DOI: http://dx.doi.org/10.26483/ijarcs.v8i7.4537

ISSN No. 0976-5697

Volume 8, No. 7, July – August 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

SOIL N-P-K PREDICTION USING LOCATION AND CROP SPECIFIC RANDOM FOREST CLASSIFICATION TECHNIQUE IN PRECISION AGRICULTURE

Mr. Ambarish G. Mohapatra
Ph.D. Research Scholar
Suresh Gyan Vihar University, Jaipur, India

Dr. Bright Keswani
Department of Computer Applications
Suresh Gyan Vihar University, Mahal Jagatpura, Jaipur, India.

Dr. Saroj Kumar Lenka
Department of Information Technology
Mody University, Lakshmangarh, 332311, Rajasthan, India

Abstract : Agriculture is a basic and most important profession of any country as it balances the food requirement and also the essential raw materials of industry. In the similar sense, the adaptation of implementing smart technology in agriculture practices needs to be focused on better land productivity. The fertilizer and manure should be precisely applied during agriculture practice throughout the countryside with the use of digital technologies. This motivates us to develop a reactive web application which will accurately predict the required soil N-P-K (Nitrogen-Phosphorus-Potassium) content by utilizing one time soil testing results of available soil N-P-K contents as per the yield target. This predictive model is designed by considering standard experimental data sets from Indian Council for Agriculture Research (ICAR), India. The prediction model is designed using Random Forest (RF) algorithm which is capable of handling large dataset. The predicted N-P-K content are shown in a reactive R Shiny user interface to notify required N-P-K values for the necessary action by the farmer. The complete web based prediction model is efficiently conveying the required N-P-K content information of the particular farm location to the farmer as well as the agriculture specialists.

Keywords: Precision Agriculture; Soil N-P-K; Prediction Model; Random Forest Algorithm; Web Application; R Shiny

1. INTRODUCTION

The agriculture plays a dominant role in the growth of the country's economy. The agriculture land productivity needs to be improved for maintaining food requirement by utilizing smart techniques [4, 5, 25]. In the similar context, precision agriculture is also called site-specific-agriculture. The farmers are able to manage large fields as they are a group of small fields. This can be achieved by utilizing smart management techniques through information and communication technology (ICT). The ICT possesses the capability to connect the world globally and it is also influencing our lifestyle and social consciousness. Now we can explore the possible contribution of information science for efficient and stable production of agriculture commodities by utilizing models like crop growth prediction, decision support system, mathematical and statistical modeling, and the development of software packages for agriculture research. In this work, we are focusing on the prediction of soil N-P-K content by utilizing random forest based algorithm along with the development of a dynamic web interface to assist the farmers and agriculture specialist. German scientist Justus Von Liebig proposed the theory that nitrogen (N), phosphorous (P), and potassium (K) contents are the essential components for the crop growth. Furthermore, other essential soil components like sulphur, hydrogen, oxygen, carbon, magnesium are also very much responsible for better crop health. Other nutrients like hydrogen, carbon, sodium, magnesium, copper, oxygen, molybdenum, cobalt, boron, sulphur, and zinc are also essential for the plant health as N-P-K [1, 3]. Therefore to track the soil N-P-K content of every farm land the agriculture departments have to handle the huge volume of data. The latest technological challenge is to handle very large agriculture data in a real-time environment. This is a quite tedious task for handling these data manually.

The latest technological challenge is to handle very large agriculture data in a real-time environment. This is a quite tedious task for handling these data manually. In this work, we proposed a novel technique to handle these high volume contents using cloud computing architecture as well as data analytics. The handling of data will not solve all the problems like prediction of specific requirement, analysis of utilization of the specific element. Therefore, data analytics [12, 13] is also required to achieve these targets.

