Data Mining based Neural Network Model for Rainfall Forecasting

P. Arumugam
Professor & Head
Department of Statistics
Manonmaniam Sundaranar University
Abishekapatti, Tirunelveli, Tamil Nadu 627012

R. Ezhilarasi
Research Scholar
Information Technology
Manonmaniam Sundaranar University
Abishekapatti, Tirunelveli, Tamil Nadu 627012

ABSTRACT
India is basically an agricultural country and the success or failure of the harvest and water scarcity in any year is always considered with the greatest concern. The average annual or seasonal rainfall at a place does not give sufficient information regarding its capacity to support crop production. Rainfall distribution pattern is the most important. The rainfall forecasting is scientifically and technologically challenging problem around the world in the last century. In this paper Neural Network model was developed for the rainfall forecast performance and the results were compared with Seasonal Auto regressive integrated moving average (SARIMA) model. The performance by (ANN) model and statistical time series model for prediction were examined using visualization technique and statistical test.

Keywords
Data mining, Neural networks, Time Series, SARIMA, BIC and RMSE.

1. INTRODUCTION
India is basically an agricultural country and the success or failure of the harvest and water scarcity in any year is always considered with the greatest concern. The average annual or seasonal rainfall at a place does not give sufficient information regarding its capacity to support crop production. Rainfall distribution pattern is the most important. The rainfall forecasting is scientifically and technologically challenging problem around the world in the last century. In this chapter a Neural Network model was developed for the rainfall forecast performance and the results were compared with Seasonal Auto regressive integrated moving average (SARIMA) model. The performance by (ANN) model and statistical time series model for prediction were examined using visualization technique and statistical test. Zhang et al.,(1998) Have discussed the back propagation neural network is a feed forward network and is the most widely applied neural network technique in time series forecasting. Chen et al., (2012) have discussed the novel forecasting model based on empirical mode decomposition and neural network model for tourism demand. Bhatnagar et al., (2012) have develop a prediction model for dengue fever using time series data over the past decade in Rajasthan. Senthamarai Kannan et al., (2013) have discussed the outlier detection methods in neural network and time series models. Narvekar and Fargose (2015) have discussed the daily weather forecasting using neural network. They have used various input parameters to forecast temperature, rainfall, humidity, cloud condition and weather of the day.

In the analysis of rainfall, however, the concerns of farmers go further, since they need to consider also how variable the rainfall is from year to year or for a given month and how frequently droughts of a certain level of severity are likely to recur. The reservoir of water from which crops draw their moisture supply through the soil is derived mainly in the form of rainfall, with relatively minor contributions in India from dew. The median is the middle value when a data series is ranked from highest to lowest. It therefore designates a statistically expected value, with as many years having been wetter than the median as there are years having had less rain than the median value. Mean values are frequently inflated by a few heavy and extreme events or outlier events, which may have occurred - a phenomenon which is especially prevalent in dry regions or in generally dry months. Under such rainfall regimes the mean is therefore not as representative of expected conditions as the median.

2. MODELING SEASONALITY
To model seasonality, the length of the series must exceed the length of the span of the seasonality. Incomplete spans of seasonality may add errors to the analysis. Enders writes that when seasonal variations predominate, much of the errors in the forecast may derive from this variation. Therefore, we should remove or model seasonality to whatever extent possible before forecasting.

Especially when a series is being used for forecasting, the seasonality, which contributes to error variance, should be removed. When that is done the series is called seasonally adjusted. If the series is not seasonally adjusted first, seasonality can be modeled in the Box-Jenkins approach by employing seasonal components alone or mixing these seasonal with regular nonseasonal components to construct multiplicative Box-Jenkins model. Within Box-Jenkins model, seasonality may refer to any repetition of pattern of activity. Seasonal variation has an order to it. By convention, the order of seasonality is the number of seasons in an annual period. Quarterly seasonal peaks in data indicate a seasonal order of 4. If the data are measured daily but monthly seasonality is present, then the order of the seasonality is 12. In order to approach the basics of seasonal modeling, we turn first to the subject of seasonal stationarity and its complement, seasonal nonstationarity.

2.1 Identifying the Seasonal Model
Seasonality can be assessed from an autocorrelation plot. Box and Jenkins recommend the differencing approach to achieve stationarity. At the model identification stage, our goal is to detect seasonality, if it exists, and to identify the order for the seasonal autoregressive and seasonal moving average terms. For many series, the period is known and a single seasonality term is sufficient. First step is stationarity and seasonality have been addressed, the next step is to identify the order of p and q. For higher-order autoregressive processes, the sample autocorrelation needs to be supplemented with a partial autocorrelation plot. The partial autocorrelation of an AR (p)

process becomes zero at lag p+1 and greater, so we examine the sample partial autocorrelation function to see if there is evidence of a departure from zero. This is usually determined by placing a 95% confidence interval on the sample partial autocorrelation plot .The confidence band is approximately, with N denoting the sample size.

