SUGARCANE YIELD FORECASTING USING ARTIFICIAL NEURAL NETWORK MODELS

Sunil Kumar [1], Vivek Kumar[2] and R. K. Sharma[3]

[1]Research Scholar, Department of Computer Science,
Gurukula Kangri Vishwavidyalaya, Haridwar, India
[2]Associate Professor, Department of Computer Science,
Gurukula Kangri Vishwavidyalaya, Haridwar, India
[3]Professor, Department of Computer Science and Engineering,
Thapar University, Patiala, India

Abstract

Neural networks have gained a great deal of importance in the area of soft computing and are widely used in making predictions. The work presented in this paper is about the development of Artificial Neural Network (ANN) based models for the prediction of sugarcane yield in India. The ANN models have been experimented using different partitions of training patterns and different combinations of ANN parameters. Experiments have also been conducted for different number of neurons in hidden layer and the algorithms for ANN training. For this work, data has been obtained from the website of Directorate of Economics and Statistics, Ministry of Agriculture, Government of India. In this work, the experiments have been conducted for 2160 different ANN models. The least Root Mean Square Error (RMSE) value that could be achieved on test data was 4.03%. This has been achieved when the data was partitioned in such a way that there were 10% records in the test data, 10 neurons in hidden layer, learning rate was 0.001, the error goal was set to 0.01 and traincgb algorithm in MATLAB was used for ANN training.

Keywords
ANN, BPNN, RMSE, Sugarcane yield, Forecasting

1. INTRODUCTION

Despite rapid urbanization everywhere, agriculture continues to be the mainstay of a large percentage of population the world over. Technological advancements may have led to fairly accurate weather predictions, resulting in improved yields, but a lot still needs to be done so that farmers get a bankable method, preparing them for future yields. Statistically sound objective forecasts of crop yields based on weather variables are required so that comprehensively reliable forecasts can be obtained. Most of the earlier works in this field have utilized regression models, discriminant function analysis, agro-meteorological models, etc. for crop yield forecasting (to cite a few, [1]-[4]).

ANN, of late, has been garnering a lot of attention because it is turning out to be a trustworthy tool to treat complicated problems even in the face of the data being imprecise and imperfect. A detailed study of ANN models vis-à-vis traditional statistical models has been done by Cheng and

DOI : 10.5121/ijaia.2015.6504

Titterington [5] who established that statistical procedures like regression, principal component analysis, density function and statistical image analysis can be given neural network expressions. They arrived at these findings after reviewing the relevant literature on ANNs. They explained the learning algorithm and made a comparison between regression and neural network models in terms of notations, terminologies and implementations.

Kaastra and Boyd [6] developed neural network model for forecasting financial and economic time series. Dewolf and Francl [7]-[8] demonstrated the applicability of neural network technology for plant diseases forecasting. Zhang *et al.* [9] provided a general summary of the work in ANN forecasting, providing the guidelines for neural network modeling, general paradigm of the ANNs especially those used for forecasting. They reviewed the relative performance of the ANNs with the traditional statistical methods and, invariably, ANNs were found to be better than the latter. Chakraborty *et al.* [10] utilized the ANN technique for predicting severity of anthracnose diseases in legume crop. Gaudart *et al.* [11] compared the performance of multilayer perceptron (MLP) and that of linear regression for epidemiological data with regard to quality of prediction and robustness to deviation from underlying assumptions of normality, homoscedasticity and independence of errors and it was found that MLP performed better than linear regression.

A broad application domain for neural networks has been forecasting the behaviour of complex systems. In particular, ANN models have been used in applications such as electric load forecasting [12]-[13], economic forecasting [14], river flow forecasting [15], forecasting student admission in colleges [16], empirical comparisons of feed-forward connectionist and conventional regression models for prediction of first lactation 305-day milk yield in Karan Fries Dairy Cows [17], short-term load forecasting [18], forecasting the Saudi Arabia stock prices [19], short term electricity price forecasting [20], stock market prediction [21], automatic apple grading model development [22].

Thusly, present day necessity is that of models that can foresee the future agricultural production in view of recorded information and to develop agriculture based decision support systems [23]-[24]. Anticipating the conduct of complex frameworks has been a wide application area for neural systems. ANNs being data driven, self-adaptive methods with hardly any prior assumptions, subtle functional relationships among the data are captured even if the underlying relationships are hard to describe. ANNs can identify and learn correlated patterns between input data sets and corresponding target values through training.

In the present work, ANN models have been developed with respect to sugarcane production data of India for past 61 years (1950 - 2011). The production of sugarcane, as any other crop, depends on a lot of extraneous factors like weather and climate conditions, agricultural policy and social factors [25]. Hence the forecast of yield of any crop becomes a difficult exercise and here comes the reliability of ANN for it can handle multivariate non-linear non parametric statistical approach effectively [5], [26], [27]. When compared with mechanistic models, ANN is a powerful but simple empirical approach. ANN models are more comprehensive empirical models as compared to the linear regression methods adopted for sugarcane yield because the sugarcane yield is non-linear and autoregressive in nature. ANN models do not make use of rigorous assumptions regarding distribution of samples [28]-[29], but solely depend on illustrations of complex and non-linear relationships. Hence, wherever training data is readily available, regardless of the possibility that there is inadequate comprehension of the problem to be unravelled, ANN technique ventures in.

Since the ANN works like a human brain, so the human brain's biological, neural processes may be used for development of empirical agronomic models. The models are developed on the interrelationship of variables that are correlated and symbolize neurons or nodes of the brain. ANN models find relationships by observing a large number of input and output examples to develop a formula that can be used for prediction [30]. Nonlinear relationships overlooked by other methods can be determined with little prior knowledge of the functional relationship [31]. There are at least three layers in a typical ANN model - the input, hidden and output layers. Nodes in input layer correspond to input variables and nodes in output layer relate with output variables. Data move between layers across weighted connections. A weighted sum (t_i) of all inputs is calculated from data accepted by nodes in the previous layer as

$$t_i = \sum_{j=1}^{n} w_{ij} x_j$$

Here, n is the number of inputs, w_{ij} is the weight of the connection between nodes j and i, and x_j is the input from node j. A transfer function is now applied on t_i in order to get node output, $O_{i.}$ The hidden layer uses linear transfer function whereas the input layer uses, usually, a sigmoidal function as transfer function.

The number of hidden nodes determines the number of connections between inputs and outputs. It also differs from one problem to the other. However, there must not be a large number of nodes in hidden layer, because the model may get over-trained by memorizing the training data eventually leading to not very reliable predictions [32]. The *learning rate (r)* determines the amount the weights change during a series of iterations to bring the predicted value within an acceptable range of the observed value. The *training tolerance* or *error goal* (*e*) refers to the maximum *error* at which the network must converge during training. An approximate function is developed and utilized for future predictions once the network converges [33]. The trained network is then tested with a separate data set with its output information omitted.

The 61 year information may mirror the wide varieties in sugarcane production on account of normal and man-made disasters, yet ANN has the capacity of enrolling the information giving due thought to the clamour bringing about components but then wind up with a suitable model with high precision [34]-[35]. The present work tries to understand and analyze crop forecasting techniques using different back propagation ANN models and with different partitioning strategies, different *learning rate (r)*, different *error goal* (or *training tolerance)(e)* and finally comparing all these partitioning strategies keeping in view their minimum error.

Though ANN applications in different fields of agriculture are growing [36], there has been very little research into application of ANNs in the field of sugarcane production, more so in India. Literature review does not report a significant study on agriculture yield prediction in India as well as abroad using ANN [37] - [44]. Subsequently, this study has been attempted to examine the adequacy of *back-propagation neural networks* (*BPNNs*) to anticipate sugarcane yield in India.

2. METHODOLOGY

2.1 Data collection

The data set covers the sugarcane yield in India from the year 1950 to 2011 [45]. In this implementation, a three input, one output neuron model has been considered. Considering the yield data of year 1950, 1951, and 1952 as input values and the yield of year 1954 as output value, a training pattern has been developed. This strategy of using three consecutive years as input and the fourth year data as output was adopted for training the whole network. This idea was extended to generate 58 training patterns from the collected data.

2.2 Artificial neural network model development

As reported earlier, we have experimented with a number of ANN models in this work. These models have 3 input neuron, 1 output neuron structure with 1 hidden layer. The number of neurons in the hidden layer has been taken as 10, 15, 20, and 25 for the experimentation. The *learning rate* (*r*) was considered as 0.01, 0.001 and 0.0001. The *training tolerance* (*e*) is taken as 0.01, 0.005, 0.001 and 0.0005. Different data partitioning strategies have also been explored in the experiments. These are 70, 75, 80, 85, or 90 percent records kept in 'train set' and remaining records in the 'test set'. These data partitioning strategies are named as '*v*', '*w*', '*x*', '*y*', and '*z*' in this work.