The above aspects encourage us to model a web application which can be utilized to predict required soil N-P-K content. This will help the agriculture scientists to take quick decisions for assisting the farmers for the application of necessary nutrients in the agriculture land. And also, it will prevent over use of nutrients in the farm land. Therefore, we have used R Shiny framework for developing a smart web application which can solve many issues during the entire course of farming [11, 23]. This proposed work is based on the accurate prediction of soil N-P-K requirement which can be easily accessed by farmers in any location and at any time. The preliminary knowledge of soil N-P-K availability, soil type and crop type are used to predict the required N-P-K content. The remaining sections of the article explain the complete architecture of our proposed model. Section 2 explains the complete methodology used to achieve this objective. In the similar context, section 3

shows all the experimental results obtained during testing of the dynamic web application.

2. METHODOLOGY

Engineers and scientists are always utilizing high performance computational tools to perform complex simulations regarding large agriculture datasets. In this work, we have used experimental NPK dataset from ICAR [1, 2] to develop random forest based decision tree model for the prediction of required NPK for the crop and soil specific agriculture land. The dataset attributes along with few sample data are listed in the table 1.

Table 1. Few dataset used for the prediction of soil N-P- K requirement

Soil Type	Crop Type	Available N Content (Kg/ha)	Available P Content (Kg/ha)	Available K Content (Kg/ha)	Yield Target (q/ha)	Required N Content (Kg/ha)	Required P Content (Kg/ha)	Required K Content (Kg/ha)
Vertisols	Rice-Mashuri	150	5	150	50	115	144	52
Vertisols	Rice-Mashuri	150	5	150	55	133	144	60
Inceptisols	Wheat	250	5	100	45	122	73	89
Inceptisols	Wheat	250	5	100	50	137	80	100
Calcareous soil	Maize	120	4	60	30	90	69	41
Calcareous soil	Maize	120	4	60	40	137	95	61
Cotton	Typic Haplusterts	100	6	250	20	187	119	126
Cotton	Typic Haplusterts	100	6	250	24	239	146	160
Note: The complete data are collected from Indian Council for Agriculture Research (ICAR) experimental results [2]								

The proposed model is such customised that the farmer can easily send available soil N-P-K content as well as the crop and soil variety in the web application portal. The input data by the farmer will be passed to R analytics server and simultaneously the experimental dataset will be acquired from the external web server [9, 10]. The experimental datasets are used to develop random forest based classification tree in the R server. After successfully training the dataset, the farmer's input will be used to predict the required N-P-K content of the particular location. The complete architecture of our proposed web based prediction model is shown in the fig.1. In the subsequent sections, the architecture of random forest algorithm is explained with complete flow diagram. During the complete development we have adapted the R Shiny functionality [15] to test our prediction and classification model using two basic components such as ui.R and server.R [19, 20]. The request and response between the client and server side model is shown in the fig. 2.

