IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 10, Issue 4, April 2022

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.165

9940 572 462 6381 907 438 ijircce@gmail.com www.ijircce.com

| DOI: 10.15680/IJIRCCE.2022.1004151|

Random Forest Algorithm : A Supervised Machine Learning Technique

Om Prakash Thakur, Kanhaiya Lal, Sonia Wadhwa

BE Scholar, Department of Computer Science and Engineering, Government Engineering College, Bilaspur (C.G.), India

BE Scholar, Department of Computer Science and Engineering, Government Engineering College, Bilaspur (C.G.), India

Assistant Professor, Department of Computer Science and Engineering, Government Engineering College, Bilaspur (C.G.), India

ABSTRACT: -Random Forest is an ensemble supervised machine learning technique. Machine learning techniques have applications in the area of Data mining. Random Forest has tremendous potential of becoming a popular technique for future classifiers because its performance has been found to be comparable with ensemble techniques bagging and boosting. Hence, an in-depth study of existing work related to Random Forest will help to accelerate research in the field of Machine Learning. This paper presents a systematic survey of work done in Random Forest area. In this process, we derived Taxonomy of Random Forest Classifier which is presented in this paper. We also prepared a Comparison chart of existing Random Forest classifiers on the basis of relevant parameters. The survey results show that there is scope for improvement in accuracy by using different split measures and combining functions; and in performance by dynamically pruning a forest and estimating optimal subset of the forest. There is also scope for evolving other novel ideas for stream data and imbalanced data classification, and for semi-supervised learning. Based on this survey, we finally presented a few future research directions related to Random Forest classifier.

KEYWORDS: random forest, decision tree, regression, classification, supervised machine learning.

I.INTRODUCTION

Random Forest is a relatively new Ensemble Supervised Machine Learning approach. Data mining approaches can benefit from machine learning techniques. Descriptive and predictive data mining are two types of data mining. Descriptive data mining is more concerned with explaining the data, categorising it into categories, and summarising it. Predictive data mining examines historical data and generates trends or conclusions that can be used to forecast the future. Predictive data mining has its roots in the traditional statistical model-building procedure.The feature analysis of predictor variables is used to develop predictive models. One or more characteristics are thought to be predictive. The hypothesis is a function of the predictors, and the output is that function. Acceptance or rejection of the hypotheses generated is tested. The accuracy of this model is determined using a variety of error estimation techniques. Unsupervised machine learning techniques are typically used for descriptive data mining, while supervised machine learning techniques are used for predictive data mining.Labeled data samples are used in supervised machine learning to classify samples into multiple categories. A predictive model is trained on a dataset. The test dataset is used to estimate the model's accuracy.A decision tree is a popular supervised machine learning approach.The decision tree is used as the foundation classifier in Random Forest. Random Forest builds many decision trees, with randomization occurring in two ways: (1) random data sampling for bootstrap samples, as in bagging, and (2) random feature selection for constructing individual base decision trees. The generalisation error of a Random Forest classifier is determined by the strength of each individual decision tree classifier and the correlation among base trees.The Random Forest classifier's accuracy is comparable to that of existing ensemble techniques such as bagging and boosting. Random Forest, according to Breiman, runs efficiently on large databases, can handle thousands of input variables without variable deletion, provides estimates of important variables, generates an internal unbiased estimate of generalisation error as the forest grows, has an effective method for estimating missing data and maintaining accuracy when a large proportion of data is missing, and has methods for balancing class error in unbalanced data sets.Random Forest's intrinsic parallel nature has led to multithreading, multi-core, and parallel architectures being used to implement it.

Because of the qualities outlined above, Random Forest is used in many modern classification and prediction applications. Rather than researching and dissecting the theoretical underpinning of the Random Forest classifier in depth, we focused on practical research in this paper.