The autocorrelation function of a MA (q) process becomes zero at lag q+1 and greater, so we examine the sample autocorrelation function to see where it essentially becomes zero. We do this by placing the 95% confidence interval for the sample autocorrelation function on the sample autocorrelation plot. Most software that can generate the autocorrelation plot can also generate this confidence interval..

2.2 Seasonal Auto Regressive Integrated Moving Average Model

A complexity to add to ARIMA model is seasonality. In the same way that consecutive data points might exhibit AR, MA, mixed ARMA or mixed ARIMA properties, so data separated by a whole season may exhibit the same properties. The ARIMA models can be extended to handle seasonal components of a data series. Seasonal ARIMA (SARIMA) is an extension of the method to a series in which a pattern repeats seasonally over time and is represented as SARIMA (p, d, q) (P, D, Q)s. Analogous to the simple ARIMA parameters, these are: Seasonal autoregressive (P), seasonal differencing (D), and seasonal moving average parameters (M); s- defines the number of time periods until the pattern repeats again approximately stationary. A stationary time series is one whose statistical properties such as mean and variance are constant over time. Seasonality usually causes the series to be non-stationary because the average values at some particular times within the seasonal span may be different than the average values at other times.

2.3 SARIMA Model

The general SARIMA (p, d, q) (P, D, Q)$_s$ model is written as

$$\phi(B)\Phi(B^s)(1-B)^d(1-B^s)^D Y_t = \theta(B)\Theta(B^s) \quad \ldots (1)$$

where ϕ, Φ, θ and Θ are the parameters whose values have to be estimated. The simplest general ARIMA (1, 1, 1) (1, 1, 1)$_4$ is

$$(1-\varphi_1 B)(1-\Phi_1 B^4)(1-B)(1-B^4)Y_t = (1-\theta_1 B)(1-\theta_1 B^4)e_t. \quad \ldots (2)$$

where

$(1-\varphi_1 B)$ - Non – seasonal AR (1)
$(1-\Phi_1 B^4)$ - Seasonal AR (1)
$(1-B)$ - Non– seasonal difference
$(1-B^4)$ - Seasonal difference
$(1-\theta_1 B)$ - Non – seasonal MA (1)
$(1-\theta_1 B^4)$ - Seasonal MA (1)

The seasonal part of an AR or MA model will be seen in the seasonal lags of the PACF and ACF. For example, the seasonal MA model ARIMA (0, 1, 1) (0, 1, 1)12 will show a spike at lag 12 in the ACF but no other significant spikes. The PACF will show the exponential decay in the seasonal lags; that is at lags 12, 24, 36 . . . . Similarly an ARIMA (0, 1, 1) (0,1,1)12 will show exponential decay in the seasonal lags of the ACF, and a single significant spike at lag 12 in the PACF

2.4 Feed Forward Neural Network

In feed forward neural network, the architecture consists of not only input and output layer but also one or more intermediary layers called hidden layers. The computation units of the hidden layer are called as hidden neurons. The function of hidden neurons is to intervene between the external input and the network output in some useful manner. By adding one or more hidden layers, the network is enabled to extract higher order statistics. The ability of hidden neurons to extract higher order statistics is particularly valuable when the size of the input layer is large. The source nodes in the input layer of the network supply respective elements of the activation pattern which constitute the input signals applied to the neurons in the second layer. The output signals of the second layer are used as inputs to the third layer and so on for the rest of the network. Typically, the neurons in each layer of the network have their inputs as the output signals of the preceding layer only. The set of output signals of the neurons in the output layer of the network constitutes the overall response of the network to the activation pattern supplied by the source nodes in the input layer.

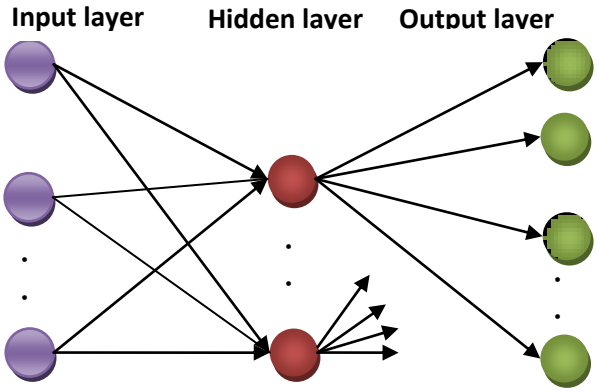

Figure 1: Multi layered feed forward Network

Multi layered feed forward network with m-source nodes, h1 neurons in the first hidden layer, h2-neurons in the second hidden layer and n-neurons in the output layer can be referred as m-h1-h2-n network. A feed forward neural network (FFNN) is the most popular neural networks for forecasting of time series data. The input nodes are the previous lagged observations, while the output provides the forecast for the future values. Hidden nodes with appropriate non-linear transfer functions are used to process the information received by the input nodes.