Training of the models has been done using ten back-propagation learning algorithms, *viz., Variable Learning Rate Gradient Descent* (*GDX*), *Gradient descent with adaptive learning rate* (*GDA*), *Polak-Ribiére, Resilient Back propagation* (*RP*), *Fletcher-Powell Conjugate Gradient* (*CGF*), *Conjugate Gradient* (*CGP*), *Conjugate Gradient with Powell/Beale Restarts* (*CGB*), *Scaled Conjugate Gradient* (*SCG*), *BFGS Quasi-Newton* (*BFG*), *One Step Secant* (*OSS*), and *Levenberg-Marquardt* (*LM*) up to 50000 epochs or till the algorithms actually converged. Initialization of weights and bias matrix was done in a random manner taking values between -1 and +1. Different *learning rate* (*r*), and *error goal* (*e*) were used for these learning algorithms. These parameters of ANN modelling give rise to 2160 different ANN models for the purpose of experimentation. It has been observed that these models converged most of the times indicating that *error goal (e)* was achieved. The output from the aggregation of weighted inputs of neurons in hidden layer was calculated with non-linear activation function, *tangent sigmoid.* Network response was generated using *pure linear* transformation function at output layer.

The performance of developed models was evaluated using *Root Mean Square Error (RMSE)* defined as,

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}((X_{obs,i} - X_{model,i})/X_{obs,i})^2}{n} X100\%}$$

where X_{obs} is observed value and X_{model} is modelled value at time/place *i*.

RMSE measures the magnitude of forecast error. The *RMSE* gives relatively high weights to large errors [35]. This means RMSE is most useful when large errors are undesirable. Trial and error approach is used to determine the optimum network architecture to achieve maximum prediction accuracy. A multilayer feed forward with *back propagation* of error learning mechanism was developed in MATLAB 7.0 software. All the figures presented here are crafted using MS-Excel 2013.

3. RESULTS AND DISCUSSION

There are five data partitioning strategies that have been used for training ANN. There were three different *learning rate* (*r*) and four different *error goal* (*e*) using different number of *neurons* (*n*) in hidden layer, *i.e*., 10, 15, 20, and 25. Also, ANN models could converge when nine back propagation learning algorithms viz., *GDA*, *RP*, *CGF*, *CGP*, *CGB*, *SCG*, *BFG*, *OSS* and *LM* were used. However, the ANN models could not converge when training algorithm *GDX* was used. The performance of the developed models was calculated using *Root Mean Square Error*, as given in previous section.

The results for those combinations of parameters for which RMSE is on an average less than 10% for all data partitioning strategies have been included in this section. Figures 1 to 16 depicts these results presenting *RMSE* for different partitioning strategies and learning functions. The figures also include *learning rate (r), error goal (e)* and number of *neurons (n)* in the hidden layer that are used for that graph. The convergence factor indicates that ANN was able to develop a yield prediction function because error goal was achieved. Table 1 summarizes the important points from these graphs.

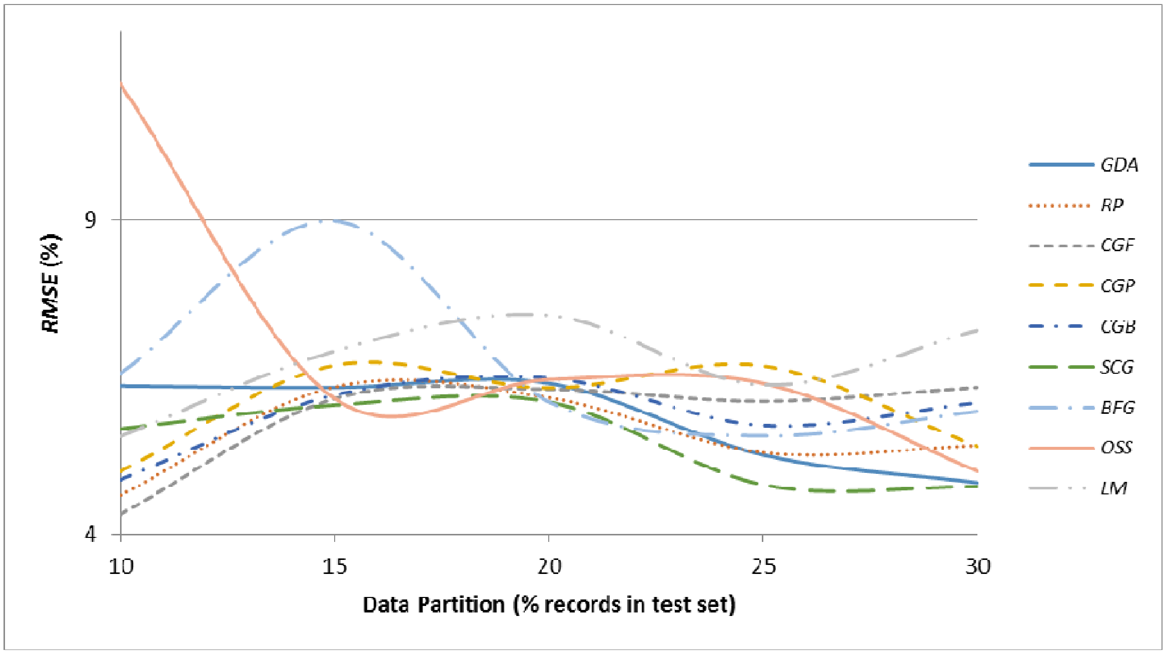

Figure 1. Effect of data partitioning on performance of prediction models using $r = 0.01$, $e = 0.01$, and $n = 10$

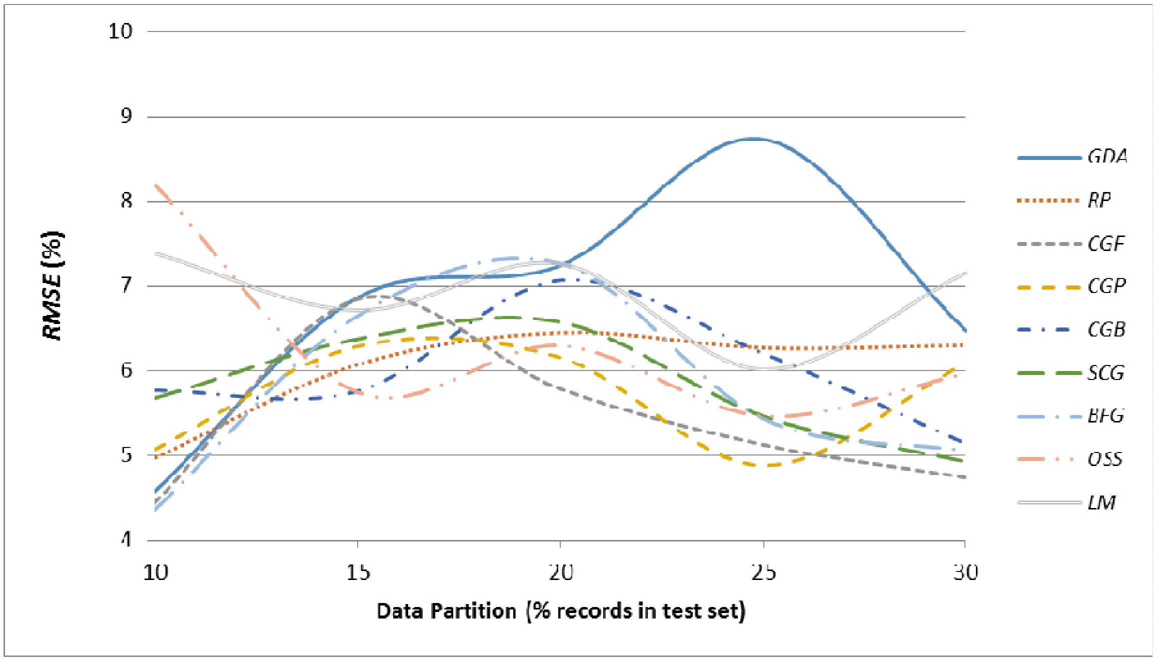

Figure 2. Effect of data partitioning on performance of prediction models using $r = 0.01$, $e = 0.01$, and $n = 15$