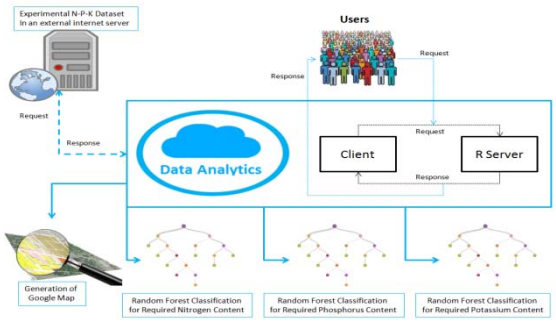

Fig. 1. The complete architecture of the proposed web application model

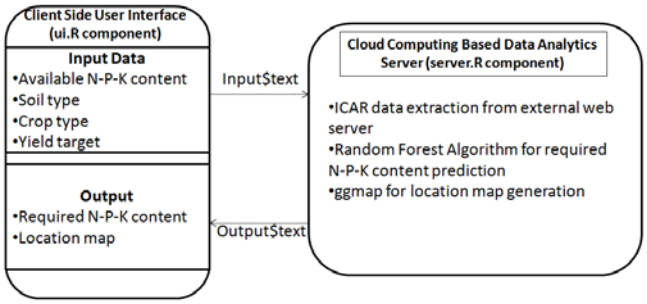

Fig. 2. The input and output instances used in R-Shiny environment

2.1. Random Forest Algorithm

Random Forest (RF) algorithm was proposed by Breiman (2001) which is a collection of many tree predictors where the trees are constructed using various random features [24]. Therefore, the name of the algorithm is taken as "Random Forest (RF)". The Random forest (RF) algorithm is a Classification and Regression Tree framework which is also depicted as CART model. Random vectors are generated to represent the growth of trees where the trees are never being pruned [22]. A random combination of features is selected at each and every node to perform splitting. Breiman introduces a new technique called Bootstrap Aggregating or Bagging to select random features and monitor error in the algorithm [22]. The Bagging method claims to increase the accuracy of the random forest algorithm by minimizing the generalization error for the ensemble trees owing to the use of random features [21]. This generalization error estimates are done using Out of Bag (OOB) method. This bootstrapping procedure leads to improve the model performance because it also decreases the variance of the model without increasing the bias [29]. It signifies that the predictions of a single tree are highly sensitive to noise inside the training set while the average of

many trees is not correlated [21, 22]. For a training set of dataset (D) such as $[(x_1, y_1)\ldots (x_n, y_n)]$ and the dimension of x_i is d, bagging repeatedly (suppose B times) selects a random sample with replacement of the training set and fits trees to these samples [29].

For i = 1... B:
Choose bootstrap sample D_i from D;
Construct decision tree T_i using D: such that at each node, choose a random subset of the features and only consider splitting those features.
End
For given x, consider the majority vote (for class) or average (for regression).

The predictions for unseen samples (x') can be developed by averaging the predictions from all the individual regression trees on x' according to the equation 1.

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} \hat{f_b}(x') \quad \ldots (1)$$

The random forest algorithm is also suited for wide range of datasets [26, 27]. It is a collection of many classification and regression trees. The sum of the prediction made from every decision trees is used to determine the overall prediction of the target sets [27]. That is why it is suitable for the analysis of complex data structures consisting of small or medium data sets and having less size but a very large set of columns [28].

3. RESULTS AND DISCUSSIONS

The proposed random forest model is successfully implemented and it is integrated with R Shiny framework to display the predicted required soil N-P-K content. The web interface is tested using four types of soil such as vertisols, sandy loam, alluvial and black soil. The crop types included in the web application are Rice-Mashuri, Rice-Pothana, Rice-MTU-2067 and Rice-MTU-5182. The dynamic web application runs three prediction algorithms such as required N, P and K inside server.R program. The frequency of variations in available N-P-K content, yield target are shown in the fig. 3, fig. 4, fig. 5 and fig. 6 respectively. It can be observed from the frequency of variations plots in the fig. 3, fig. 4, fig. 5 and fig. 6 that many N-P-K contents, as well as the yield target values, have similar frequency values.

