Time-series prediction of rainfall in rural India with SVM and comparison with MLP

Animesh Anant Sharma
Electrical Engineering Department
Indian Institute of technology Roorkee
Email: anantanimesh36@gmail.com
Enroll. No.: 13115021

Ankita Puwar
Electrical Engineering Department
Indian Institute of technology Roorkee
Email: ankitapuwar@gmail.com
Enroll. No.: 13112011

Abstract—In this course project report we have compared various rainfall prediction models for the city of Nagpur in state of Maharashtra. The rainfall prediction can be useful for the field of agriculture. The time series prediction of rainfall was done using SVM and MLP. The conventional models based on various climatic features were also constructed using SVM and MLP. The Support Vector Regression has been implemented using LIBSVM library in MATLAB. Among these four models the conventional model constructed using SVM appears to be the best solution with a mean absolute error of 13.66.

Keywords—Multilayer Perceptron(MLP), Support Vector Regression(SVR), Time series prediction, Climatic features

I. Main Objectives

- Time series prediction of rainfall using SVM and MLP.
- Comparison with conventional model based on climatic features built using SVM and MLP.

II. Status and Other Details

- Completed
- Percentage Contribution of Members
  - Animesh Anant Sharma: 50%
  - Ankita Puwar : 50%
- Total time spent on the project: 18 Days

III. Major Stumbling Blocks

- Problems in feature selection for time series model.
- Extraction of values from the website due to lack of API.

IV. Introduction

Rainfall is considered to be one of the most important meteorological factor in the second and third world countries; its process of formation and prediction involve a rather complex physics which is not clearly understood. In tropical and sub-tropical countries, this variable is highly important for productive sectors, such as agriculture and water resource management, and for this reason, efforts have been focused on improving rainfall forecast [1]. An understanding of the temporal variability of rainfall can be useful in the more efficient planning of several human activities. For example, the distribution of rainfall during the year determines, to a large extent, the growth of plants and, hence, food production; therefore, the understanding of the temporal variability of rainfall can aid in the choice of the most suitable crop variety [2]. Due to this reason in many papers the rainfall predictions have been made on a monthly basis through conventional model as in [3] considering various climatic features and it can been compared with the time series model as in [2].

This project report compares the conventional model and time series model using both Support Vector Machines and multi-layer perceptron. Technique has been tested on the rainfall dataset of city of Nagpur [4]. Paavni Rattan ,Rishita Anubhai ,Amogh Vasekar [3] proposed that linear regression can be used for both the time series prediction and conventional model. Here Support Vector Machine and multi-layer perceptron have been used and the results have been compared. Mean absolute error and Root mean squared error have been calculated for all the four cases of time series SVM, time series MLP, conventional model with SVM and conventional model with MLP.

The rest of the report has been organised as: Section 5 gives description about dataset for both the time series model and conventional model, Section 6 explains about the concepts of SVM regression, Multi-layer perceptron MLP and time series modelling, Section 7 explains the results of all the four cases and finally the report has been concluded in section 8.

V. Dataset

A. Time series model

The dataset for this particular model was taken from the meteorological data provided on the website indiawaterportal.org [4]. The data has been provided from year 1901 to 2002. The values were extraction from the year 1902 to 2002 which gave readings for 101 years.

For the time series model only the precipitation values were considered. The features selected were the monthly mean values for rainfall in the past 12 months based on the intuition that the observed rainfall depends on the rainfall values recorded in the past one year. Also it can be argued that the particular time of the year also has a significant impact on the rainfall recorded. As such the values for that particular month in the past 10 years have also been considered. Since the last year value has already been considered in the 12 month period, there are 21 features in total. The target values were recorded from the year 1912. For example, if we have to predict the rainfall in the month of August of 2000, then period of August 1999 to July 2000 will be considered and the rainfall recorded in the month of August from 1990 to 1998 will be considered for input features.