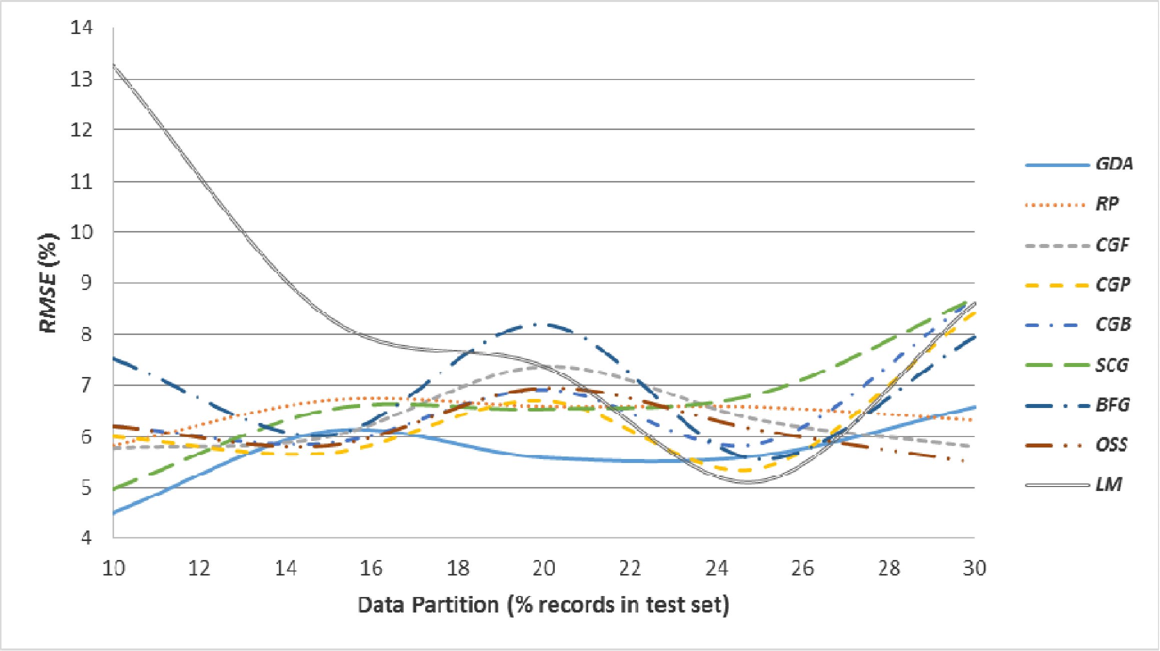

Figure 3. Effect of data partitioning on performance of prediction models using $r = 0.01$, $e = 0.01$, and $n = 20$

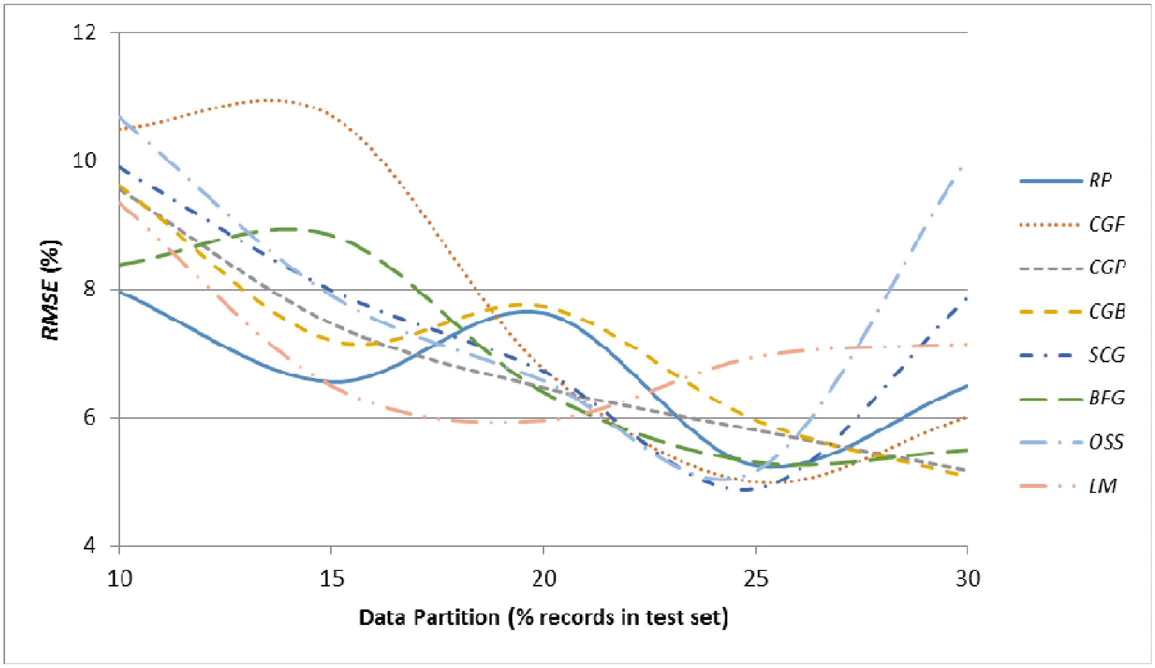

Figure 4. Effect of data partitioning on performance of prediction models using r = 0.01, e = 0.005, and n = 15

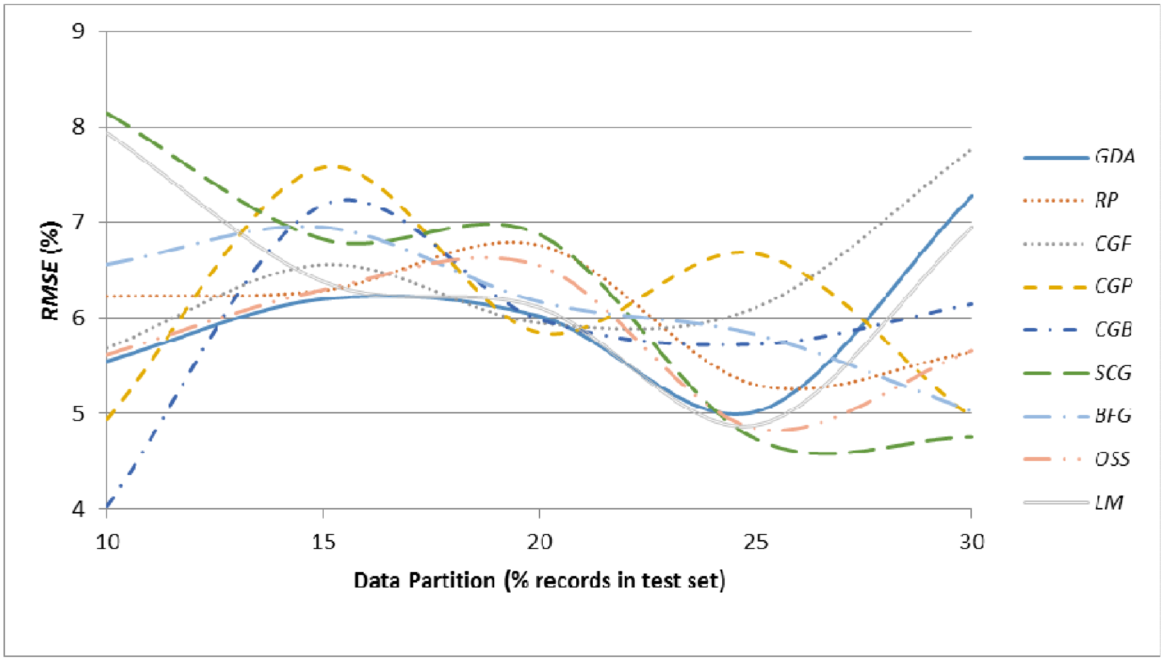

Figure 5. Effect of data partitioning on performance of prediction models using *r* = 0.001, *e* = 0.01, and *n* = 10