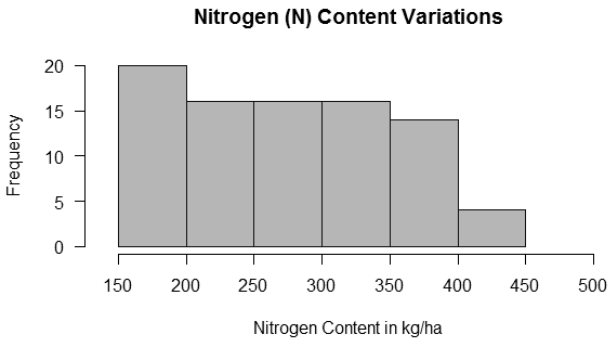

Fig. 3. Frequency of variations in available nitrogen content

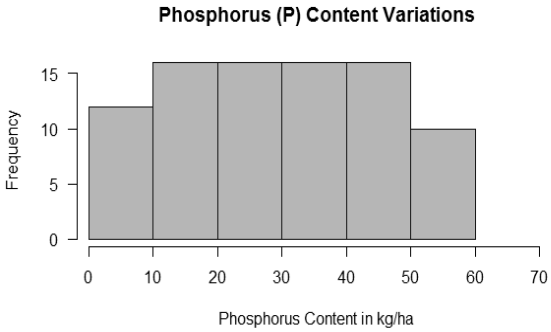

Fig. 4. Frequency of variations in available phosphorus content

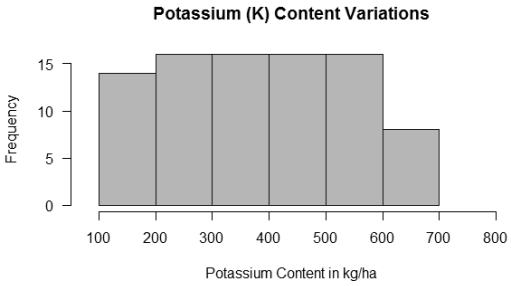

Fig. 5. Frequency of variations in available potassium content

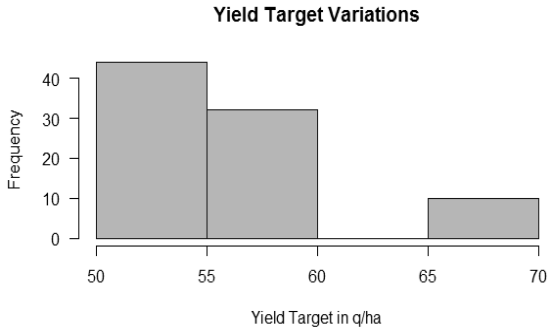

Fig. 6. Frequency of variations in yield target

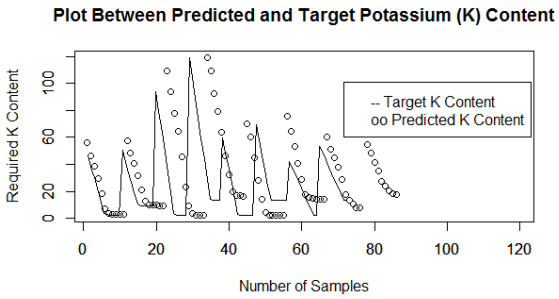

Fig. 7. Predicted and target potassium (K) content using RF

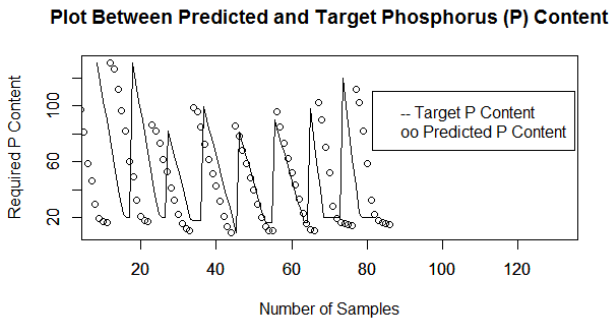

Fig. 8. Predicted and target phosphorus (P) content using RF

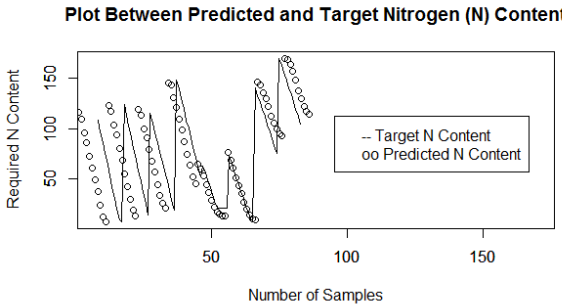

Fig. 9. Predicted and target nitrogen (N) content using RF

The predicted required N-P-K contents are plotted against target N-P-K content as in the fig. 7, fig. 8 and fig. 9. It can be observed from the fig. 7, fig. 8 and fig. 9 that the predicted required amount of soil N-P-K content has minimum deviation from the target N-P-K levels. The RMSE (Root Mean Square Error) obtained during neural network training in R shiny environment are listed in the table 2.

Table 2. RMSE obtained during the prediction of soil N-P-K required

	Required N content	Required P content	Required K content
RMSE	6.118521	5.195799	4.710358

The RMSE minimization Vs the number of trees obtained for required nitrogen (N), phosphorus (P) and potassium (K) content during the prediction of N-P-K contents are shown in the fig. 10, fig. 11 and fig. 12. From the RMSE minimization plots as shown in the fig. 10, fig. 11 and fig. 12 it can be observed that the minimum RMSE is obtained at about 100, 100 and 50 number of trees for nitrogen (N), phosphorus (P) and potassium (K) prediction respectively.