B. Conventional model

The attributes that were used here mostly consist of climate related features. There is a strong anti-correlation between precipitation and daytime maximum temperature from daily to annual scales [5]. Therefore, maximum temperature has been included as one of the features. Similarly other different climate related features like cloud cover, diurnal temperature range and evapotranspiration have been considered. All the attributes used are given in Table 1.

Attribute	Attribute Information
Min_temp	Minimum temperature
Avg_temp	Average temperature
Max_temp	Maximum temperature
Cloud_cover	Cloud cover
Vapour_pressure	Vapour pressure
Wet_day_freq	Wet day frequency
Diurnal_temp_range	Diurnal temperature range
Ground_frost_freq	Ground frost frequency
Ref_crop_evap	Reference crop evapotranspiration
Pot_evap	Potential evapotranspiration

TABLE I. THE DESCRIPTION OF DATASET ATTRIBUTES

VI. INTELLIGENT TECHNIQUES USED

There are a number of techniques used for regression problems and the time series prediction including the Multilayer perceptron and the Support Vector regression techniques. We have used and compared the above mentioned methods for our dataset.

A. Multi-layer Perceptron (MLP)

A Multi-layer perceptron is a network of simple neurons called Perceptron. Each neuron computes the output which is some linear or usually a non-linear function of the linear combination of the input and the weights including the bias. The output for a neuron is given as follows:

$$y = \varphi\left(\sum_{i=1}^{n} w_i x_i + b\right) = \varphi(\mathbf{w}^T \mathbf{x} + b)$$

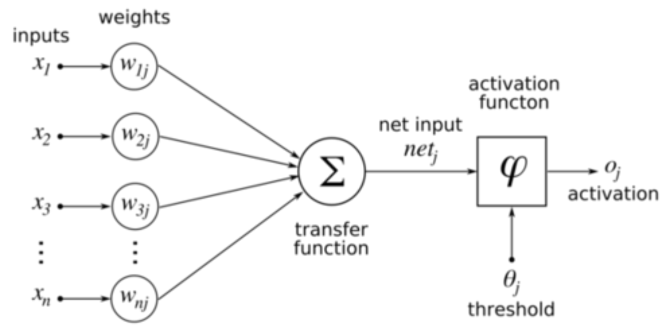

Fig. 1. A Simple Model Neuron [7]

A simple neuron has mapping limitations and thus a number or layers with a large number of neurons are used. The input neuron layer represents each instance which is fed to the hidden layer. The hidden layer computes the output depending on the input it receives, the corresponding weights and the activation function which is then fed to the output layer [7]. Thus the input signal propagates the network layer by layer. The number of the layers depends on the structure of the network.

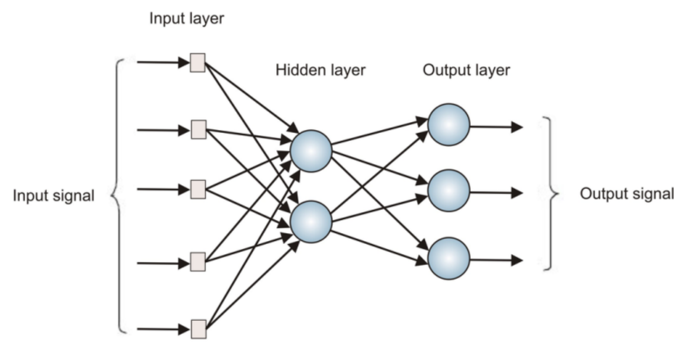

Fig. 2. Structure of a Multi-layer Perceptron [7]

Multilayer perceptron with single layer can solve a wide variety of problems. The weights can initially be assumed randomly and then the network can be trained by backpropagation [8].

B. Support Vector Regression (SVR)

Support Vector Machine can be applied not only to the classification problems but also to the regression problems. Similar to classification, the basic aim is to optimize the generalization bounds for regression [6]. It uses an epsilon intensive loss function which ignores errors that are situated within the certain distance of the true value. The figure below shows an example of one-dimensional linear regression function with epsilon intensive band.