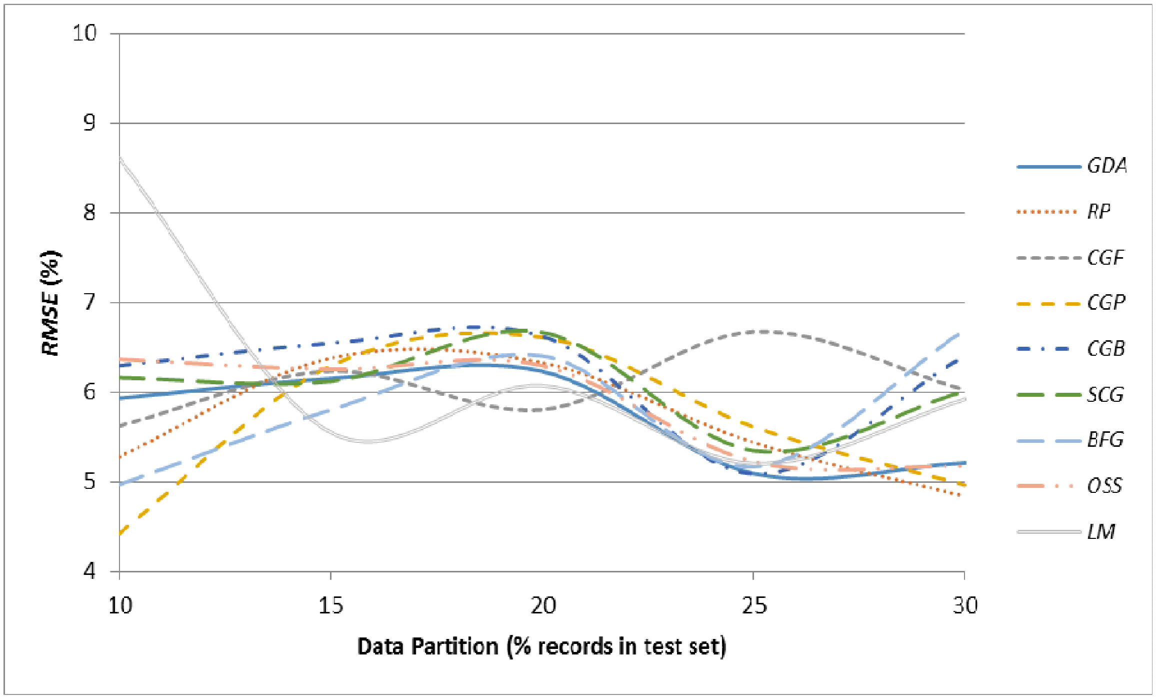

Figure 6. Effect of data partitioning on performance of prediction models using *r* = 0.001, *e* = 0.01, and *n* = 15

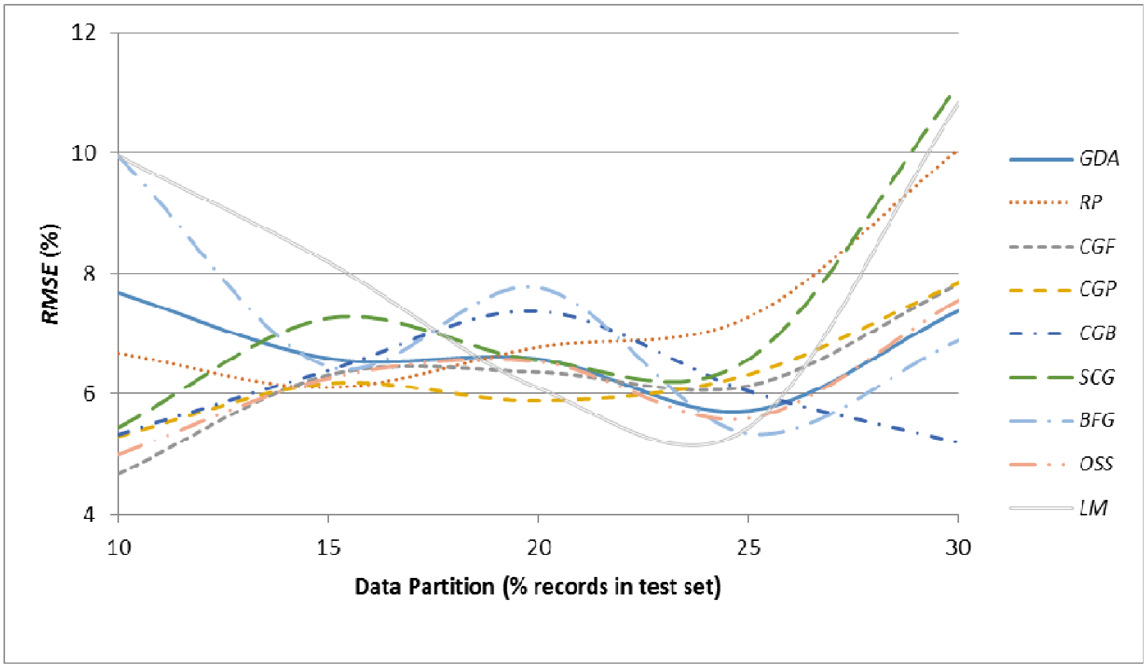

Figure 7. Effect of data partitioning on performance of prediction models using *r* = 0.001, *e* = 0.01, and *n* = 20

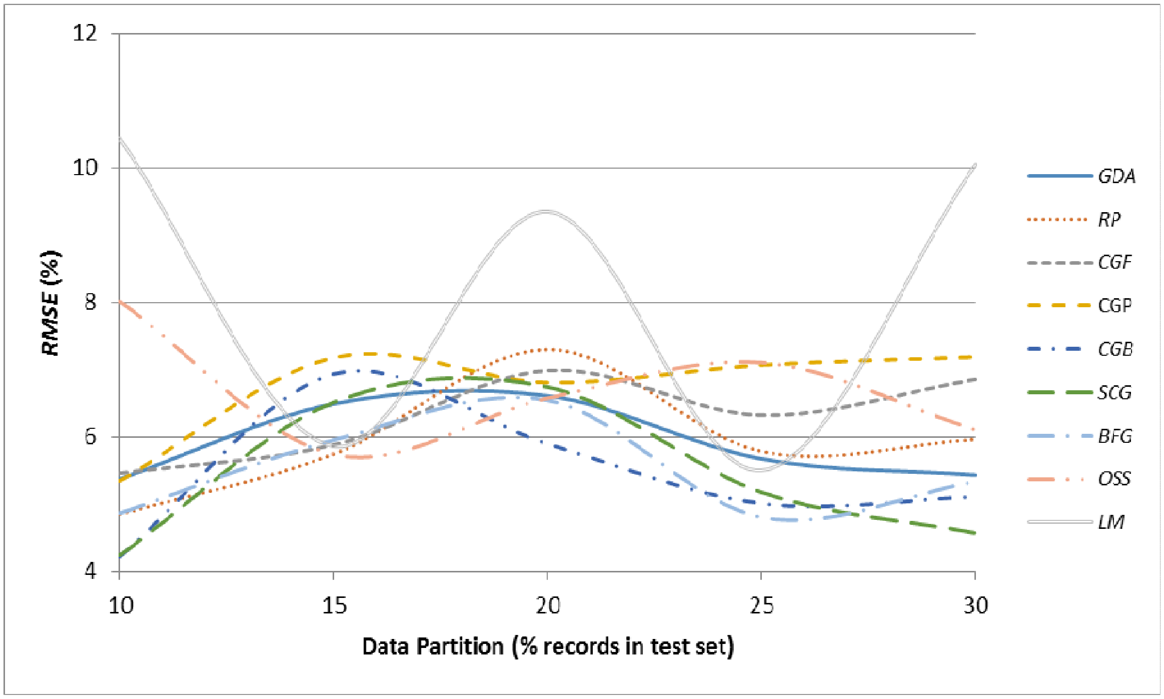

Figure 8. Effect of data partitioning on performance of prediction models using *r* = 0.001, *e* = 0.01, and *n* = 25

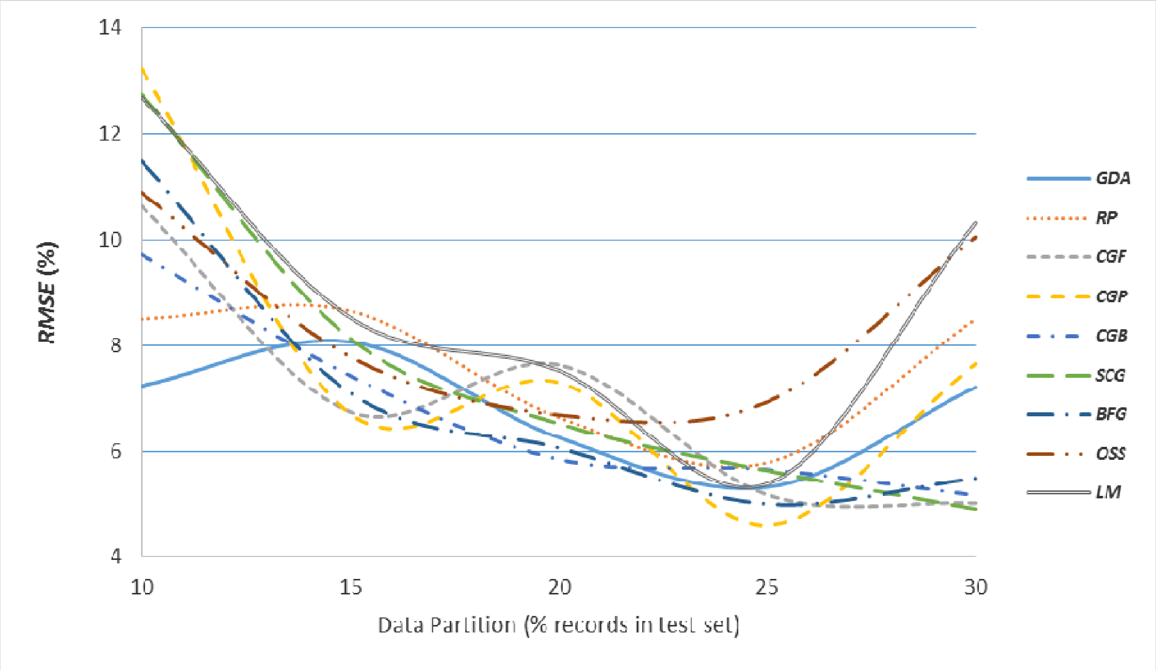

Figure 9. Effect of data partitioning on performance of prediction models using *r* = 0.001, *e* = 0.005, and *n* = 15