II. LITERATURE REVIEW

Ensemble Classifiers

A set of individually trained classifiers makes up an ensemble (such as neural networks or decision trees). For classifying fresh cases, their predictions are pooled. According to previous research, an ensemble is often more accurate than any of the individual classifiers in the ensemble. Bagging and boosting are two prominent ensemble-creation techniques. To get separate training sets for each of the classifiers, these methods employ re-sampling strategies. Bootstrap aggregation, often known as bagging, is based on the concept of bootstrap samples. If the original training dataset is N by N and m individual classifiers are to be constructed as part of the ensemble, sampling with replacement is used to generate m different training sets, each of size N. Bagging generates numerous classifiers that are independent of one another. Weights are assigned to each sample from the training dataset in the case of boosting. If there are m classifiers to be created, they are created in order so that one classifier is created every iteration. Weights of training samples are updated based on classification results of classifier Ci-1 to generate classifier Ci. Boosting produces classifiers that are mutually reliant. An ideal ensemble, according to theoretical and practical studies on ensemble, consists of highly correct classifiers that disagree as little as feasible. The generalizability of such ensembles was empirically proven by Opitz and Shavlik. Bagging is successful for unstable learning algorithms, according to Breiman. Kuncheva describes four methods for assembling ensembles of different classifiers:

1. Combination level: Design different combiners.
2. Classifier level: Use different base classifiers.
3. Feature level: Use different feature subsets.
4. Data level: Use different data subsets.

Random Forest

Definition: Random Forest is a classifier consisting of a collection of tree-structured classifiers {h(x, Θk) k=1, 2, ….}, where the {Θk } are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x .

Random Forest produces a collection of decision trees. Breiman used the randomization approach, which works well with bagging or random subspace approaches, to achieve variation among base decision trees. Breiman used the methods below to create each tree in Random Forest: If the training set has N records, then N records are picked at random from the original data, with replacement. This is known as a bootstrap sample. This sample will serve as the tree's training set. If there are M input variables, a number m<< M is chosen so that m variables are chosen at random from M at each node, and the best split on these m characteristics is used to split the node. During forest growth, the value of m is kept constant. Each tree is grown to its maximum potential. Pruning is not an option.

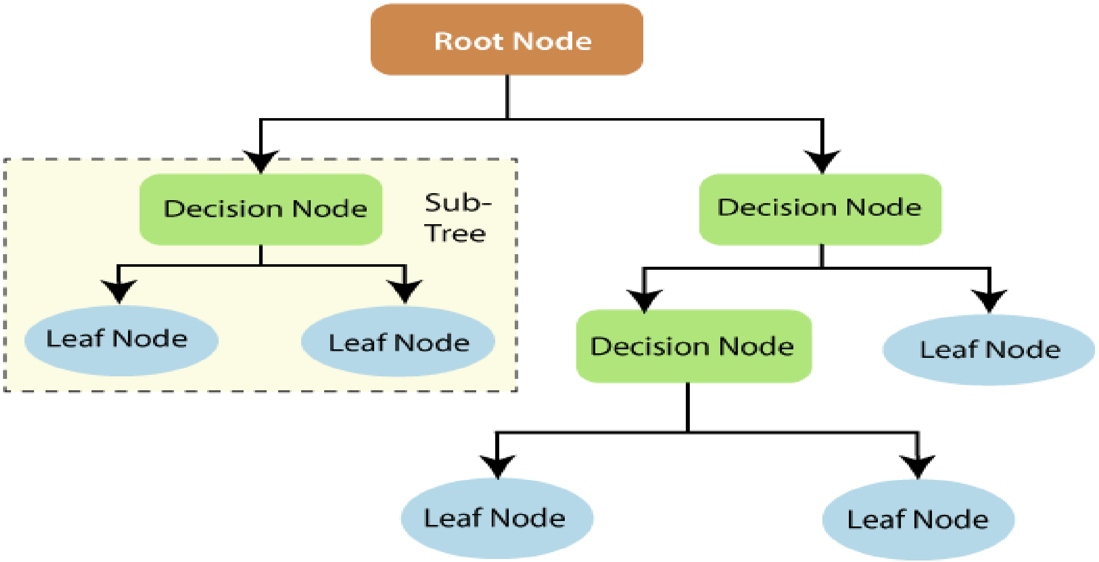
Multiple trees are induced in the forest in this fashion, with the number of trees determined by the parameter N tree. In the literature, the number of variables (m) selected at each node is also referred to as mtry or k. The nodesize (number of instances in the leaf node) parameter, which is generally set to one, controls the tree's depth. To classify a new instance after the forest has been trained or developed as described above, it is run across all of the trees growing in the forest. Each tree assigns a categorization to a new instance, and this classification is recorded as a vote. The votes from all trees are added together, and the class with the most votes (majority voting) is proclaimed the new instance's categorization.Forest RI is the name given to this process in the literature. Random Forest refers to the forest of decision trees built using the Forest RI technique from now on. When a bootstrap sample set is produced by sampling with replacement for each tree in the forest-building process, around 1/3 of the original cases are left out. OOB (Out-of-Bag) data refers to this collection of instances. Each tree has its own OOB data collection, which is used to estimate the error of each tree in the forest, a process known as OOB error estimation. The Random Forest method also includes a function for calculating variable relevance and proximity. Missing values and outliers are replaced using proximities.
Illustrating Accuracy of Random Forest:

The Generalization error (PE*) of Random Forest is given as,
PE * = P x,y (mg(X,Y)) < 0.
The Margin function is mg (X, Y). The Margin function determines how much the average number of votes for the right class at (X, Y) outnumbers the average vote for any other class. The predictor vector is X, and the classification is Y.
The Margin function is given as,
mg (X, Y) = avk I(hk (X) = Y) – max j≠Y avk I(hk (X) = j)
Here I(.) is Indicator function.
The margin is proportional to the classification's confidence. The expected value of the Margin function is expressed as *S = E X, Y (mg (X, Y))* for Random Forest.
The generalization error of ensemble classifier is bounded above by a function of mean correlation between base classifiers and their average strength (s). If ρ is mean value of correlation, an upper bound for generalization error is given by,
PE ≤ ρ (1 – s2) / s2*

III. METHEDOLOGY

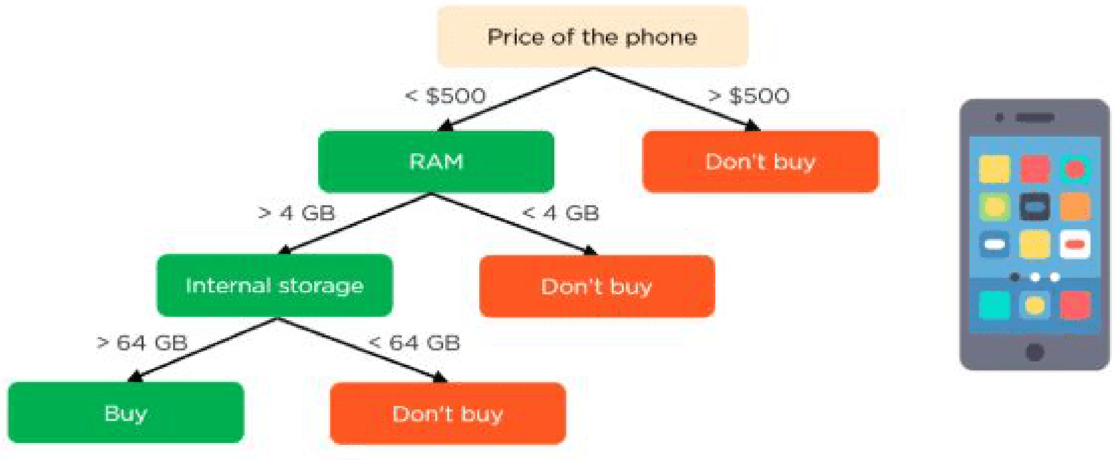
Understanding Dicision Tree

A random forest algorithm's building components are decision trees. A decision tree is a decision-making tool with a tree-like structure. A basic understanding of decision trees will aid our understanding of random forest algorithms. There are three parts to a decision tree: decision nodes, leaf nodes, and a root node. A decision tree method separates a training dataset into branches, each of which is further divided into branches. This pattern repeats until a leaf node is reached. The leaf node cannot be further separated. The attributes utilised to forecast the outcome are represented by the nodes in the decision tree. The leaves are connected to the decision nodes. The three types of nodes in a decision tree are depicted in the diagram below.