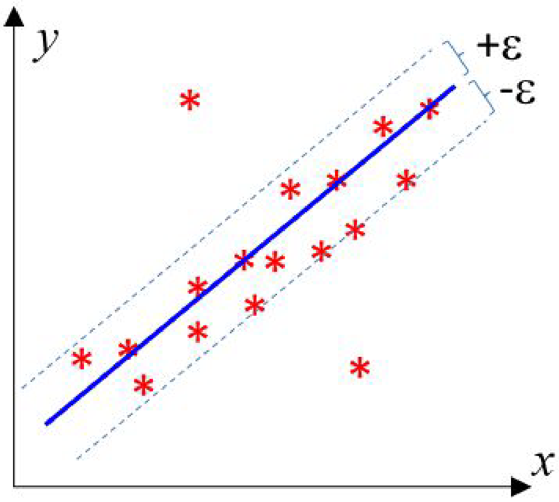

Fig. 3.Linear regression function with epsilon insensitive band [6]

The variables measure the cost of the errors on the training points. The error is zero for all points inside the epsilon band.

The training data points for the support vector regression are as follows:

$$D = \{(x^1,y^1),....,(x^l,y^l)\} \subset \aleph \text{ X } R^n \quad (1)$$

Where ℵ denotes the space of the input patterns. In ε − SV regression, the goal has been to find a function $f(x)$ that has at most ε deviation from the actually obtained targets y^i for all the training data and at the same time as flat as possible [6]. The case of linear function $f(x)$ has been described in the form as:

$$f(x)=\langle w,x\rangle+b \qquad \text{with } \omega \in\aleph,\ b\in\Re \quad (2)$$

Where <.,.> denotes the dot product in ℵ . Flatness in (2) means smaller ω. For this, it is required to minimize the Euclidean norm i.e.$\|\omega\|^2$. Formally this can be written as a convex optimization problem by requiring

$$\text{Minimize} \quad \tfrac{1}{2}\|\omega\|^2$$

$$\text{Subject to} \quad \begin{array}{l} y^i - \langle w,x\rangle - b < \varepsilon \\ \langle w,x\rangle + b - y^i < \varepsilon \end{array} \quad (3)$$

The above convex optimization problem is feasible in cases where f actually exists and approximates all pairs (x^i,y^i) with ε precision. Sometimes, some errors are allowed. Introducing slack variables ξ^i, ξ^{i*} to cope with otherwise infeasible constraints of the optimization problem (3), the formulation becomes

$$\text{Minimize } \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{l}(\xi_i+\xi_i^*)$$

$$\text{subject to } \begin{cases} y_i - \langle \omega, x_i\rangle - b \le \varepsilon + \xi_i \\ \langle \omega, x_i\rangle + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases} \quad (4)$$

The constant C > 0 determines the trade-off between the flatness of *f* and the amount up to which deviations larger than ε are tolerated. ε intensive loss function is described by

$$|\xi|_\varepsilon = \begin{cases} 0 & \text{if } |\xi| < \varepsilon \\ |\xi| - \varepsilon & \text{otherwise} \end{cases} \quad (5)$$

Fig. 4. Depicts the situation graphically.

The dual formulation provides the key for extending SV machine to nonlinear functions. The standard dualization method utilizing Lagrange multipliers has been described as follows:

$$L = \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{l}(\xi_i+\xi_i^*) - \sum_{i=1}^{l}\alpha_i(\varepsilon+\xi_i - y_i + \langle\omega,x_i\rangle + b)$$
$$\sum_{i=1}^{l}\alpha_i^*(\varepsilon+\xi_i^* + y_i - \langle\omega,x_i\rangle - b) - \sum_{i=1}^{l}(\eta_i\xi_i + \eta_i^*\xi_i^*) \quad (6)$$

The dual variables in (6) have to satisfy positivity constraints. It follows from saddle point condition that the partial derivatives of L with respect to the primal variables have to vanish for optimality [6].