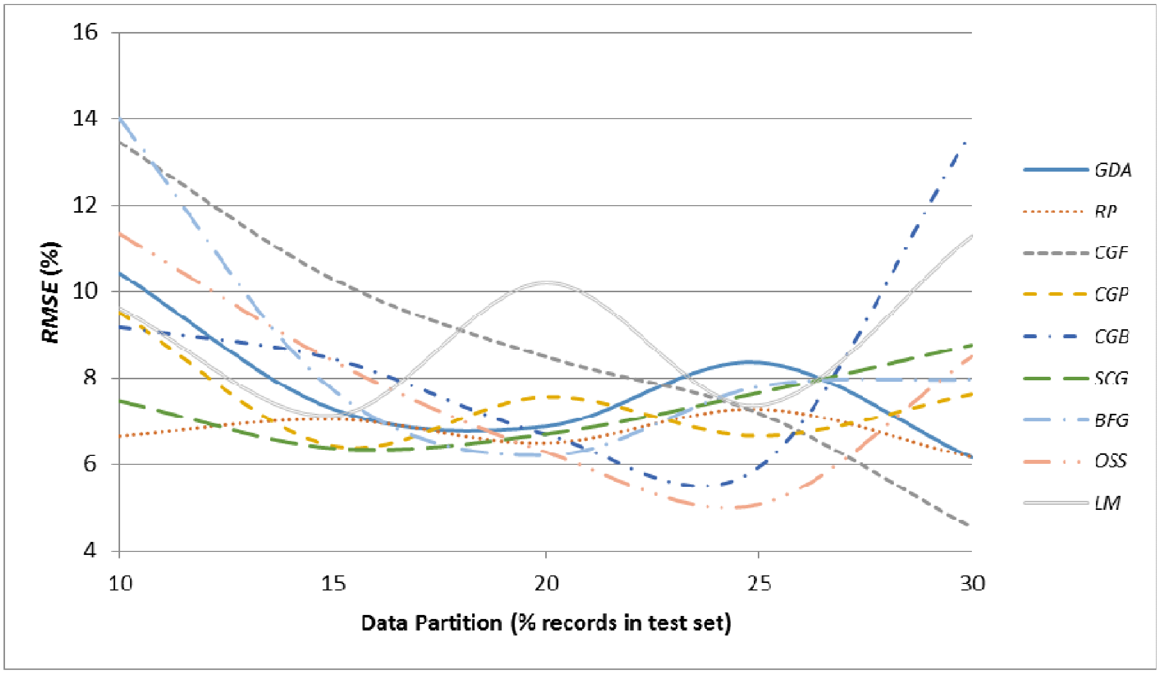

Figure 10. Effect of data partitioning on performance of prediction models using $r = 0.001$, $e = 0.005$, and $n = 25$

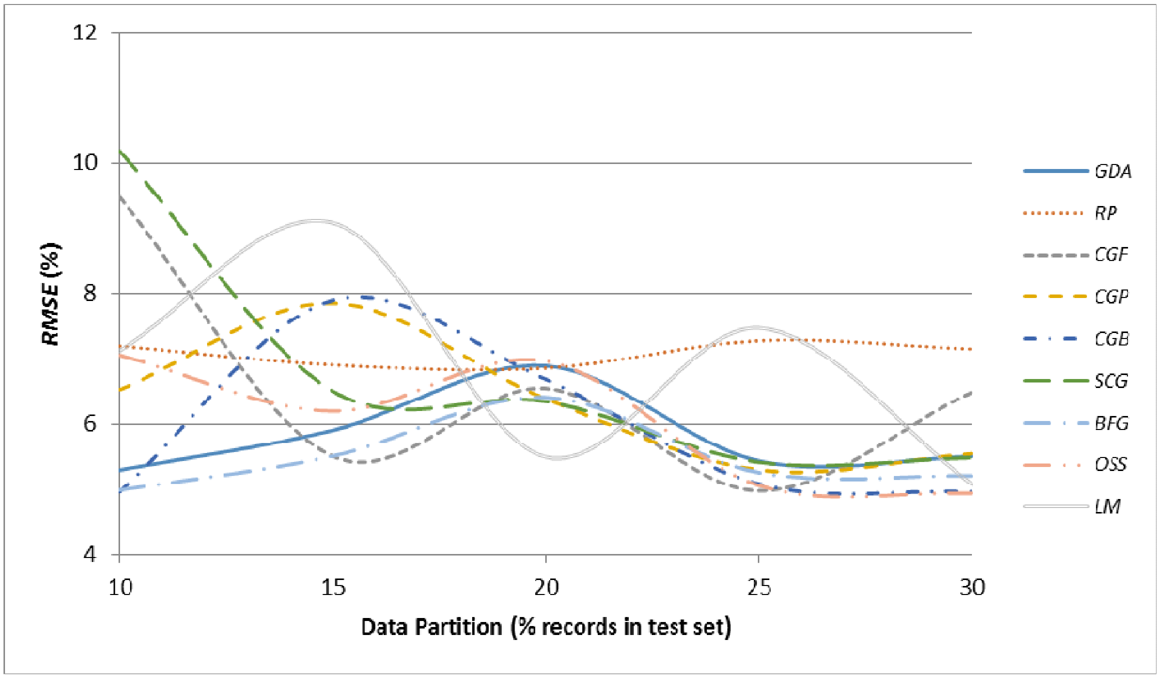

Figure 11. Effect of data partitioning on performance of prediction models using $r = 0.0001$, $e = 0.01$, and $n = 10$

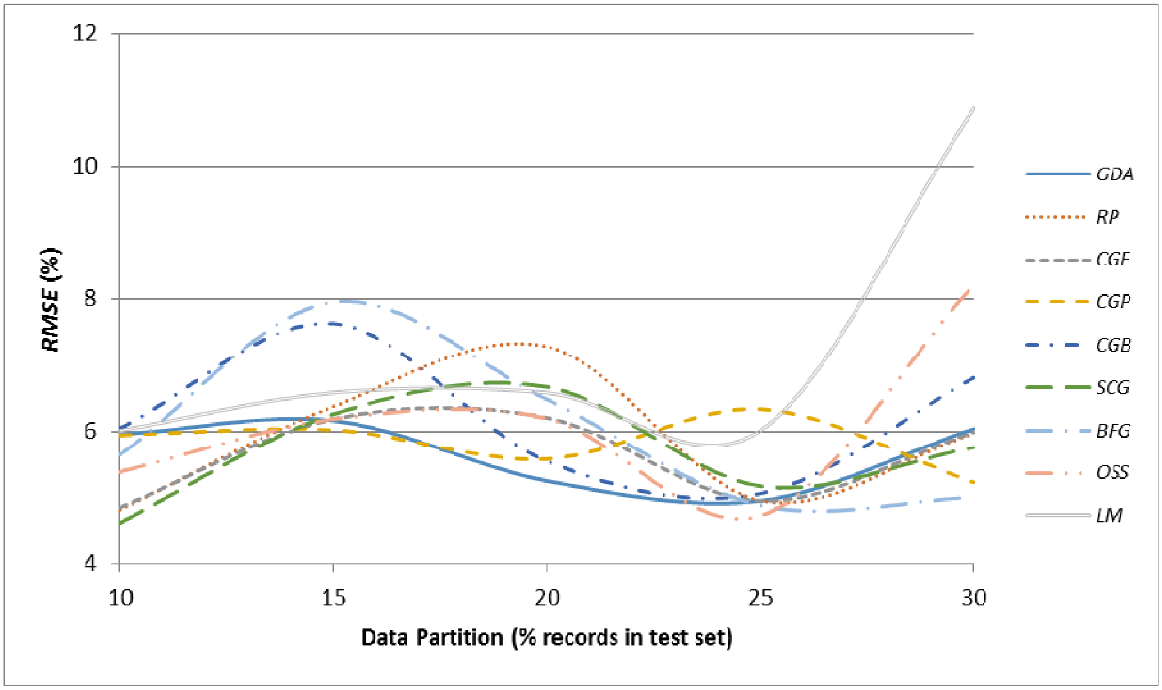

Figure 12. Effect of data partitioning on performance of prediction models using *r* = 0.0001, *e* = 0.01, and *n* = 15