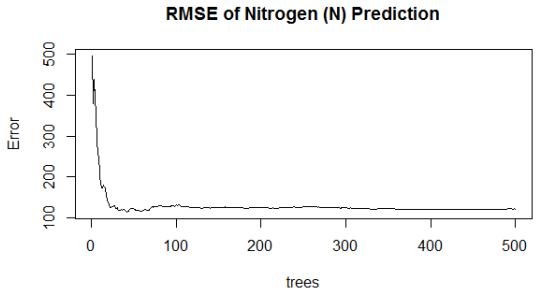

Fig. 10. RMSE obtained during nitrogen (N) prediction using RF

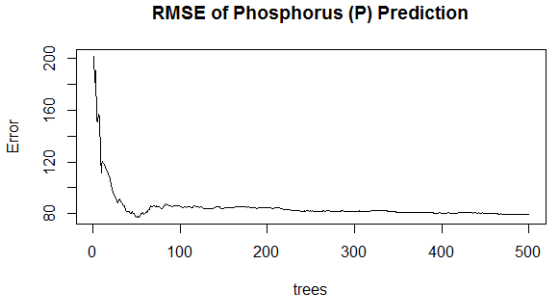

Fig. 11. RMSE obtained during phosphorus (P) prediction using RF

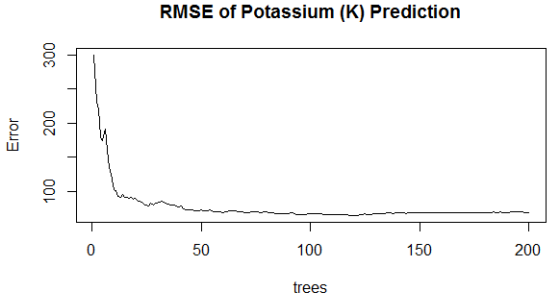

Fig. 12. RMSE obtained during potassium (K) prediction using RF

Table 3. Configuration of Random Forest (RF) model for required soil N-P-K content prediction

	RF configuration
Nitrogen (N) Content Prediction	Number of trees: 500 No. of variables tried at each split: 3 Mean of squared residuals: 122.0516 % Variance explained: 94.9
Phosphorus (P) Content Prediction	Number of trees: 500 No. of variables tried at each split: 3 Mean of squared residuals: 79.70919 % Variance explained: 94.28
Potassium (K) Content Prediction	Number of trees: 200 No. of variables tried at each split: 3 Mean of squared residuals: 69.71243 % Variance explained: 92.52

The structure of the random forest model developed during the classification and regression of required soil N-P-K content are shown in the table 3. The percentage of variance for nitrogen, phosphorus and potassium content prediction is listed in the table 3. It can be observed that the predicted and target contents have fewer variations.