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{l}(\alpha_i^* - \alpha_i) = 0$$
$$\frac{\partial L}{\partial \omega} = \omega - \sum_{i=1}^{l}(\alpha_i^* - \alpha_i)x_i = 0$$
$$\frac{\partial L}{\partial \xi_i^{(*)}} = C - \alpha_i^{(*)} - \eta_i^{(*)} = 0 \quad (7)$$

Substituting (7) into (6) yields the dual optimization problem

$$\text{Maximize } \left\{ -\frac{1}{2}\sum_{i,j=1}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)\langle x_i,x_j\rangle - \varepsilon\sum_{i=1}^{l}(\alpha_i+\alpha_i^*) + \sum_{i=1}^{l}y_i(\alpha_i-\alpha_i^*) \right\}$$
$$\text{Subject to } \sum_{i=1}^{l}(\alpha_i-\alpha_i^*)=0 \text{ and } \alpha_i,\alpha_i^* \in [0,C] \quad (8)$$

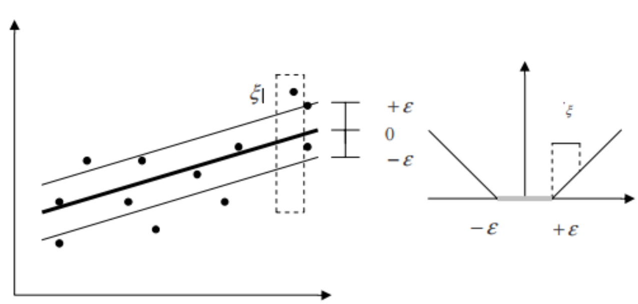

Fig. 4.Soft Margin loss setting corresponds to linear SV machine [6]

Dual variables through condition (7) have been eliminated for deriving (8). Equation (7) can be rewritten as follows:

$$\omega = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) x_i \text{ and therefore, } f(x) = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) \langle x_i, x \rangle + b \quad (9)$$

This is the so-called support vector expansion, i.e. ω can be completely described as a linear combination of the training patterns x^i.

SV algorithm can be made nonlinear by simply pre-processing the training patterns x^i by a map $\phi : X \rightarrow \Im$, into some feature space $\Im$ and then applying the standard SV regression algorithm. The expansion in (9) becomes

$$\omega = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) \phi(x_i)$$

$$f(x) = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) k(x_i, x) + b \quad (10)$$

The difference with the linear case is that ω is no longer explicitly given. In the nonlinear setting, the optimization problem corresponds to finding the flattest function in feature space, not in input space. The standard SVR to solve the approximation problem is as follows:

$$f(x) = \sum_{i=1}^{N} (\alpha_i^* - \alpha_i) k(x_i, x) + b \quad (11)$$

The kernel function k(x, x^i) has been defined as a linear dot product of the nonlinear mapping, i.e.,

$$k(x_i, x) = \varphi(x_i)\varphi(x) \quad (12)$$

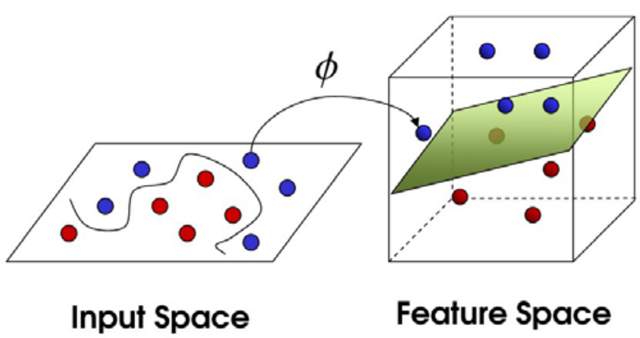

Fig. 5. Transformation of non-linear data to high dimension to make lit linearly separable [6]