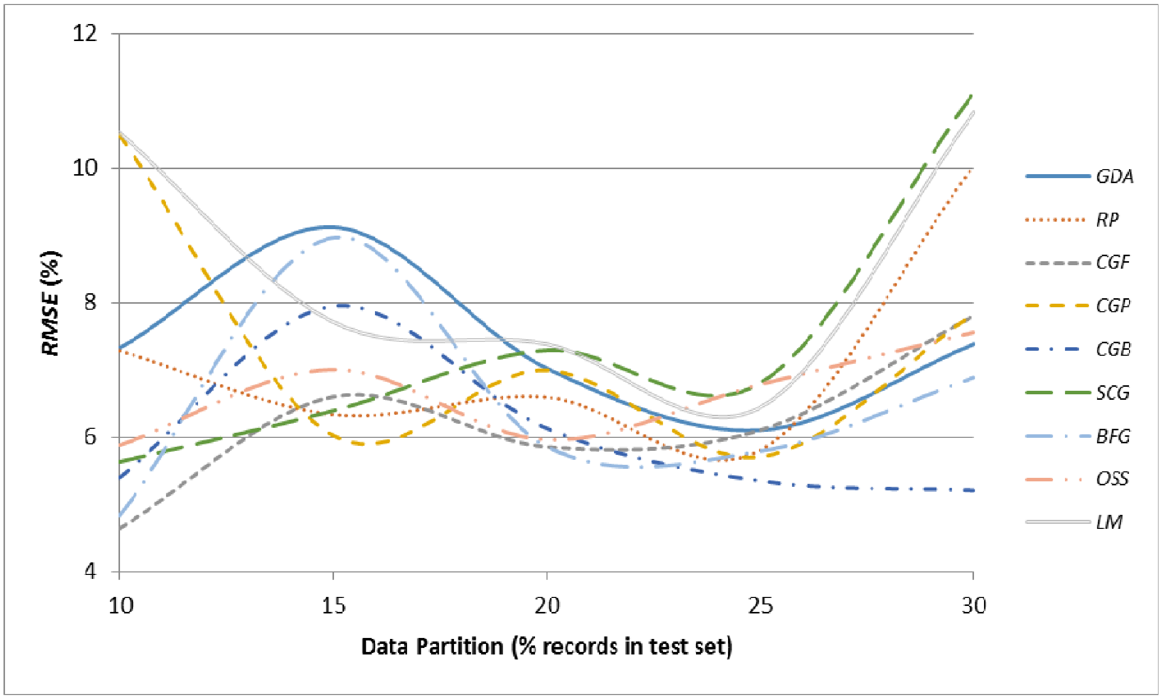

Figure 13. Effect of data partitioning on performance of prediction models using *r* = 0.0001, *e* = 0.01, and *n* = 20

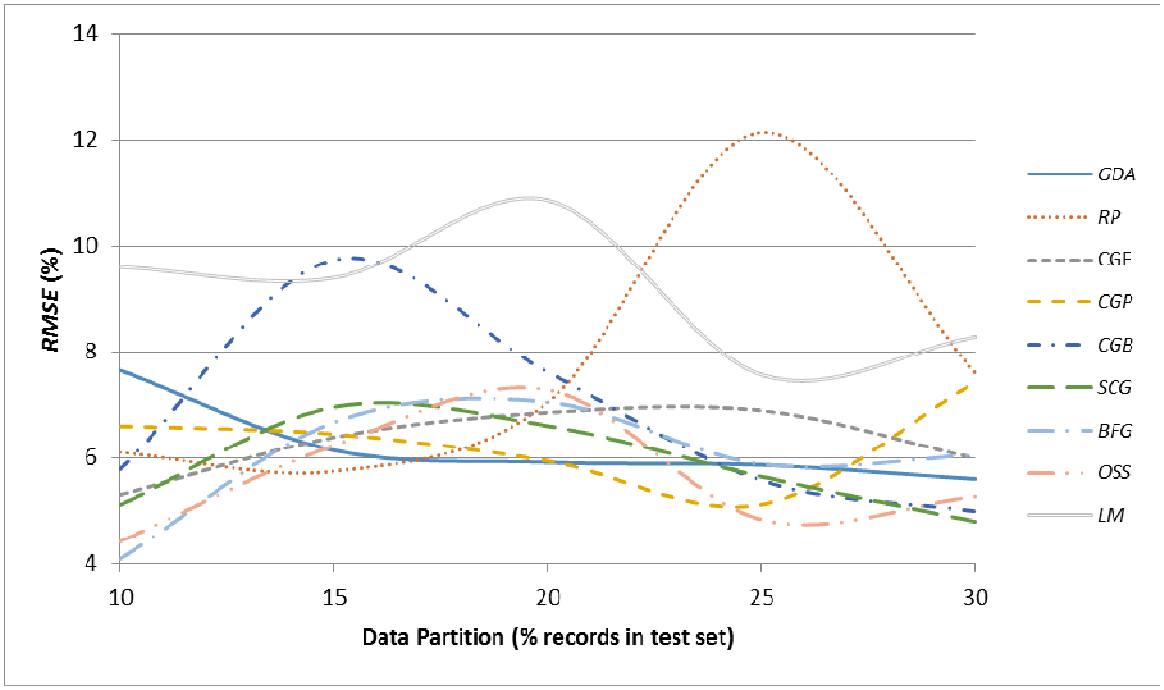

Figure 14. Effect of data partitioning on performance of prediction models using $r = 0.0001$, $e = 0.01$, and $n = 25$

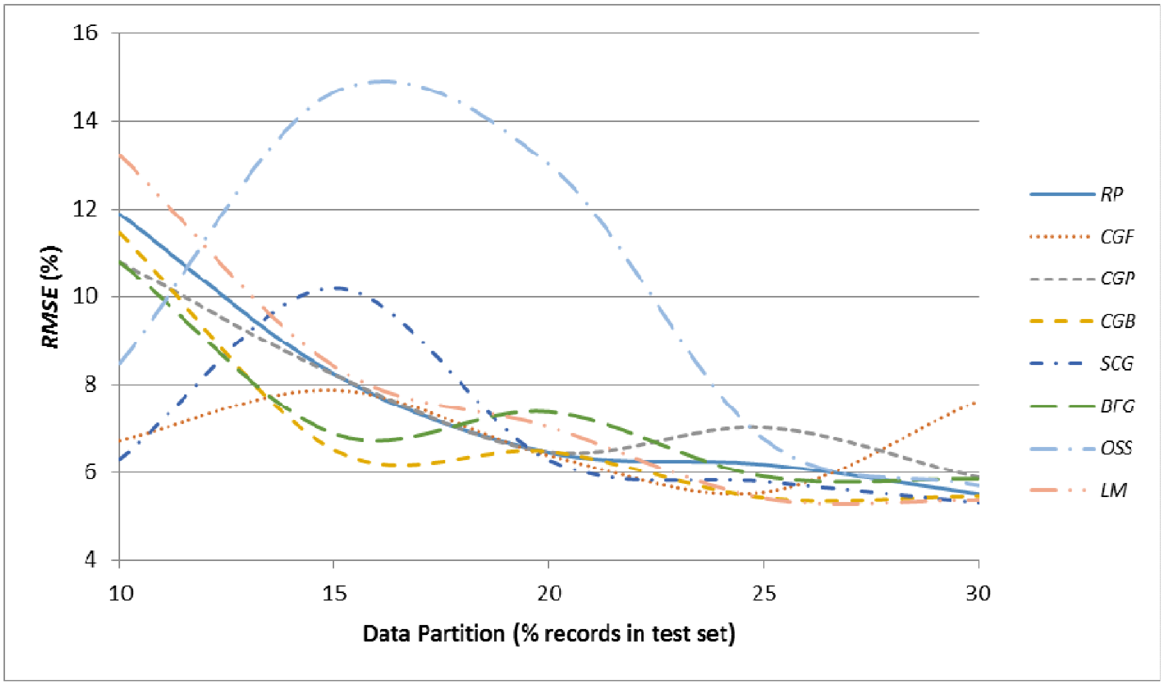

Figure 15. Effect of data partitioning on performance of prediction models using $r = 0.0001$, $e = 0.005$, and $n = 10$