More information on how decision trees work can be found in information theory. The basic blocks of decision trees are entropy and information gain. A basic comprehension of these principles will help us better comprehend how decision trees are constructed. The measure entropy is used to calculate uncertainty. Given a set of independent variables, information gain is a measure of how much uncertainty in the target variable is decreased

The notion of information gain entails employing independent variables (features) to learn about a target variable (class). The information gain is calculated using the entropy of the target variable (Y) and the conditional entropy of Y (given X). The conditional entropy is deducted from the entropy of Y in this scenario. In the training of decision trees, information gain is used. It contributes to the reduction of uncertainty in these trees. A high degree of uncertainty (information entropy) has been reduced with a substantial information gain. Splitting branches, a fundamental action in the creation of decision trees, requires entropy and information gain. Let's look at how a decision tree works in practise. Let's say we want to anticipate whether a customer will buy a phone or not. His judgement is based on the phone's specifications. A decision tree diagram can be used to convey this analysis. The features of the phone are represented by the root node and decision nodes of the decision. The final output, whether buying or not buying, is represented by the leaf node. The pricing, internal storage, and Random Access Memory are the primary factors that influence the decision (RAM). The decision tree will look like this.



Applying decision tree in randon forest

The fundamental distinction between the decision tree and the random forest algorithms is that the latter randomly establishes root nodes and segregates nodes. The bagging method is used by the random forest to generate the required forecast. Instead of using just one sample of data, bagging requires using multiple samples (training data). A training dataset is a collection of observations and attributes used to make predictions. Depending on the training data provided to the random forest algorithm, the decision trees produce varied results. These results will be ranked, with the highest ranking being chosen as the final result. Our original example can still be utilised to demonstrate the operation of random forests. The random forest will have several decision trees instead of a single decision tree. Assume we only have four decision trees.

The training data, which includes the phone's observations and features, will be separated into four root nodes in this example. The root nodes may symbolise four characteristics that may impact a customer's decision (price, internal storage, camera, and RAM). The random forest splits the nodes by randomly picking features. The outcome of the four trees will determine the final prediction. The final choice will be determined by most decision trees. The final forecast will be buying if three trees predict buying and one tree predicts not buying. The customer is expected to purchase the phone in this situation.