C. Time Series Prediction

A time series is a sequence of real-valued signals that are measured at successive time intervals. It has a wide range of applications such as speech analysis, noise cancelation, and stock market analysis (Hamilton (1994); Box et al. (1994); Shumway and Stoffer (2005); Brockwell and Davis (2009)). Roughly speaking, it is based on the assumption that each new signal is a noisy linear combination of the last few signals and independent noise terms. A great deal of work has been done on parameter identification and signal prediction using these models, mainly in the "proper learning" setting, in which the fitted model tries to mimic the assumed underlying model [8]. Most of this work relied on strong assumptions regarding the noise terms, such as independence and identical Gaussian distribution [8].

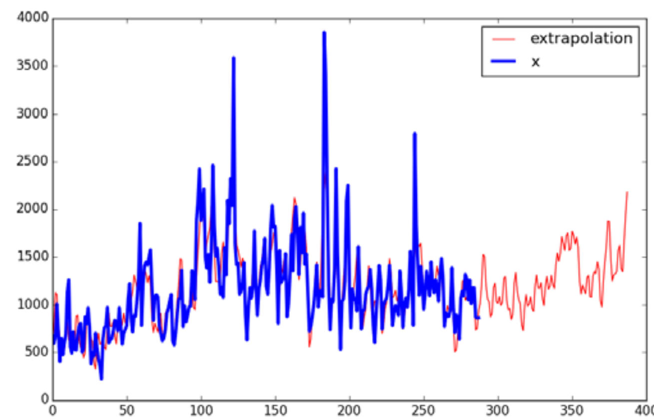

Fig. 6. An illustration of time series prediction where the data in the red is the predicted value [8].

There are three different types of time series predictions depending on the values it considers for prediction [8]:

1) In the first type of time series problem, we predict future values of a time series y(t) from past values of that time series and past values of a second time series x(t).
y(t) = f(y(t – 1), ..., y(t – d), x(t – 1), ..., (t – d))

2) In the second type of time series problem, the future values of a time series y(t) are predicted only from past values of that series. This form of prediction is called nonlinear autoregressive or NAR, and can be written as follows:

   y(t) = f(y(t – 1), ..., y(t – d))

3) The third time series problem is similar to the first type but in this two series are involved, an input series x(t) and an output/target series y(t). This input/output model can be written as follows:

   y(t) = f(x(t – 1), ..., x(t – d))

VII. Results and Discussions

A. Time series prediction using SVR:

First the time series prediction was done using SVR. The input features were selected as for time instances t-1, t-2,........,t-12 and also the values of previous years are considered with these given by t-(12*2), t-(12*3),........., t-(12*10), where t represents the month for which rainfall is to be predicted.

We have data available from 1902 to 2002. As such there was precipitation data available for 1212 months. Out of this the target values were acquired starting from the 121st month since the data for the last 10 years have to be considered for input values.

Now we checked the values of MSE for the model for different values of C and the value of 80 for which the error was least was selected. The least RMSE came out be 52.62 and the mean absolute error (MAE) came out be 33.29. The plot of C values vs. MSE has been shown in fig. 7.

The kernel function that was used was RBF and the gamma value used was .0.0000017 and epsilon value of loss function used was 19.5 which were achieved through plotting the MSE vs. the desired variable and the values for which MSE was least were used in the SVM kernel function. The most optimum value of performance has been achieved in this way. 80% of the instances were used for training and 20% were used for testing the data.