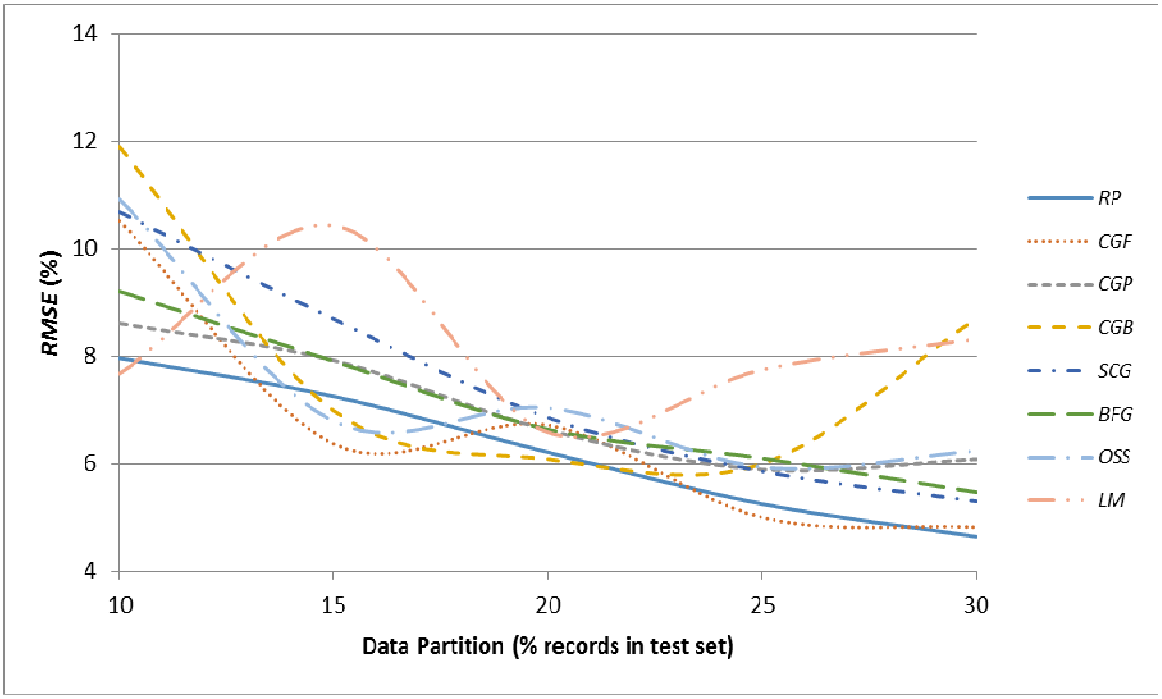

Figure 16. Effect of data partitioning on performance of prediction models using $r = 0.0001$, $e = 0.005$, and $n = 15$

Table 1
Prominent Feature extracted from Graphs in Figures 1-16.

Figure Number	**Parameters (r, *e*, *n*)**	**Learning function (s) that could not converge, with data partitioning strategy**	**Lowest *RMSE* achieved with partition strategy and learning function**
Figure 1	$r = 0.01$, $e = 0.01$, $n = 10$	---	4.31% (*z*, *CGF*)
Figure 2	$r = 0.01$, $e = 0.01$, $n = 15$	---	4.35% (*z*, *BFG*)
Figure 3	$r = 0.01$, $e = 0.01$, $n = 20$	---	4.51% (*z*, GDA)
Figure 4	$r = 0.01$, $e = 0.005$, $n = 15$	*GDA*, *z*	4.89% (***w***, SCG)
Figure 5	$r = 0.001$, $e = 0.01$, $n = 10$	---	4.03% (*z*, *CGB*)

Figure 6	*r* = 0.001, *e* = 0.01, *n* = 15	---	4.42% (***z*, *CGP***)
Figure 7	*r* = 0.001, *e* = 0.01, *n* = 20	---	4.67% (***z*, *CGF***)
Figure 8	*r*= 0.001, *e* = 0.01, *n* = 25	---	4.22% (***z*, *CGB***)
Figure. 9	*r* = 0.001, *e* = 0.005, *n* = 15	---	4.62% **(*w*, *CGP*)**
Figure 10	*r* = 0.001, *e* = 0.005, *n* = 25	---	4.53% (***v***, *CGF*)
Figure 11	r = 0.0001, *e* = 0.01, *n* = 10	---	4.94% (***v***, OSS)
Figure 12	*r* = 0.0001, *e* = 0.01, *n* = 15	---	4.62% (***z***, SCG)
Figure 13	*r* = 0.0001, *e* = 0.01, *n* = 20	---	4.63% (***z***, *CGF*)
Figure 14	*r* = 0.0001, *e* = 0.01, *n* = 25	---	4.08% (***z***, *BFG*)
Figure 15	*r* = 0.0001, *e* = 0.005, *n* = 10	*GDA*, ***v***, ***w***, ***x***, ***y***, *z*	5.29% (***v*, *SCG***)
Figure 16	*r* = 0.0001, *e* = 0.005, *n* = 15	*GDA*, ***v***, ***w***, ***x***, ***y***, *z*	4.64% (***v***, *RP*)

In these experiments, it has also been found that the network with one hidden layer containing 10 neurons could achieve a minimum *RMSE* of **4.03%** (*i.e.,* 95.97% prediction accuracy).The parameters for this case are given below.

Parameters for the model with minimum *RMSE*

Numbers of hidden layers	1
Number of hidden neurons (*n*)	10
Learning rate (*r*)	0.001
Error goal (*e*)	0.01
Numbers of epochs	50000
Data partition strategy	***z***
Training function	*traincgb*

4. **CONCLUSION**

In this work, we have developed ANN based models for the prediction of sugarcane yield in India. Different partitions of training patterns and different combinations of ANN parameters using MATLAB have been availed for these experiments with ANN models. The data set takes in its purview the sugarcane yield in India from the year 1950 to 2011. ANN models were trained with three different learning rates (*r*), namely, 0.01, 0.001, 0.0001; four different error goals (*e*), viz., 0.01, 0.005, 0.001, 0.0005; four different number of neurons (*n*) in hidden layer, i.e., 10, 15, 20, and 25; five different data partitioning strategies; and nine back propagation learning algorithms, viz., *GDA*, *RP*, *CGF*, *CGP*, *CGB*, *SCG*, *BFG*, *OSS* and *LM*. *RMSE* was used to evaluate the performance of developed model. It was found that the best network with one hidden layer containing 10 neurons could achieve 4.03% *RMSE*. This was achieved when the data was partitioned in such a way that there were 10% records in the test data, 10 neurons in hidden layer, learning rate was 0.001, the error goal was set to 0.01 and *traincgb* algorithm was used for ANN training. This ANN model can be used for sugarcane prediction in India.

It is worth mentioning here that there is a possibility of redefining training patterns, and considering two hidden layers in the network architecture in order to have better performance. The network parameters can further be optimized by considering some more combinations. One cane also explore the possibility of optimizing the weights of ANN models with the help of evolutionary algorithms. One can also explore the possibility of using genetic/fuzzy/neuro-fuzzy/SVM based approaches in order to refine the results reported in this work. The usage of these types of ANN models can be explored for forecasting production of food grains, fruits and agricultural commodities. These models, when implemented efficiently, will increase accuracy in forecasting and will help stakeholders, including farmers, by ensuring optimization of resources.

REFERENCES

[1] A. Chowdhury and M.B. Sarkar, “Estimation of rice yield through weather factors in a dry sub-humid region,” Mausam, vol. 32, no. 4, pp. 393-396, 1981.

[2] Ranjana Agarwal, R.C. Jain, “Forecast of sugarcane yield using eye estimate alongwith plant character,” Biometrical Journal, vol. 38, no.5, pp. 731-39, 1996.

[3] R. Prasad and S.N. Dudhane, “Forecasting mustard yield in gangetic West Bengal using rainfall and agricultural technology,” Mausam, vol. 40, no. 4, pp. 441-6, 1989.

[4] Amrender Kumar and Lalmohan Bhar, “Forecasting model for yield of Indian mustard (Brassica junea) using weather parameter,” Indian journal of agricultural science, vol. 75, no. 10, pp. 688-90, 2005.

[5] B. Cheng, and D.M. Titterington, “Neural networks: A review from a statistical perspective,” Statistical Science., vol. 9, no. 1, pp. 2-54, 1994.

[6] I. Kaastra and M. Boyd, “Designing a neural network for forecasting financial and economic time series,” Neurocomputing, vol, 10, no.3, pp. 215-236, 1996.

[7] E.D. Dewolf and L.J. Francl, “Neural network that distinguish in period of wheat tan spot in an outdoor environment,” Phytopathalogy, vol. 87, pp. 83-87, 1997.

[8] E.D. Dewolf and L.J.Francl, “Neural network classification of tan spot and stagonespore blotch infection period in wheat field environment,” Phytopathalogy, vol. 20, pp. 08-11, 2000.

[9] G. Zhang, B.E. Patuwo, and M.Y.Hu, “Forecasting with artificial neural networks: The state of the art,” International Journal of Forecasting, vol. 14, pp. 35-62, 1998.

[10] S. Chakraborty, R. Ghosh, M. Ghosh, C.D. Fernandes, and M.J. Charchar, "Weather- based prediction of anthracnose severity using artificial neural network models," Plant Pathology, vol. 53, pp. 375- 386, 2004.

