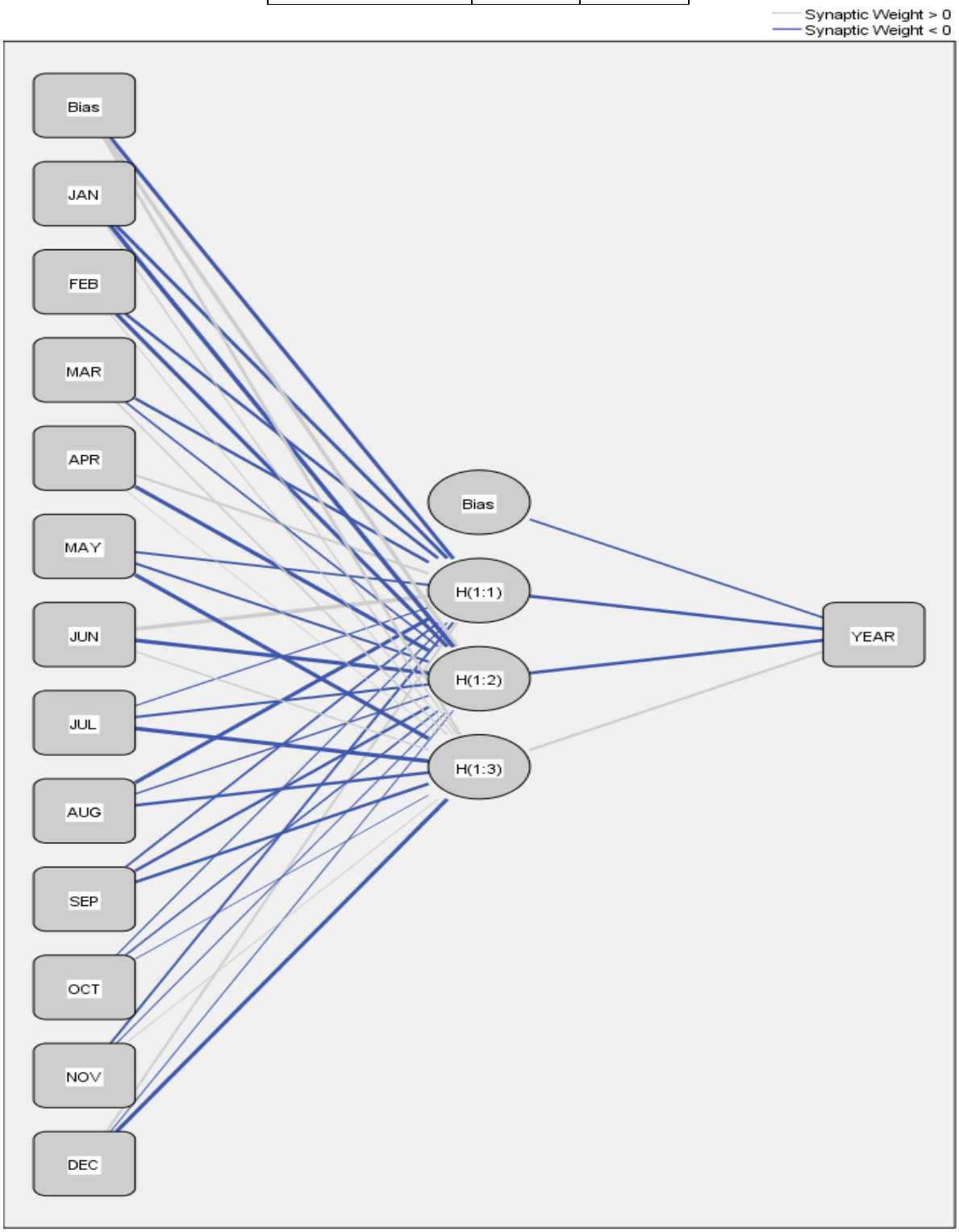


3.2 Building FFNN Model:

The data set is partitioned into two sets namely training set and testing set. For model building 71.2% of the data is taken as training set and 28.8% of the data is taken as testing set. The feed forward neural network (FFNN) consists of input layer, hidden layer and output layer. Input layer consists of 12 units representing the month (numbers from 1 to 12), Z_{t-1} and Z_{t-12} values. Output layer consists of only one neuron and represents the forecast value ($\hat{Z}_t$) of the series. Number of hidden neurons in the hidden layer is determined using forward selection method. The optimum number of hidden neurons is four. Hyperbolic tangent function is used as an activation function and scaled conjugate gradient algorithm is used to train the network. The network is trained until the number of epochs is equivalent to 10,000. SPSS software is used to train the network. With the above specifications the following synaptic weights are obtained.