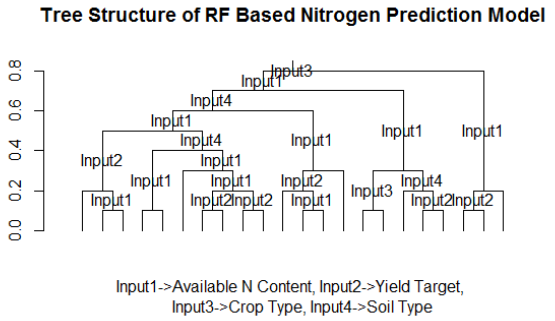

Fig. 13. Random Forest Tree model of nitrogen content prediction

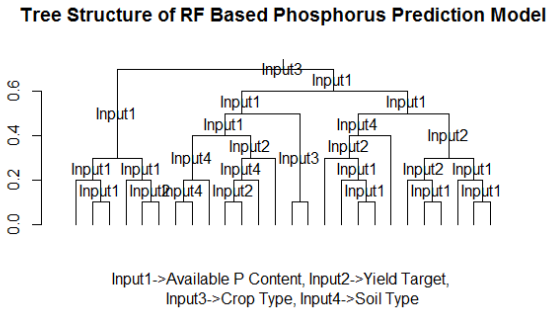

Fig. 14. Random Forest Tree model of phosphorus content prediction

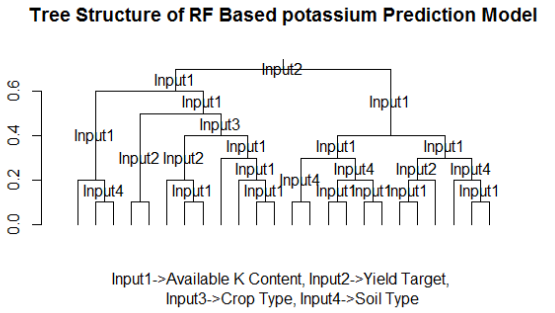

Fig. 15. Random Forest Tree model of potassium content prediction

Finally, the random forest model is integrated with R Shiny framework and a location map (Google Map) of the particular location is shown on the R Shiny user interface using R ggmap package [13, 17, 18]. The location of the particular place is obtained from google locations such as latitude and longitude. A reactive function is used to make the google map as a dynamic web interface [16]. It is performed using R Shiny reactive ({function ()}) method. An example of the location maps is shown in the fig. 16. Shiny is an R package that makes it easy to build interactive web applications (apps) straight from R [7, 8]. There are two types of graphics functions in R: functions based on the default graphics package and functions based on the grid graphics package [6, 14]. The complete web application is shown in the fig. 17.

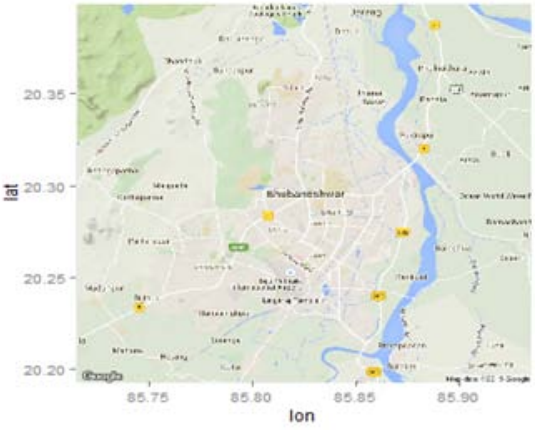

Fig. 16. Location maps of Bhubaneswar using ggmap integrated with web application

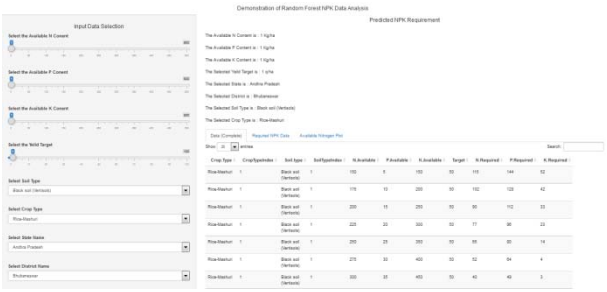

Fig. 17. Complete schematic of the web application [Random Forest algorithm is running in the R Server]

4. CONCLUSION

The random forest algorithm for real-time N-P-K prediction model is successfully implemented, and it is integrated with R Shiny reactive web application framework. The random forest classification and regression model is configured with minimum RMSE. The reactive web application is developed with R shiny which is easy to use, secure and scalable. In this article, only CART was used for the determination of required soil N-P-K content. The proposed model provides a better accessibility to determine required soil N-P-K content by utilizing one time soil testing data like available soil N-P-K content, soil type, crop type and yield target. This will help the farmers and agriculture specialists to estimate the required N-P-K content without using manual calculations. The proposed algorithm can also be modeled with very high volume dataset of various crops, N-P-K availability, yield targets, soil types and regions.