[11] J.Gaudart, B. Giusiano, and L. Huiart,"Comparison of the performance of multi-layer perceptron and linear regression for epidemiological data," Comput. Statist. & Data Anal., vol. 44, pp. 547-70, 2004.

[12] D. Park, M. El-Sharkawi, R. Marks, I. Atlas, and M. Damborg, "Electric load forecasting using an artificial neural network," IEEE Trans. Power Syst., vol. 6, pp. 442-49, 1991.

[13] A. Refenes and M. Azema-Baras, "Neural network application in financial asset management," Neural Comput. Applicat., vol. 2, pp. 13-29, 1994.

[14] Neil A. Gershenfeld and Andreas S. Weigend, 1994, "The future of time series." In A.S. Weigend and N.A. Gershenfeld, eds. Time series prediction: Forecasting the future and understanding the past, pp. 1-70. MA: Addison-Wesley, 1994.

[15] Amin F. Atiya and Samir I. Shaheen, "A Comparison between Neural Network Forecasting Technique – Case Study: River Flow Forecasting," IEEE transaction on neural networks. vol. 10, no. 2, pp. 402-409, 1994.

[16] Priti Puri, Maitrei Kohli, "Forecasting student admission in colleges with neural network," International Journal of Computer Science and Network Security, vol. 7, no. 11, pp. 298-303, 2007.

[17] Adesh K. Sharma, R.K. Sharma, and H.S. Kasana, "Empirical comparisons of feed-forward connectionist and conventional regression models for prediction of first lactation 305-day milk yield in Karan Fries dairy cows," Neural Comput & Applic., vol. 15, pp. 359-365, 2006.

[18] Otavio A.S. Carpinteiro, Agnaldo J.R. Treis, and Alexandre P.A. da Silva, "A hierarchical neural model in short-term load forecasting," Journal of Applied Soft Computing., vol. 4, pp. 405-412, 2004.

[19] S. O. Olatunji, Mohammad Saad Al-Ahmadi, Moustafa Elshafei, and Yaser Ahmed Fallatah, "Forecasting the Saudi Arabia stock prices based on artificial neural networks model," International Journal of Intelligent Information Systems., vol. 2, no. 5, pp. 77-86, 2013.

[20] Intan Azmira W. A. R, Rahman T. K. A., Z. Zakaria, and Arfah Ahmad, "Short Term electricity Price Forecasting using Neural Network," in Proc. of the 4th International Conference on Computing and Informatics., pp. 103-108, 2013.

[21] Prakash Ramani, P.D. Murarka, "Stock market prediction using artificial neural network," International journal of advanced research in computer science and software engineering, vol.3, no..4, pp. 873-877, 2013.

[22] A.K. Bhatt and D. Pant, "Automatic apple grading model development based on back propagation neural network and machine vision, and its performance evaluation," AI & Society: Journal of knowledge, culture and Communication, imprint by Springer (AI & Soc DOI 10.1007/s00146-013-0516-5), 2013.

[23] Y.W. Jame and H W. Cutforth, "Crop growth models for decision support systems," Canadian Journal of Plant Science., vol. 76, pp. 9-19, 1996.

[24] W. Stephens, T. Middleton, "Why has the uptake of decision support systems been so poor? Crop-Soil Simulation Models: Applications in Developing Countries." In R. B. Matthews, W. Stephens, eds., pp. 129–147. Wallingford, UK: CAB International, 2002.

[25] Houghton, J.T., L.G. Meira Filho, B.A. Callander, N. Harris, A. Kattenberg and A.,K. Maskell (Eds.), Climate Change 1995: The Science of Climate Change. Contribution of Working Group I to the Second Assessment Report of the Intergovernmental Panel on Climate Change, Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 1996, pp. 572.

[26] H. White, "Learning in artificial neural networks: A statistical perspective," Neural Computation, vol. 1, pp. 425-464, 1989.

[27] B.D. Ripley, "Statistical aspects of neural networks." In Barndroff– Nielsen OE and Jensen JL, eds. Networks and Chaos: Statistical and Probabilistic Aspects, pp. 40-123. London: Chapman and Hall, 1993.

[28] C.M. Bishop, Neural Networks for Pattern Recognition. Oxford: Oxford University Press, 1995.

[29] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, Classification and Regression Trees. Belmont, CA: Wadsworth, 1984.

[30] Y.A. Pachepsky, D. Timlin, G. Varallyay, “Artificial neural networks to estimate soil water retention from easily measurable data,” Soil Science Society of America Journal., vol. 60, pp. 727-733, 1996.

[31] D.A. Elizondo, R.W. Mccledon, and G. Hoogenboom, “Neural network models for predicting flowering and physiological maturity of soyabean,” Transactions of the ASAE, vol. 37, pp. 981-988, 1994.

[32] J. Lawrence, Introduction to Neural Networks, California Scientific Software Press, Nevada City, CA, 1994.

[33] D. Schmueli, “Application of neural networks in transportation planning,” Progress in Planning, vol. 50, pp. 143-201, 1998.

[34] K. Funahashi, “On the approximate realization of continuous mappings by neural networks,” Neural Networks., vol. 2, pp. 183-192, 1989.

[35] K. Hornik, “Some new results on neural network approximation,” Neural Networks., vol. 6, pp. 1069-1072, 1993.

[36] F. Salehi, R. Lacroix, and K.M. Wade, “Improving dairy yield predictions through combined record classifiers and specialized artificial neural network,” Comput Electron Agric., vol. 20, pp. 199-213, 1998.

[37] Reza Ghodsi, Ruzbeh M. Yani, Rane Jalali, and Mahsa Ruzbahman, “Predicting wheat production in Iran using an artificial neural networks approach,” International Journal of Academic Research in Business and Social Sciences., vol. 2, no. 2, pp. 34-47, 2012.

[38] O.O. Obe and D.K. Shangodoyin, “Artificial Neural Network Based Model for Forecasting Sugar Cane Production,” Journal of Computer Science., vol. 6, no. 4, pp.439-445, 2010.

[39] Ratna Raj Laxmi and Amrender Kumar, “Weather based forecasting model for crops yield using neural network approach,” Statistics and Applications, vol.9, pp. 55-69, 2011.

[40] B.Ji, Y. Sun, S. Yang and J. Wan, “Artificial neural networks for rice yield prediction in mountainous regions,” Journal of Agricultural Science., vol. 145, no. 3, pp. 249-261, 2007.

[41] Adesh K. Pandey, A.K. Sinha, V.K. Srivastava, “A Comparative Study of Neural-Network & Fuzzy Time Series Forecasting Techniques – Case Study: Wheat Production Forecasting,” Journal of Computer Science and Network Security, vol. 8, no. 9, pp. 382-387, 2008.

[42] Sangram S. Nikam, A.K. Mishra, A. Sarangi, Paresh B. Shirsath, D.K. Singh, and V. Ramasubramanian, “Artificial Neural Network Models to predict Wheat Crop Evapotranspiration,” Journal of Agricultural Engineering., vol. 47, no. 2, pp. 20-25, 2010.

[43] Rama Krishna Singh and Prajneshu, “Artificial Neural Network Methodology for Modelling and Forecasting Maize Crop Yield,” Agricultural Economics Research Review, vol. 21, pp. 5-10, 2008.

[44] Sunil Kumar, Vivek Kumar and R. K. Sharma,” Artificial Neural Network Based Models for Rice Yield Forecasting,” International Journal of Computational Intelligence Research, vol. 10, no. 1, pp. 73-90, 2014.

[45] Agricultural Statistics At a Glance 2010. Directorate of Economics and Statistics, Department of Agriculture and Cooperation, Ministry of Agriculture, Government of India. http://eands.dacnet.nic.in/At_Glance_2010/4.23(A).xls, May 21, 2014.

Authors

Sunil Kumar received the MCA degree from Indira Gandhi National Open University, New Delhi and pursuing Ph.D. from Gurukula Kangri Vishwavidyalaya, Haridwar, India. His research interests include machine learning and computer networks.

Vivek Kumar has completed his Master degrees in Applied Mathematics (1985-1987) and Computer Applications (1987-1988) from University of Roorkee (now IIT), Roorkee , India and Ph.D. from Gurukul Kangri University, Haridwar, India. His research interests include Machine Learning, distributed computing and wireless adhoc networks.

Rajendra Kumar Sharma received his Ph.D. degree in mathematics from University of Roorkee, Roorkee (Now, IIT Roorkee), India in 1993. He is currently working as a Professor at Thapar University, Patiala, India, where he teaches, among other things, statistical models and their usage in computer science. He has been involved in the organization of a number of conferences and other courses at Thapar University, Patiala. His main research interests are statistical models in computer science, Neural Networks, and Pattern Recognition.