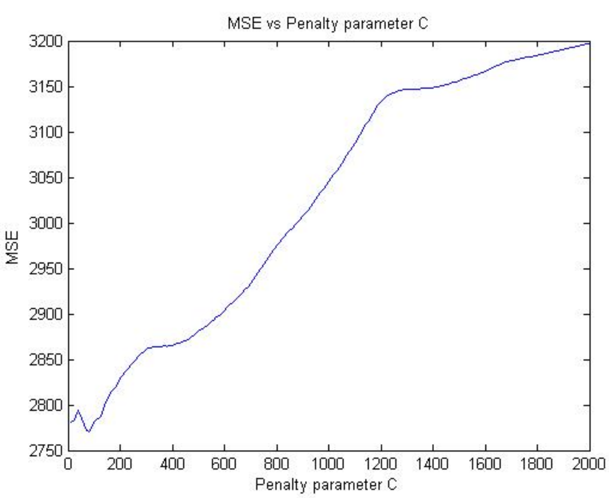

Fig. 7. SVR MSE with penalty parameter C for time series model

B. Prediction model based on climatic features and trained using SVR

The input features used here were various climate related attributes like cloud cover, temperature, etc. Overall there were 10 features used and the training was done for 80% of the instances and testing was done for 20% of the instances.

Now we checked the values of MSE for the model for different values of C and the value of 1500 for which the error was least was selected. The least RMSE came out be 20.22 and the mean absolute error (MAE) came out be 13.23. The plot of C values vs. MSE has been shown in fig. 8.

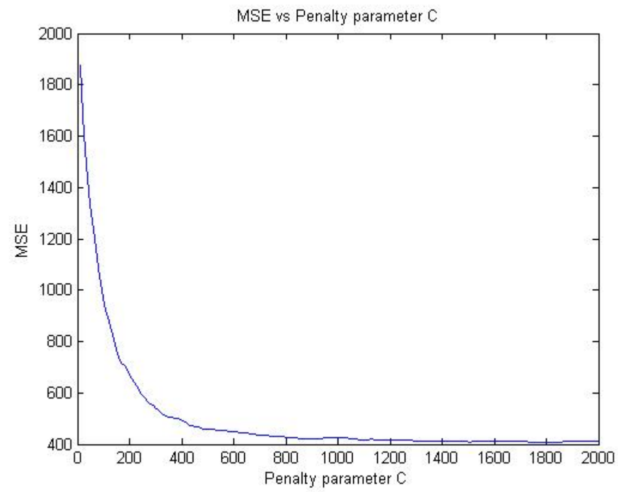

Fig. 8. SVR MSE with penalty parameter C for conventional model

The kernel function that was used was RBF and the gamma value used was .0.00019 and epsilon value of loss function used was 19.6 which were achieved through plotting the MSE vs. the desired variable and the values for which MSE was least were used in the SVM kernel function. The most optimum value of performance has been achieved in this way.

C. Time series prediction using MLLP

The number of hidden neurons used were 10. The feature engineering done was same as that done for SVR. Here again 80% of the instances were used for training and 20% were used for testing the data.

The RMSE came out to be 65.06 and MAE came out to be 44.06.

The performance plot has been shown in fig. 9 and it indicates the best performance for validation set.

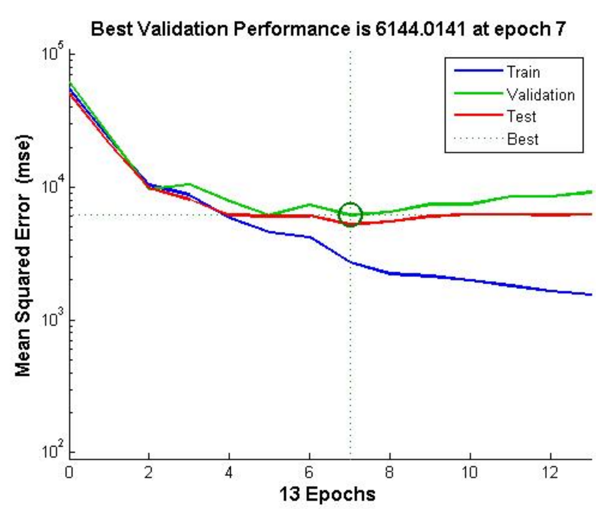

Fig. 9 MLP MSE for time series model

D. Prediction model based on climatic features and trained using MLP

The number of hidden neurons used were 10. The feature engineering done was same as that done for SVR. Here again 80% of the instances were used for training and 20% were used for testing the data.