5. REFERENCES

[1]. Agriculture in India - of Planning Commission, Link: http://www.planningcommission.nic.in/reports/sereport/ser/vision2025/agricul.doc

[2]. STCR Crop Wise Recommendations, Link: http://www.iiss.nic.in/downloads/stcrCropwiseRecommendations.pdf

[3]. Jage Singh, S.C. Khurana. Department of Vegetable Science, CCS Haryana Agricultural University, Hisar, 125 004,

Haryana, India, Productivity of Paddy-Potato-Wheat System in Haryana, Potato J. 2005; 32 (3-4): 171-172.
[4]. Paul Murrell, Simon Potter. The gridSVG Package, The R Journal. June 2014; 6(1): 133-143.
[5]. RStudio Inc. shiny: Web Application Framework for R.
[6]. Link: http://CRAN.R-project.org/package= shiny. R package version 0.3.0, 2013.
[7]. D. Sarkar. Lattice: Multivariate Data Visualization with R. Springer-Verlag, New York.
[8]. Link: http://lmdvr.r-forge.r-project.org. ISBN 978-0-387-75968-5, 2008
[9]. Ramnath Vaidyanathan. Interactive Visualizations with rCharts and Shiny.
[10]. Link: http://ramnathv.github.io/rChartsShiny/, 2013.
[11]. H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag, New York; 2009,
[12]. ISBN 978-0-387-98140-6, Link: http://www.springer.com/in/book/9780387981406.
[13]. Y. Xie. Animation: An R package for creating animations and demonstrating statistical methods. Journal of Statistical Software; 2013, 53:1–27, Link: http://www.jstatsoft.org/v53/i01.
[14]. M. Gesmann and D. de Castillo. googleVis: Interface between R and the Google Visualisation API. The R Journal; December 2011, 3(2):40–44, Link: http://journal.r-project.org/archive/2011-2/RJournal_2011-2_Gesmann+de~Castillo.pdf.
[15]. Google. Google Visualization API; 2013,
[16]. Link: https://developers.google.com/chart/interactive/docs/reference.
[17]. Richard Newton, Andrew Deonarine, Lorenz Wernisch. Creating web applications for spatial epidemiological analysis and mapping in R using Rwui, Source Code for Biology and Medicine; 2011, 6 (6): 1-5.
[18]. Newton R, Wernisch L, Rwui: A web application to create user friendly web interfaces for R scripts. R News; 2007, 7(2):32-35.
[19]. Google Static Maps API. Link: http://code.google.com/apis/maps/documentation/staticmaps.
[20]. Roberto García , Juan Manuel Gimeno, Ferran Perdrix, Rosa Gil, Marta Oliva, Juan Miguel López, Afra Pascual, Montserrat Sendín. Building a Usable and Accessible Semantic Web Interaction Platform, World Wide Web; March 2010, 13 (1): 143-167.
[21]. Roberto García , Juan Manuel Gimeno, Ferran Perdrix, Rosa Gil, Marta Oliva, Juan Miguel López, Afra Pascual, Montserrat Sendín. Building a Usable and Accessible Semantic Web Interaction Platform, World Wide Web; March 2010, 13 (1): 143-167.
[22]. Breiman, Leo. Random Forests, Machine Learning; 2001, 45 (1): 5–32.
[23]. S.Thenmozhi, P.Thilagavathi. Impact of Agriculture on Indian Economy, International Research Journal of Agriculture and Rural Development; December 2014, 3 (1): 96-105.
[24]. D. Vitorino, S.T.Coelho, P.Santos, S.Sheets, B.Jurkovac, C.Amado. A random forest algorithm applied to condition based wasterwater deterioration modeling and forecasting, 16th conference on water distribution system analysis (WDSA), Procedia Engineering; 2014, 89:401-410.
[25]. Martin Junga,Susanne Tautenhahna, Christian Wirthb, Jens Kattgea. Estimating basal area of spruce and fir in post-fire residual stands in Central Siberia using Quickbird, feature selection, and Random Forests, International Conference on Computational Science (ICCS), Procedia Computer Science; 2013, 18:2386-2395.
[26]. Manfred Kratzenberga, Hans Helmut Zürna, Pal Preede Revheimb, Hans Georg Beyerb. Identification and handling of critical irradiance forecast errors using a random forest scheme – a case study for southern Brazil, European Geosciences Union General Assembly (EGU), Energy Procedia; 2015, 76:207-215.
[27]. Ashutosh Patri, Yugesh Patnaik. Random forest and stochastic gradient tree boosting based approach for the prediction of airfoil self-noise, International Conference on Information and Communication Technologies (ICICT), Procedia Computer Science; 2015, 46:109-121.

