

# Weight Optimization through Differential Evolution Algorithm in Neural Network based Ensemble Approach

Antima Modi

Department of Computer Science and Engineering,  
UCE, Rajasthan Technical University,  
Kota, India

Amit Swami

Department of Computer Science and Engineering,  
UCE, Rajasthan Technical University,  
Kota, India

**Abstract**— In recent years, with the advent of computational intelligence, machine learning has remained an extremely popular area for research. Till now, in this field, many machine learning models are available for prediction. So, rather than implementing new machine learning method, we should go for ensembles of these models for getting better results. The ensemble learning to represent the most innovative strategy in the field of machine learning. It provides better predictions than ordinary machine learning approaches and provides better accuracy. Generally, in machine learning, only one learner is used for the training and prediction, whereas ensemble learning methods to build a number of learners and then aggregate their results to obtain a more precise prediction. In this work, neural network ensemble is used and the weights for predictions are optimized using Differential Evolution (DE) Algorithm to obtain a more precise final prediction. Comparison between NN-ensemble and other ensemble approaches is performed on various evaluation measures and NN-ensemble outperforms with accuracy 91.82%.

**Keywords**— Machine learning, Neural Network, Differential Evolution (DE) Algorithm, Ensemble.

## I. INTRODUCTION

Machine learning represents an application of Artificial Intelligence (AI) that provides frameworks the capacity to consequently take on and enhance as a matter of fact without being expressly customized. An ensemble is a group of items viewed as a whole rather than individually. Ensemble Learning utilizes a technique that applied to provide training to multiple learners and aggregate their outcomes, considering them as a "committee of decision-makers". The key idea behind it is the decisions of the committee, with single predictions aggregated suitably, ought to have more acceptable overall accuracy, on average, as in contrast to any committee member. The ensemble members might be predicting real-values, labels for class, posterior probabilities, clustering, rankings, or any quantity. So that their predictions can be integrated with distinct techniques, including voting, averaging, and probabilistic methods. Predictions from machine learning algorithms play a vital role in modern human's life. As a user of technology, we complete certain tasks that desired to make a decision or classify something like in the medical field, research and development, also in our day to day life and in any data analysis task. There are several machine learning algorithms like the random forest, K-nearest neighbor, decision tree, support vector machine, etc. Pantola et al. [1] presented an ensemble technique with the weighted

average strategy to combine machine learning models to obtain a more precise prediction. In weighted average strategy, weights are assigned manually to achieve precise results from ensemble model. Hence, it is a limitation of traditional ensemble approaches. Manual assignment of weights increases the probability of not getting the increased accuracy of ensemble model.

Instead of assigning weights manually there may be some method or algorithm to find these results. In the literature, there are many methods for finding weights for ensemble model like iDHS-EL [2], for finding weights for fusion used for grid search and iDNA-KACC-EL also used for the same purpose [3]. Liu, Zang et al proposed a method that measures the distance of incorrect classifying data values and for clustering, it used affinity propagation algorithm [4]. Many more methods with fuzzy logic are used in literature to aggregate classifiers like multiple kernel learning for better prediction [5, 6, 7]. Samples having high fuzziness get higher incorrect classification cases. To get rid-off divide and conquer method is proposed [8].

For getting better performance through machine learning models there ensembles approaches should be refined. In Bioinformatics for protein structure prediction, Rana et al. showed that random forest gives a more precise prediction for classification and regression data [9, 10]. Then neural network ensemble [11] approach is proposed by Rayal and Rana on regression dataset and more promising results are gained. They compare classical ensemble approaches with neural network based ensemble approach and found that N-semble [12] outperform. But there was not any scheme to optimize weights to get better predictions. So, in our proposed work we are also working on neural network based ensemble approach for protein structure prediction using regression dataset. Over here, weight optimization is done through Nature Inspired Algorithm (NIA) which is Differential Evolution (DE) Algorithm.

The paper would be structured as: in Material and Models, the description of Dataset and Models is provided. In Methodology and Implementation, Differential Evolution Algorithm and proposed work are discussed. Then Experimental Results and K-fold Cross-Validation for proposed work is explained and complete work is enclosed in conclusion.

## II. MATERIAL AND MODELS

### A. Data with its Description

Proteins are the core component of our life. Proteins provide the foundation of structures such as skin, hair, and tendon and they are the primary cause for catalyzing and synchronizing bio-chemical reactions, transporting molecules [13]. Regression data [14] is used with the proposed approach, for protein structure prediction. The physicochemical properties of a protein structure are used to determine the quality of the protein structure. Therefore, these properties are utilized to identify native or native-like structure from various predicted structures. In this work, the machine learning regression models are experimented with six physicochemical properties to predict the root mean square deviation (RMSD) of the protein structure. This prediction works in the absence of its true native state and each protein structure ranges from 0Å to 5 Å RMSD space. Physicochemical properties explored in the paper are as described with each attribute in TABLE I. RMSD value is predicted and equated with known protein structure [9]. In the TABLE II, a sample of data of five instances is shown from the specified dataset. For Regression purpose, the value of RMSD is considered as continuous.

TABLE I. DESCRIPTION OF ATTRIBUTES

S. No.	Table Column Head	
	Attribute	Description
1	RMSD	Root Mean Square Deviation
2	Area	Total Surface Area
3	ED	Euclidean Distance
4	Energy	Total Emperical Energy
5	SS	Secondary Structure penalty
6	SL	Sequence Length
7	PN	Pair Number

TABLE II. A SMALL DATA SAMPLE

Table Column Head						
RMSD	Area	ED	Energy	SS	SL	PN
3	12462.1	60594.3	-3668.59	32	247	2754
0	15337.3	121624	-5856	35	299	4918
5	9418.94	17629.7	-593.842	195	133	890
4	9331.73	11717.2	-1059.27	135	156	654
1	3769.08	644.369	-615.49	24	50	42

### B. Machine Learning Models

This section explores the entire machine learning model description with configuration and specifications and these are depicted in TABLE III.

## III. METHODOLOGY AND IMPLEMENTATION

### A. Differential Evolution(DE) Algorithm

Differential Evolution (DE) algorithm is an optimization technique with stochastic nature. It is an Evolutionary Algorithm (EA) with faster and simpler evaluation process. It

was invented by Storn and Price [17]. The steps involve in DE is shown as a flowchart in Fig 1.

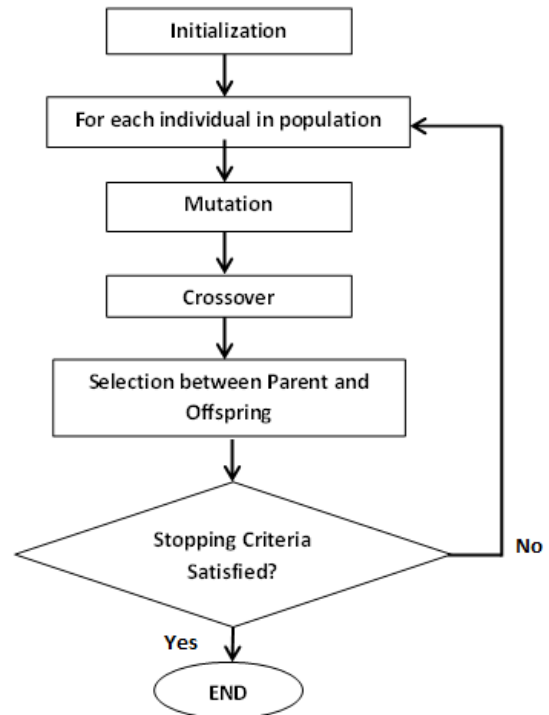


Fig 1: DE : Flow Chart

The procedural illustration is provided in Algorithm 1. In DE, a solution is finding by potential solutions (individuals). The search space is a D-dimensional space defined as  $i = 1, 2, \dots, SN$ . Where, SN is the population (number of individuals). The algorithm employs the operators, namely Mutation, Crossover, and Selection. Initially, to find a trial vector mutation operator is utilized for each individual of the present population. Then, using crossover operator, parent vector and trail vector are generated and after this generated offspring is compared with the parent. In selection operator, two functions are there: first, mutation operation with a selected individual is used to generate a trial vector. Second, for the next generation select the best fitness values between parent and Offspring. At the end return individual with the best fitness as solution [18]. In our work, we are using DE for finding optimized weights so that we can achieve improved results from the complete ensemble model. For this reason, we have considered our objective function as follows:

$$Y_i = \sum W_i * X_i \dots\dots\dots (1)$$

Where,  $Y_i$  represents the Actual RMSD and  $X_i$  represents the Predicted value of the top three models.  $W_i$  is the optimized weights for each model's prediction in top three models.

TABLE III. DESCRIPTION OF CLASSIFICATION MODEL USED

Table Column Head				
S. no.	Models	Description	Required Packages	Parameters
1	Random Forest	Random decision forest of trees	randomForest	sampling=adaBoost, mtry
2	Neural Network	Neural Network with back propagation is used	neuralnet	algorithm=rprop+, size=15
3	CART	Classification And Regression tree for Classification	Rpart	maxdepth=15
4	Ksvm	SVM with kernel fuction	Kernlab	kernel function
5	SVMPoly	SVM with polynomial method	Kernlab	svmPoly
6	Decision Tree	Tree based model for classification	Rpart	method=rpart2
7	Xgboost	Extreme Gradient Boosting for faster and scalable output	Xgboost	nrounds=2000
8	SVM	Discriminative classifier formally defined by a separating hyperplane	e1071	method, kernel
9	LM	Multinomial classifier	Caret	Multinom

**Algorithm 1 : Differential Evolutionary Algorithm [18]**

**Initialization**

Initialize control parameters CR (Crossover Rate) and F (Scale Factor)

Initialize and create population vector P(0) of SN individuals

**Evaluation with Operators**

while (stopping condition(s) ≠ true) do

for each individual,  $x_i(G) \in P(G)$  do

- Evaluate Fitness,  $f(x_i(G))$ ;
- Mutation : Generate trail vector  $u_i(G)$
- Crossover : Generate an Offspring  $x'_i(G)$
- if  $f(x'_i(G))$  is better than  $f(x_i(G))$  then  
 Add  $x'_i(G)$  to P(G+1);  
 else  
 Add  $x_i(G)$  to P(G+1)  
 end if

end for

end while

Return individual with best fitness as solution;

**B. Methodology**

The proposed approach is shown as a step-down flow diagram in Fig 2. It is organized in four parts: (1) data splitting, (2) train and test base models, selection, (3) weight Learning through DE algorithm,(4) train and test neural network. Dataset is loaded and 75% of the dataset is the training data (Set-1) and rest of the dataset (Set-2) is used to

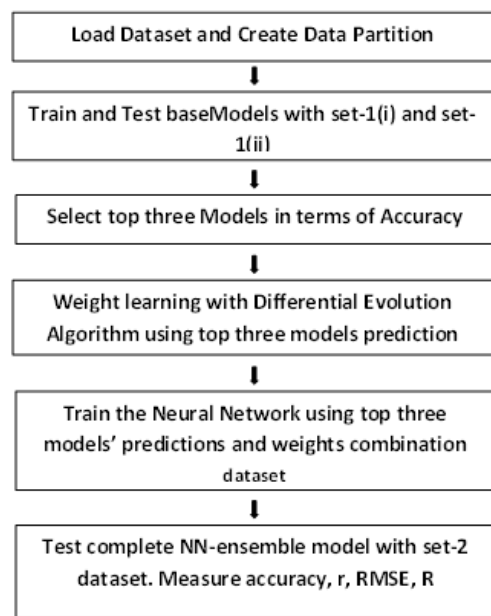


Fig 2: Steps for Proposed Approach

test the NN-ensemble model. So, the training data is expressed as Set-1 and testing data is expressed as Set-2.

In the Phase-II, after getting data from phase-I, dataset (Set-1) is further partitioned into a ratio of 2:1. This specific data splitting is used to fit more efficient ensemble model without compromising the testing data. Further, Base models are trained on 50% of the whole dataset and tested on 25% of the dataset. After obtaining prediction results Top three base models are selected based on accuracy parameter.

In the Phase-III, top three models prediction and corresponding target RMSD are used to form a dataset and weight learning is done through DE Algorithm 1. It will tune the parameters and determine the Optimized weights to form new training and testing dataset for complete ensemble model.

In the Phase-IV, Train the neural network by establishing the relationship between top three model's actual-predicted values with optimized weights. This special combined data is

used to improve the performance of final ensemble model. The Set-2 dataset is used in the same way as training to perform testing of the final ensemble model.

TABLE IV. RESULTS EVALUATION FROM PHASE-II

Table Column Head					
S. No.	Model	R	R <sup>2</sup>	RMSE	Acc%
1	Random Forest	0.84	0.71	0.59	87.04
2	Neural Network	0.29	0.08	1.42	52.75
3	CART	0.76	0.58	0.69	84.4
4	Ksvm	0.71	0.5	0.8	81.35
5	SVMPoly	-0.01	0	1.55	52.85
6	Decision Tree	0.01	0	1.48	37.91
7	Xgboost	0.78	0.61	0.62	85.85
8	SVM	0.47	0.22	1.24	63.99
9	LM	0.8	0.64	1.88	47.21

C. Model Evaluation Parameters

The prediction results are compared and priority is given to the models by considering their respective accuracy.

- Accepted Error: The acceptable error range e ranges from [0,1]. It defines acceptable error range between predicted and actual value.
- Accuracy: Accuracy is defined in eq (2) for the regression data model:

$$Acc\% = \frac{100}{n} \sum_{i=1}^n a_i \quad \dots (2)$$

$$a_i = \begin{cases} 1 & \text{abs}(y - \hat{y}) \leq \text{Acceptable Error} \\ 0 & \text{Otherwise} \end{cases} \quad \dots (3)$$

Where, y is predicted target and  $\hat{y}$  is actual target, total number of instances is denoted by n.

- Correlation(r): Correlation(r) for actual and predicted value is measured as Pearson correlation which is defined as:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad \dots (4)$$

Where x is actual and y is predicted. The total number of instances is denoted by n. Value of r ranges from -1 to 1.

- R<sup>2</sup>: R<sup>2</sup> represents the variance proportion of the dependent variable explained by the regression model.

$$R^2 = r^2 \quad \dots (5)$$

- RMSE: Root Mean Square Error (RMSE) is used to measure error rate of regression data model. It is measured as:

$$\sqrt{\sum_{i=1}^n (p_i - a_i)^2} \quad \dots (6)$$

Where, a is used for actual and p is used for predicted value of target and n represents the total number of instances.

D. Implementation

The proposed procedure is followed as shown in Algorithm 2. The procedure is followed by four phases to process data and evaluates results from complete ensemble model.

Algorithm 2 : Complete NN-ensemble Algorithm

```

while d ≠ 0
    Phase I : DATA SPLITTING
        d1 = random (d, frac = 0.75)
        d2 = d - d1
        d1 = Training [Set - 1]
        d1 = Testing [Set - 2]
    Phase II : TRAINING TESTING BASE-MODELS
        dBaseTrain = random(d1, frac = 0.66)
        dBaseTest = d1 - dBaseTrain
        Train Base-Models on dBaseTrain
        Test Base-Models on dBaseTest
        absolute (yi - y'i)
        Acc%
        r
        R2
        RMSE
        BaseResult : NN(x)
    Phase III : WEIGHT LEARNING THROUGH DE
        Input NN(x) → DE Algorithm
        for i = 1 to 3
            measure Wi
        Combine weights to top three models predictions
    Phase IV : NN-ENSEMBLE AND TESTING PHASE
        Train NN - ensemble with training dataset after DE
        Test NN - ensemble on Set-2 Ddataset
        Acc%
        r
        R2
        RMSE
end
    
```

IV. EXPERIMENTAL RESULTS

A. Results and Discussions

This section provides, experimental results obtained from base models and ensemble models are presented in tabular format. TABLE IV shows the base models prediction results in terms of evaluation parameters, i.e., accuracy, r, R<sup>2</sup>, and RMSE on the test dataset (set-1 (II)). Top three models are selected on the basis of accuracy for further processing with an ensemble model as expressed in the algorithm. These top models are highlighted that are more accurate than other models, having least RMSE, highest accuracy, correlation and R<sup>2</sup>. the According to Pearson correlation formula if the attributes are significantly correlated with each other then, correlation will be high. Over here, correlation is compared between actual and prediction values of these models and according to that accuracy is measured.

TABLE V. PREDICTIONS FROM TOP MODELS

Table Column Head			
Actual RMSD	CART	Random Forest	Xgboost
5	5	5	5
0	0	0	0
5	5	4	5
4	4	4	2
5	4	3	3
4	4	4	4
0	0	0	0
3	3	3	3
1	1	1	1

According to results, top three selected models are random forest, Cubist, and xgboost are used for finding optimized weights according to the objective function definition with DE algorithm. Top three models prediction sample is depicted in TABLE V. DE Algorithm uses actual and predicted values of the top three models as input and returns output. It provides the most appropriate combination of weights or optimized weight for better results. After that, we will apply these predictions and weight combination to ensemble model as training data and testing dataset set-2. It is also using these weight combinations and on this dataset, complete neural network ensemble model is tested. Complete ensemble is evaluated on actual and predicted values. Results achieved from neural network ensemble are also compared with classical ensemble models.

TABLE VI. PERFORMANCE COMPARISON

Table Column Head					
S. No.	Model	r	R <sup>2</sup>	RMSE	Acc%
1	Random Forest	0.8	0.64	0.65	87.99
2	CART	0.79	0.62	0.67	86.35
3	Xgboost	0.77	0.58	0.74	86.32
4	NN-ensemble	<b>0.83</b>	<b>0.69</b>	<b>0.63</b>	<b>90.48</b>
5	Classical ensemble approach	0.78	0.45	0.82	83.52

TABLE VI show the prediction results in terms of accuracy obtained from selected models, its classical ensemble technique and neural network ensemble technique. Experimental results show that neural network ensemble outperforms among all these ensembles. It is because NN-ensemble training establishes the more accurate relationship between actual and predicted classes. Because of weight optimization, neural network performance increases and provides the more efficient performance as compared to other ensembles.

**B. Validation**

Validation i.e., proving the validity or accuracy of anything is a technique that is properly used in machine learning to measure the robustness of the final model. For this reason, we have performed K-fold cross validation [8] with the final ensemble model. Over here, we have considered K size as 10-fold, i.e. the models are trained and tested each time with random data samples of the same size. This test found that in each fold ensemble perform in a uniform way. So, these results testify to that NN-ensemble outperforms for regression dataset.

The validation results for all ensemble models are presented in TABLE VII with ten iterations of folds. Fig 3, shows that ensemble of neural network performs more efficiently than classical ensemble approach. It demonstrates that due to optimization of weights through DE algorithm neural network based ensemble approach performs in a more accurate way.

TABLE VII. 10-FOLD CROSS-VALIDATION FOR ACCURACY

ColumnTable Head					
Fold	NN-ensemble	Cubist	Random Forest	Xgboost	Classical Ensemble
1	<b>91.82</b>	84.58	87.99	86.32	84.67
2	<b>90.85</b>	83.46	87.62	85.51	84.16
3	<b>89.86</b>	85.43	88.02	85.87	84.66
4	<b>91.58</b>	82.09	86.86	87.25	83.95
5	<b>89.53</b>	83.88	87.22	88.18	83.45
6	<b>91.33</b>	84.48	87.83	85.61	84.33
7	<b>91.79</b>	83.54	88.46	87.63	85.36
8	<b>90.08</b>	84.7	87.52	88.08	85.09
9	<b>92.01</b>	81.98	87.94	85.99	85.61
10	<b>90.01</b>	84.41	87.09	88.66	84.05

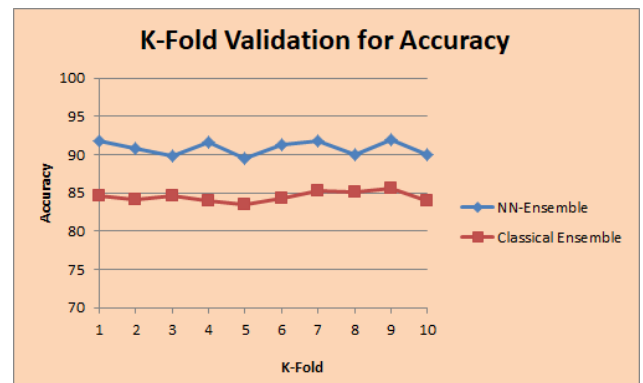


Fig 3: K-Fold Cross Validation NN-Ensemble and Classical Ensemble for Accuracy

**C. Conclusion**

In this work, Differential Evolution (DE) Algorithm is applied for optimizing prediction from top three base models to improve the results of neural network based ensemble approach. Initially, ten models are trained and tested and top three are selected based on accuracy. Next, these predictions are optimized through DE to improve the performance of the final ensemble model. Data partitioning is done in a specific way to achieve the extremely efficient model. The neural network is used to ensemble the models because of its generic nature. The experimented results have been evaluated through data tables, verified through K-fold cross-validation method. The obtained results clearly show we can improve ensemble model results by optimizing parameters. Model evaluation is done through evaluation parameters like r, R<sup>2</sup>, RMSE and accuracy and compared other ensemble approaches.

It is expected by using more physical and chemical properties may result in improved accuracy and can decrease computing time. Other Nature-inspired algorithms can also be used to enhance the performance of ensemble models. This idea would represent an efficient approach for protein structure identification. Additionally, this work can be

prolonged in the prediction of template-modeling (TM-score) and global distance test (GDT TS-score). So, at the last, we can say that instead of making new machine learning models we can improve ensemble results through an effective optimization scheme.

## REFERENCES

- [1] Paritosh Pantola, Anju Bala, and Prashant Singh Rana, "Consensus based ensemble model for spam detection," In Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on, pages 1724–1727. IEEE, 2015.
- [2] Bin Liu, Ren Long, and Kuo-Chen Chou, "idhs-el: identifying dnase i hypersensitive sites by fusing three different modes of pseudo nucleotide composition into an ensemble learning framework," *Bioinformatics*, 32(16):2411–2418, 2016.
- [3] Bin Liu, Shanyi Wang, Qiwen Dong, Shumin Li, and Xuan Liu. "Identification of dna-binding proteins by combining auto-cross covariance transformation and ensemble learning," *IEEE transactions on nanobioscience*, 15(4):328–334, 2016.
- [4] Bin Liu, Shanyi Wang, Ren Long, and Kuo-Chen Chou, "irspotel: identify recombination spots with an ensemble learning approach," *Bioinformatics*, 33(1):35–41, 2016.
- [5] Bin Liu, Deyuan Zhang, Ruifeng Xu, Jinghao Xu, Xiaolong Wang, Qingcai Chen, Qiwen Dong, and Kuo-Chen Chou, "Combining evolutionary information extracted from frequency profiles with sequencebased kernels for protein remote homology detection," *Bioinformatics*, 30(4):472–479, 2013.
- [6] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh, "Two-stage learning kernel algorithms," In *ICML*, pages 239–246. Citeseer 2010.
- [7] Manik Varma and Bodla Rakesh Babu, "More generality in efficient multiple kernel learning," In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1065–1072. ACM, 2009.
- [8] Xi-Zhao Wang, Rana Aamir Raza Ashfaq, and Ai-Min Fu, "Fuzziness based sample categorization for classifier performance improvement," *Journal of Intelligent & Fuzzy Systems*, 29(3):1185–1196, 2015.
- [9] Sonal Mishra, Yadunath Pathak, and Anamika Ahirwar, "Classification of protein structure ( $rmsd \leq 6 \text{ \AA}$ ) using physicochemical properties," *International Journal of Bio-Science and Bio-Technology*, 7(6):141–150, 2015.
- [10] Prashant Singh Rana, Harish Sharma, Mahua Bhattacharya, and Anupam Shukla. "Quality assessment of modeled protein structure using physicochemical properties," *Journal of bioinformatics and computational biology*, 13(02):1550005, 2015.
- [11] Lars Kai Hansen and Peter Salamon, "Neural network ensemble," *IEEE transactions on pattern analysis and machine intelligence*, 12(10):993–1001, 1990.
- [12] Rishith Rayal, Divya Khanna, Jasminder Kaur Sandhu, Nishtha Hooda, and Prashant Singh Rana, "N-semble: neural network based ensemble approach," *International Journal of Machine Learning and Cybernetics*, pages 1–9, 2017.
- [13] Er Amanpreet Kaur and Baljit Singh Khehra, "Approaches to prediction of protein structure: A review," *International Research Journal of Engineering and Technology*, 2017.
- [14] Sourceforge : <http://bit.ly/RF-PCP-DataSets>
- [15] CART website: CRAN.R-Project. <http://goo.gl/ulWSI3>
- [16] XgBoost website: CRAN.R-Project. <http://goo.gl/ulWSI3>
- [17] Kenneth V Price, "Differential evolution: a fast and simple numerical optimizer," In *Fuzzy Information Processing Society*, 1996. NAFIPS, 1996 Biennial Conference of the North American, pages 524–527. IEEE, 1996.
- [18] Harish sharma Kavita Sharma, "Nature Inspired Algorithms: An Introduction," *Soft Computing Research Society*, 2009.