The RMSE came out to be 21.49 and MAE came out to be 14.09.

The performance plot has been shown in fig. 10 and it indicates the best performance for validation set.

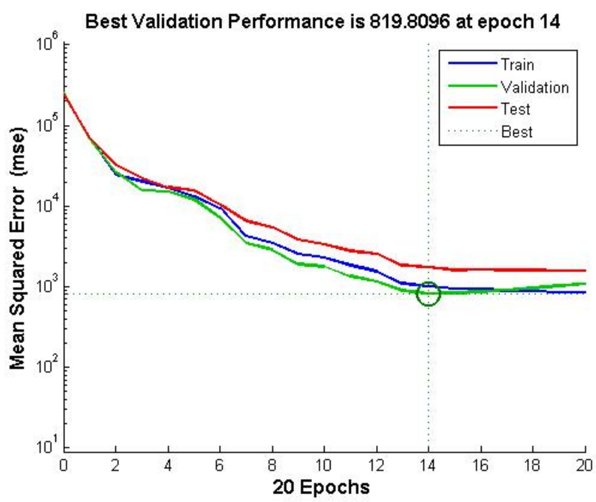

Fig. 10 MLP MSE for conventional model.

VIII. Conclusion

We have tested four different models using two different machine learning techniques and two different feature engineering techniques. It has been observed that the Support Vector Regression technique based on climatic features has outperformed others. The RMSE calculated in this case was 20.22 and MAE calculated was 13.23. The Mean absolute error (MAE) and root mean squared error (RMSE) for the four models has been shown in the table. The prediction of rainfall is very important in a developing country like India where farmers depend mostly on rainfall for the irrigation of their crops. In future some different models of machine learning can be applied to get better results and other meteorological variables can be determined using the same ideology.

Regression Model	MAE	RMSE
SVR Time Series	33.29	52.62
SVR Conventional Model	13.23	20.22
MLP Time Series	44.06	65.06
MLP Conventional Model	14.09	21.49

TABLE II. MAE and MSE for different models

References

[1] Maria Cleofé Valverde Ramíreza, Haroldo Fraga de Campos Velhob,Nelson Jesus Ferreiraa, *Artificial neural network technique for rainfall forecasting applied to the São Paulo region* in Journal of Hydrology, Volume 301, Issue 1, p. 146-162 January (2005).

[2] Kalogirou, S. A., Neocleous, C., Constantinos, C. N., Michaelides, S. C. & Schizas, C. N.*, A time series construction of precipitation records using artificial neural networks* in Proceedings of EUFIT '97 Conference, 8–11 September, Aachen, Germany. pp 2409–2413 1997.

[3] Paavni Rattan, Rishita Anubhai, Amogh Vasekar, *Contextual-window based neural network model for predicting rainfall in rural India*, Stanford course project CS 229, 2012

[4] India Water Portal. Arghyam, <http://www.indiawaterportal.org/metdata>.

[5] Dai, A.*, Characteristics and trends in various forms of the Palmer Drought Severity Index during 1900–2008*, 2011, J. Geophys. Res., 116, D12115, doi:10.1029/2010JD015541.

[6] Alex J. Samola, Bernard S, *A tutorial on Support Vector Regression*, NeuroCOLT2 Technical Report NCR2, October 1998.

[7] M.W.Gardener, S.R. Dorling, *ARTIFICIAL Neural Networks(The Multilayer perceptron)- A review of the applications in Atmospheric Sciences,* June 1998.

[8] *Pattern Recognition and Machine Learning*, book by Christopher Bishop, 2007