The model of FFNN can be written as

$$Z_t = \beta_0 + \sum_{j=1}^{q} \beta_j f\left(\sum_{i=1}^{p} \gamma_{ij} Z_{t-i} + \gamma_{0j}\right) + \varepsilon_t \quad \ldots (3)$$

where
p is the number of input nodes
q is the number of hidden nodes
f is the hyperbolic tangent function.
$\{\beta_j, j = 0,1,\ldots q\}$ is a vector of weights from hidden nodes to output nodes
$\{\gamma_{ij}, i = 0,1\ldots p;\ j = 1,2,\ldots q\}$ are weighs from the input nodes hidden nodes.

Multi layered network can be viewed as cascading of groups of single layer networks. The level of complexity in computing can be seen by the fact that many single layer networks are combined into a multilayer network.

3. RESULTS AND DISCUSSIONS

Table 1: Model Statistics

Stationary R-squared	R-squared	RMSE	Normalized BIC
0.561	0.087	123.437	9.877

Among the statistical models, SARIMA (0,0,1) (0,1,1) 12 was selected as the best model, with the lowest normalized BIC of 9.877 and a RMSE of 123.437. The model explained 56.1% of the variance of the series.

Table 2: SARIMA Model Parameters

Rainfall		Estimate	SE	Sig.
Constant		-.855	1.472	.564
Difference		1		
MA	Lag 1	.995	1.912	.605
Seasonal Difference		1		
MA, Seasonal	Lag 1	.596	.291	.047

From the above table (2) it is observed that all the parameters are significant at 5% level. So the fitted model for the fitted for rainfall in Tamil nadu is Seasonal ARIMA (0, 1, 1)(0,1,1)12.

The model equation can be represented as follows

$$(1-B)\left(1-B^{12}\right)Y_t = (1-\theta B)\left(1-\Theta B^{12}\right)e_t \quad \ldots (4)$$

Substituting the value of the parameters we get

$$(1-B)\left(1-B^{12}\right)Y_t = (1-0.995B)\left(1-0.596B^{12}\right)e_t \quad \ldots (5)$$

The adequacy of the model is checked using the ACF and PACF of the residuals of various orders of the selected model

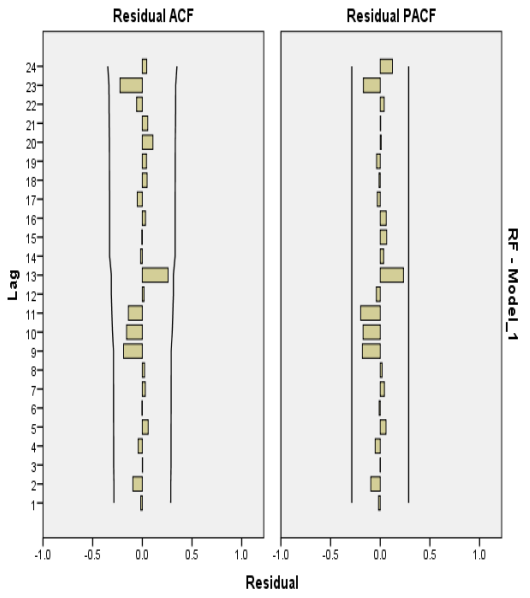

Figure 2: Residual of ACF and PACF

3.1 FFNN- Rescaling and Data Partition

In the present study, we use adjusted normalized method to rescale the variables. The adjusted normalized values fall between -1 to +1. The given data is partitioned into three samples viz. Training, testing, and holding samples. We have considered the following partitions table (3) of the data for forecasting of an optional neural network model.

Table 3: Data Partition

Partition	Partition I	Partition II	Partition III	Partition IV
Training (%)	77.4	72.6	65.5	60.7
Testing (%)	14.3	27.4	21.4	19.0
Holding (%)	8.3	0	13.1	20.2
Total (%)	100	100	100	100

3.2 Structure of Network

The model is a three layer feed forward neural network and it consists of an input, one hidden layer, and one output layer. The total number of input neurons needed in this model is two, each representing the values of production and year. In this model, only one output is needed and it indicates that the forecasts of production of rice. The number of hidden neurons are taken as the thumb rules given in the literature. The following results are obtained for each partition set.

Table 4: Layer Information

Predictor		Predicted		
		Hidden Layer 1		Output Layer
		H(1:1)	H(1:2)	RF
Input Layer	(Bias)	-.135	.441	
	Month	.538	-.145	
Hidden Layer 1	(Bias)			-.583
	H(1:1)			.738
	H(1:2)			.314

The forecasting model is given by

$$\hat{Z}_t = I(-0.583 + 0.738h_1 + 0.314h_2) \quad \ldots (6)$$

where

$$h_1 = \tanh\left(-0.135 + 0.538\hat{Z}_{t-1}\right) \text{ and}$$

$$h_2 = \tanh\left(0.441 - 0.145\hat{Z}_{t-1}\right)$$

where

I(.) is the identity function and

$\hat{Z}_{t-1}$ is an adjusted normalized lag variable

Table 5: Forested values of Rainfall

Month	SARIMA Forecast	FFNN Forecast
Jan-17	13.44	14.9
Feb-17	20.74	21.4
Mar-17	24.7	30.2
Apr-17	29.93	41.9
May-17	4.47	57
Jun-17	23.19	75.8

Jul-17	35.9	98
Aug-17	125.45	122.5
Sep-17	112.47	147.8
Oct-17	235.61	172.2
Nov-17	206.78	194.1
Dec-17	109.03	212.5
Jan-18	47.68	14.9
Feb-18	55.84	21.4
Mar-18	60.66	30.2
Apr-18	66.75	41.9
May-18	42.14	57
Jun-18	15.34	75.8
Jul-18	3.48	98
Aug-18	85.21	122.5
Sep-18	71.38	147.8
Oct-18	193.66	172.2
Nov-18	163.98	194.1
Dec-18	65.38	212.5

Table 6: Accuracy of forecast

Model	SARIMA	FFNN
RMSE	136.85	119.47

4. CONCLUSION

Seasonal autoregressive integrated moving average (SARIMA) model is used to discover the pattern and predict the future values to the data of rain fall in Tamil Nadu. After testing with various models SARIMA (0, 1, 1) $(0, 1, 1)_{12}$ was fitted and found to be an appropriate forecasting model for this particular problem. The forecasting equation is

$$(1-B)\left(1-B^{12}\right)Y_t = (1-0.995B)\left(1-0.596B^{12}\right)e_t \quad \ldots (7)$$

The FFNN forecasting model is given by

$$\hat{Z}_t = I\left(-0.583+0.738h_1+0.314h_2\right)$$

where $h_1 = \tanh\left(-0.135+0.538\hat{Z}_{t-1}\right)$ and $h_2 = \tanh\left(0.441-0.145\hat{Z}_{t-1}\right)$

The FFNN model is generally better than the SARIMA model in forecasting the rain fall data. It is suggested that future rainfall prediction of any area may be estimated for the benefits of meteorologists when applying the above model.

5. REFERENCES

[1] Bhatnagar S, Lal V, Gupta SD, Gupta OP (2012): Forecasting incidence of dengue in Rajasthan, using time series analyses. Indian J Public Health Vol.2, No.56, pp. 281-285.

[2] Box, G.E.P., and G.M. Jenkins (1976). Time Series Analysis: Forecasting and Control, Second Edition, Holden Day.

[3] Cadenas. E., and W. Rivera. (2010). Wind Speed Forecasting in Three Different Regions of Mexico, Using a Hybrid ARIMA-ANN Model, Renewable Energy, 35, 2732-2738.

[4] Chen, C., Lai, M., and Yeh, C. (2012). Forecasting tourism demand based on empirical mode decomposition and neural network. Knowledge Based Systems, Vol 26, pp. 281-287.

[5] Esling, P., and C. Agon (2012): Time-Series Data Mining. ACM Computing Surveys, Vol. 45, No.1, pp.1-34.

[6] Fausett, L (1994): Fundamentals of Neural Networks, Prentice Hall, USA.

[7] Freeman, J. A., and D.M. Skapura (1992): Neural Networks Algorithms, Applications and Programming Techniques. Addison-Wesley Publishing Company.

[8] Hansen, J.V., and R.D. Nelson (2002): Data Mining of Time Series Using Stacked Generalizers. Neurocomputing, Vol. 43, pp.173-184.

[9] Lee, T. S., and C. C. Chiu (2002). Neural Network Forecasting of an Opening Cash Price Index. International Journal of Systems Science, 33(3), 229-237.

[10] Narvekar M. and Fargose P. (2015): Daily weather forecasting using artificial neural network International Journal of Computer Applications 121 9-13.

[11] Senthamarai Kannan, K., Deneshkumar, V, and S Arumugam (2013): A Comparative Study on FFNN and ARIMA Model in the Presence of Outliers. International Journal of Computer Applications, Vol 76, No. (17), pp.12-18.

[12] Zhang. G., Patuwo B.E., and Hu. M.Y (1998): Forecasting with artificial neural network: the state of the art, international journal of foresting, Vol 14, PP.35 – 62.