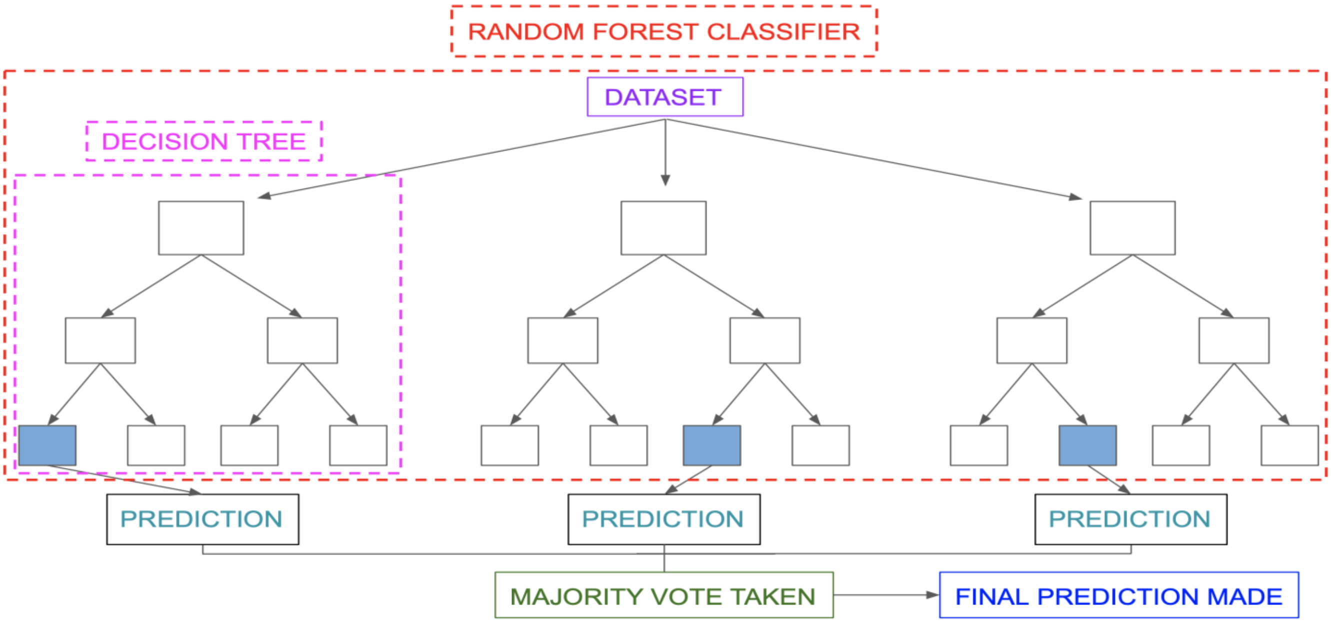
Classification In Random Forest

Random forest classification uses an ensemble methodology to achieve the desired result. Various decision trees are trained using the training data. This dataset contains observations and features that will be randomly selected during node splitting. Various decision trees are used in a rain forest system. There are decision nodes, leaf nodes, and a root node in every decision tree. The final output provided by each decision tree is represented by the leaf node. The final product is chosen using a majority-voting procedure. In this situation, the final output of the rain forest system is the output chosen by the majority of decision trees. A simple random forest classifier is shown in the diagram below.

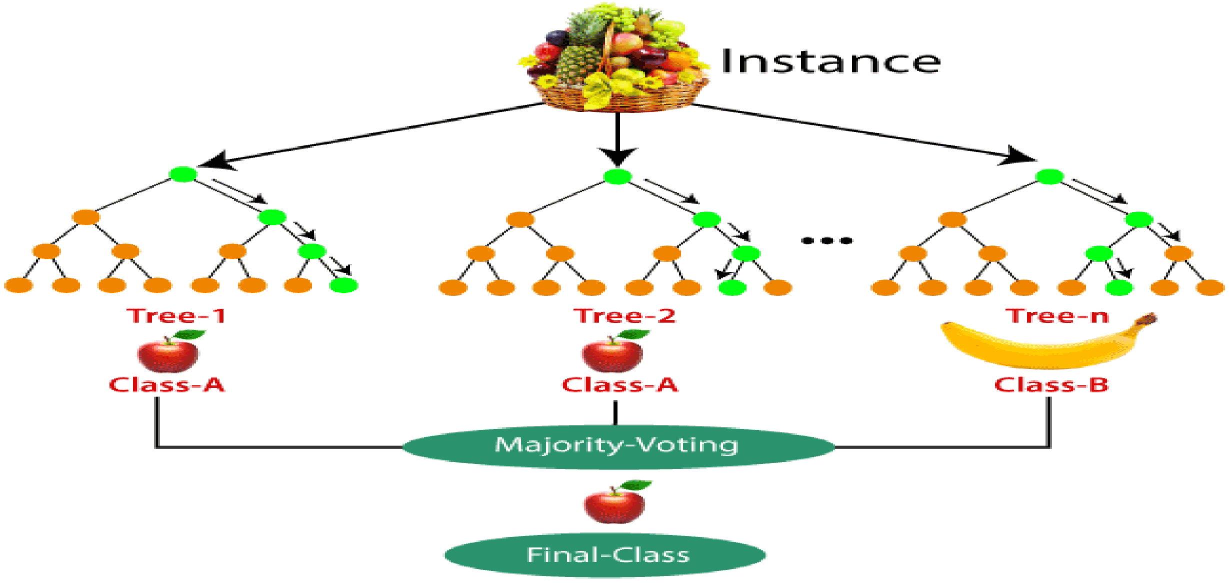
| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 4, April 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1004151|



Take, for example, a training dataset that includes bananas, apples, pineapples, and mangoes. This dataset is divided into subgroups by the random forest classifier. Every decision tree in the random forest system is given these subsets. Each decision tree generates its own output. For example, the apple is predicted for trees 1 and 2. Another decision tree (n) indicated that the outcome would be banana. The final prediction is made using the majority voting collected by the random forest classifier. The apple has been chosen as the forecast by the majority of decision trees. As a result, the classifier selects apple as its final prediction.



Regression In Random Forest

The other duty that a random forest algorithm does is regression. Simple regression is followed by a random forest regression. The random forest model passes the values of dependent (features) and independent variables. Random forest regressions can be performed in a variety of tools, including SAS, R, and Python. Each tree in a random forest regression makes a unique prediction. The regression's output is the average prediction of the individual trees. In contrast, the result of random forest classification is dictated by the decision trees' class mode. Although the concepts of random forest regression and linear regression are similar, their functions differ. y=bx + c is the formula for linear regression, where y is the dependent variable, x is the independent variable, b is the estimation parameter, and c is a constant. A sophisticated random forest regression's function is similar to that of a blackbox.