Case Processing Summary			
		N	Percent
Sample	Training	47	71.2%

	Testing	19	28.8%
Valid		66	100.0%
Total		66	



Hidden layer activation function: Hyperbolic tangent

Output layer activation function: Identity

Model Summary		
Training	Sum of Squares Error	13.417
	Relative Error	0.583
Testing	Sum of Squares Error	4.639
	Relative Error	0.702

Parameter Estimates					
Predictor		Predicted			
		Hidden Layer 1			Output Layer
		H(1:1)	H(1:2)	H(1:3)	YEAR
Input Layer	(Bias)	-0.598	0.651	0.405	
	JAN	-0.556	-0.918	0.061	
	FEB	-0.518	-0.725	0.028	
	MAR	-0.510	-0.241	0.154	
	APR	0.350	-0.777	0.004	
	MAY	-0.341	-0.352	-0.814	
	JUN	1.065	-1.037	0.143	
	JUL	-0.108	-0.383	-1.047	
	AUG	-0.740	-0.188	-0.530	
	SEP	-0.275	-0.412	-0.526	
	OCT	-0.103	-0.198	-0.023	
	NOV	-0.280	-0.056	0.024	
	DEC	0.250	-0.035	-0.693	
Hidden Layer 1	(Bias)				-0.273
	H(1:1)				-0.535
	H(1:2)				-0.572
	H(1:3)				0.371

The below model is used to forecast the values of monthly mean rainfall of Andhra Pradesh.FFNN forecasting model can be constructed using above synoptic weights as follows

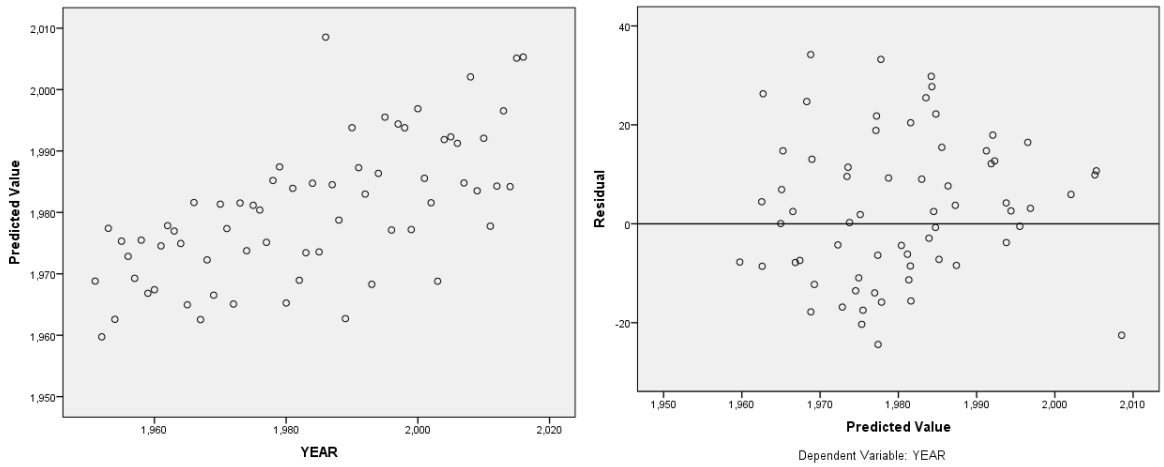
$$\hat{Z}_s = -0.273 - 0.535H(1:1) - 0.572H(1:2) + 0.371H(1:3)$$

Where

$$H(1:1) = Tanh\begin{pmatrix} -0.598 - 0.556I(JAN) - 0.518I(FEB) - 0.510I(MAR) + 0.350I(APR) \\ -0.341I(MAY) + 1.065I(JUN) - 0.108I(JUL) - 0.740I(AUG) \\ -0.275I(SEP) - 0.103I(OCT) - 0.280I(NOV) + 0.250I(DEC) \end{pmatrix}$$

$$H(1:2) = Tanh\begin{pmatrix} 0.651 - 0.918I(JAN) - 0.725I(FEB) - 0.241I(MAR) - 0.777I(APR) \\ -0.352I(MAY) - 1.037I(JUN) - 0.383I(JUL) - 0.188I(AUG) \\ -0.412I(SEP) - 0.198I(OCT) - 0.056I(NOV) - 0.035I(DEC) \end{pmatrix}$$

$$H(1:2) = Tanh\begin{pmatrix} 0.405 + 0.061I(JAN) + 0.028I(FEB) - 0.154I(MAR) + 0.004I(APR) \\ -0.814I(MAY) + 0.143I(JUN) - 1.047I(JUL) - 0.530I(AUG) \\ -0.526I(SEP) - 0.023I(OCT) + 0.024I(NOV) - 0.693I(DEC) \end{pmatrix}$$



Conclusion:

The data has been fitted to the ARIMA (1, 0, 0) (2, 0, 0)[12] model for rainfall of Andhra Pradesh. Augmented Dickey-Fuller Test has been tested for stationarity of the data. Basing on the p-value (p=0.01), the data has been stationary and we have applied for auto ARIMA to find and check the best model using R. We made the interpretation basing on the AIC and BIC values of the model. The lowest AIC and BIC will give us

the best fit of the forecast model. Based on auto ARIMA, the best fitted model has been found ARIMA (1, 0, 0) (2, 0, 0) [12], which has the seasonality. The prediction values and its graphs have been shown. Using neural network, it found that there are three hidden layers, from which it has been given the hyperbolic tangent form. Predicted and residual plot has been shown in the graph.

References:

1. Afzali, M., Afzali, A. & Zahedi, G. (2011). Ambient Air Temperature Forecasting Using Artificial Neural Network Approach, International Conference on Environmental and Computer Science, IPCBEE Vol.19, IACSIT Press, Singapore.
2. Alfaro, E. (2004). A Method for Prediction of California Summer Air Surface Temperature, Eos, Vol. 85, No. 51, 21 December 2004.
3. Anisimov O.A., (2001). Predicting Patterns of Near-Surface Air Temperature Using Empirical Data, Climatic Change, Vol. 50, No. 3, 297-315.
4. Box, G. E. P., Jenkins, G. M. & Reinsel, G. C. (1994). Time Series Analysis Forecasting and Control, 3rd ed., Englewood Cliffs, N.J. Prentice Hall.
5. Brunetti, M., Buffoni, L., Maugeri, M. & Nanni, T., (2000). Trends of minimum and maximum daily temperatures in Italy from 1865 to 1996. Theor. Appl. Climatol., 66, 49–60.
6. FAN Ke, (2009). Predicting Winter Surface Air Temperature in Northeast China, Atmospheric and Oceanic Science Letters, Vol. 2, No. 1, 14−17.
7. Faraway, J. & Chatfield, C. (1998). Time series forecasting with neural networks: a comparative study using the airline data, Journal of the Royal Statistical Society, Series C, Vol. 47, 2, 231-250.
8. Hejase, H.A.N. &Assi, A.H. (2012). Time-Series Regression Model for Prediction of Mean Daily Global Solar Radiation in Al-Ain, UAE, ISRN Renewable Energy, Vol. 2012, Article ID 412471, 11 pages.
9. J. C. Ramesh Reddy, T. Ganesh, M. Venkateswaran, PRS Reddy (2017), Forecasting of Monthly Mean Rainfall in Coastal Andhra, International Journal of Statistics and Applications 2017, 7(4): 197-204. DOI: 10.5923/j.statistics.20170704.01
10. Kulkarni M.A., Patil, S., Rama, G.V. & Sen, P.N. (2008). Wind speed prediction using statistical regression and neural network, J. Earth Syst. Sci. 117, No. 4, 457–463.
11. Lee, J.H., Sohn, K. (2007). Prediction of monthly mean surface air temperature in a region of China, Advances in Atmospheric Sciences, Vol. 24, 3, 503-508.
12. Lee, J.H., Sohn, K. (2007). Prediction of monthly mean surface air temperature in a region of China, Advances in Atmospheric Sciences, Vol. 24, 3, 503-508.
13. Shrivastava, G., Karmakar, S., Kowar, M. K., Guhathakurta, P. (2012). Application of Artificial Neural Networks in Weather Forecasting: A Comprehensive Literature Review, International Journal of Computer Applications, Vol. 51, No.18, August 2012.
14. Stein, M. & Lloret, J. (2001). Forecasting of Air and Water Temperatures for Fishery Purposes with Selected Examples from Northwest Atlantic, J. Northw. Atl. Fish. Sci., Vol. 29, 23-30.
15. Tasadduq. I., Rehman, S., Bubshait, K. (2005). Application of neural networks for the prediction of hourly mean surface temperatures in Saudi Arabia. Renewable Energy, 25, 545-554.
16. Zhang, G., Patuwo, B. E. & Hu, M. Y. (1998). Forecasting with Artificial Neural Networks: The State of the Art, International Journal of Forecasting, 14, 35-62.