IV. CONCLUSION

The rain forest method is a simple and flexible machine learning technique. It uses ensemble learning to help businesses handle regression and classification issues. This approach is useful for developers since it eliminates the problem of dataset overfitting. It's a highly useful tool for creating accurate predictions in businesses' strategic decision-making.

V. ACKNOWLEDMENT

There is always inspiration gained by many in the completion of work. This project is made possible with the guidance of guide Professor Sonia Wadhwa (Assistant professor, Department of Computer Science & Engineering). We are thankful to our respected Head Of the Department Professor Sourabh Yadav (Department of computer science and engineering) for providing us with all the necessary facilities and also grateful to our Head Of Institute Dr. B.S. Chawla Sir (Principal, Government Engineering College, Bilaspur) for his motivation and encouragement for this research to be successful.

REFERECES

1. Leistner C, Saffari A, Santner J, Godec M, Bischof H, Semi-Supervised Random Forests, ICCV IEEE, Conference Proceedings, 506-513 (2009).
2. Bernard S, Heutte L, Adam S, Using Random Forest for Handwritten Digit Recognition, International Conference on Document Analysis and Recognition 1043-1047, (2007).
3. Bernard S, Heutte L, Adam S, On the Selection of Decision Trees in Random Forest, Proceedings of International Joint Cobference on Neural Networks, Atlanta, Georgia, USA, June 14-19,302-307, (2009).
4. Hansen L, Salamon P, Neural Network Ensembles, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol 12 No 10, (1990).
5. I. H. Witten, E. Frank, Weka: Practical machine learning tools and techniques, Morgan Kaufmann publisher, (2005).
6. Kosorok M, Ma S, Marginal Asymptotics for the Large p Small n paradigm: With Applications to Microarray Data, Ann Statist 35, 1456-1486, (2007).
7. Kouzani A, Nasireding G, Multilabel Classification by BCH Code and Random forest, International Journal of Recent Trends in Engineering, Vol 2, No 1, (2009).
8. Krogh A, Vedelsby J, Neural Network Ensembles, Cross Validation, and Active Learning, Advances in Neural Information Processing Systems Vol 7, MIT Press , 231-238, (1995).
9. Kuncheva L, Diversity in Multiple Classifier Systems, Information Fusion, Vol 6, Issue 1, 3-4, (2005).
10. Latinne P, Debeir O, Decastecker C, Limiting the number of trees in Random Forest, MCS, UK (2001).
11. Leistner C, Saffari A, Santner J, Godec M, Bischof H, Semi-Supervised Random Forests, ICCV IEEE, Conference Proceedings, 506-513 (2009).
12. [Maudes J, Rodridugz J, Garcia-Osorio C, Disturbing Neighbors diversity for decision forests, Studies in Computational Intelligence, Vol 245, 113-133, (2009).
13. Nikulin V, McLachlan G, Ng S, Ensemble Approach for Classification of Imbalanced Data, Proceedings of the 22nd Australian Joint Conference on Advances in Artificial Intelligence, Springer-Verlag (2009).
14. Breiman L, Bagging Predictors , Technical report No 421, (1994).

15. Brieman L, Random Forests, Machine Learning, 45, 5-32, (2001).
16. G. Biau, F. Cerou, and A. Guyader. On the rate of convergence of the bagged nea ´ rest neighbor estimate. Journal of Machine Learning Research, 11:687–712, 2010.
17. G. Biau and L. Devroye. On the layered nearest neighbour estimate, the bagged nearest neighbour estimate and the random forest method in regression and classification. Journal of Multivariate Analysis, 101:2499–2518, 2010.
18. V. Svetnik, A. Liaw, C. Tong, J. Culberson, R. Sheridan, and B. Feuston. Random forest: A classification and regression tool for compound classification and QSAR modeling. Journal of Chemical Information and Computer Sciences, 43:1947–1958, 2003.
19. J. Shawe-Taylor and N. Cristianini. Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge, 2004.
20. Scerbo M L, Radhakrishnan H, Cotton B, et al. Utilization of the Random Forest Algorithm to Predict Trauma Patient Disposition Based on Pre-hospital Variables [J]. Journal of Surgical Research, 2013, 179(2): 271.

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING