ISSN No. 0976-5697

Volume 8, No. 5, May-June 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Enhancing Random Forest Classifier using Genetic Algorithm

Sania Jawaid
Department of Computer Science & Engineering
Jamia Hamdard (Hamdard University)
New Delhi, India

Mohd Abdul Ahad
Department of Computer Science & Engineering
Jamia Hamdard (Hamdard University)
New Delhi, India

Abstract: Classification is a problem of distinguishing and categorizing an observation into sub-populations or groups on the basis of certain prior observations whose category membership is known which are called the training data sets. Classification can also be termed as a part of pattern recognition. Its usage is not only limited to the computer analysis from discerning spam emails from genuine ones but also in daily use by doctors for classifying diseases of the patients by observing their characteristics e.g. blood pressure, heart rate, symptoms etc. Therefore, it becomes highly important to have accurate classification methods so that problems and its causes can be identified quickly and accurately in order to solve them. One such algorithm for classification in machine learning and statistics is the classifier called 'Random Forest'. While decision trees lack behind with their low bias and high variance trade-off, Random Forest is one of the algorithms in supervised learning where low bias and comparatively lower variance than decision trees win for large training data sets as they have low asymptotic error. However, by reducing the correlation between trees we can further reduce the variance and hence improve the algorithm. Therefore, by modifying the existing algorithm by overcoming a few of its demerits can make a classifier more accurate and trustworthy. This paper tries to propose a solution by combining one of the optimizing strategies i.e. Genetic Algorithm with Random Forest to overcome its problem of over-fitting the datasets.

Keywords: Machine learning; supervised learning; Classifier; Random Forests; over-fitting; Genetic Algorithm

I. INTRODUCTION

A. *Machine Learning*

It is a field of computer science which is originated from pattern recognition and artificial intelligence. Without being specially programmed, it makes the computers learn by themselves. It is based on construction of efficient algorithms for learning from sample inputs also known as training data and predicting on test data [1]. Some applications of Machine Learning include Optical character recognition (OCR), Spam Filtering, Search Engines etc. [2]. There are various types of machine learning which are listed as follows**:**

1) Supervised Learning: In this type of learning, a target or outcome variable also known as a dependent variable is predicted using a set of independent variables known as predictors [3].

2) Unsupervised learning: In this type of learning, there are no outcome or target variables to predict or estimate the given input [4]. It is used to group population into sub populations by assigning them into clusters by analyzing the similarities and dissimilarities of individual objects.

3) Semi- Supervised Learning: In this type of learning, the input data is mixed with unlabeled and labeled examples [3].

4) Reinforcement Learning: According to this technique, the machine gets trained continuously using trial and error in a specific environment [4]. It then uses past experience and learning in the form of feedback given by the environment, to analyze and make best possible business decisions [5].

5) Transduction: It is somewhat similar to supervised form of learning with a slight difference. It does not create a function specially and tries to estimate new outputs based on new and training inputs/ outputs [6].

6) Learning to Learn: In this technique, the algorithm learns by self on the basis of some induction and bias based on past experiences [6].

B. *Classifier*

In machine learning, the term classification is considered as a part of supervised learning. Classification is a process where categorization is done for instances / observations which are recognized and understood on the basis of some training data sets [2]. Classifier is the logic or algorithm needed to implement classification over observations often known as instances. A classifier can also refer to a mathematical function used in a classification algorithm [1]. Classification is most commonly used in a variety of fields and has many applications. Some include: Spam Mail Detection, Image Classification, Targeting Ads, Sentimental Analysis, Medical Diagnosis, Risk Assessment etc. [2].

C. *Types of Machine Learning Classifier Algorithms*

1) Decision Trees: It is a supervised learning type of algorithm which is mostly used for classification sort of problems [6]. It is used for both categorical and continuous values. In this algorithm, the population is divided into sub populations on the basis of a significant variable / attributes to make clear distinctions. The splitting decision can be based on Gini index, Information Gain, Chi-square entropy etc. of the variables involved [5].

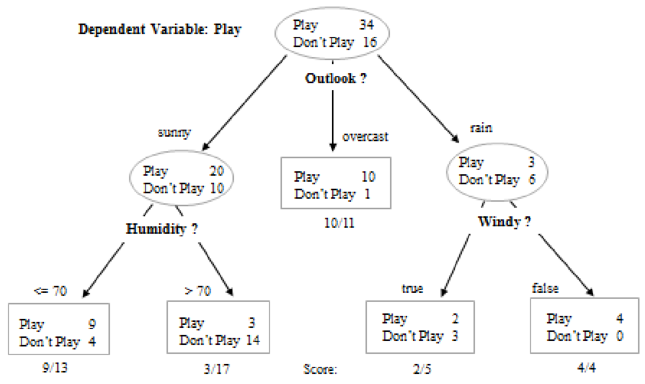

Figure 1. A Simple Decision Tree [7]

One of the disadvantages of Decision Tree is that they can easily overfit the data and are less scalable. That's why Random Forests are more popular than decision trees as they are faster and more scalable [8].

2) Random Forests: It is one of the machine learning algorithms that is capable of both classification and regression tasks. It does a fairly good job by assuming dimensional reduction mechanisms, taking into consideration the outlier and missing values and other steps of exploring data. It is one of the forms of the ensemble learning methodology where a weak group of models i.e. decision trees combine to form a more powerful model which is called a random forest [9]**.** A random forest constitutes of building a number of decision tree, where each decision tree then votes for a class like in a democratic process [4]. The final classification is done by averaging the total number of votes by each class. Random Forests present two types of models: Classification and Regression. Classification Model is based on categorical input or dependent variables whereas, the Regression model is based on the continuous or numeric input variables [8].

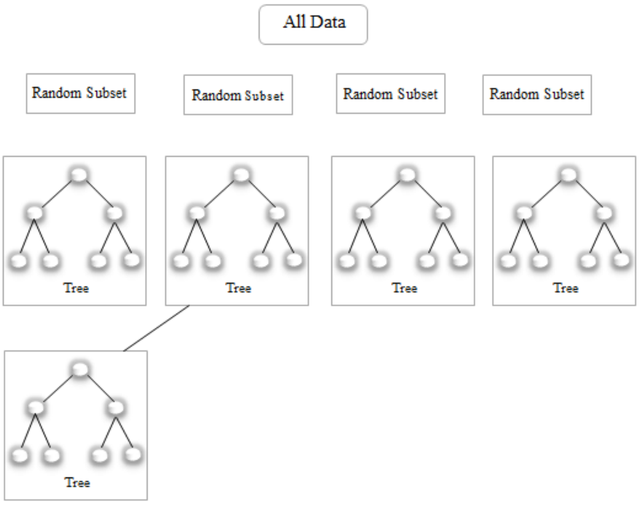

Figure 2. A Sample Random Forest [8]

D. Random Forests Classifier In Detail

Random Forest works in the following manner:

- Random Record Selection: Each Tree is trained on two-thirds of training data roughly i.e. around 66 %. Records are drawn at random with replacement [10].
- Random Variable Selection: Some variables (say m) are selected at random, known as predictors. The best predictor variable is used to split the nodes. The value of m is held constant while growing the forest [9].
- Using the left over 34% of data, misclassification rate is calculated for each tree and aggregated. This is also known as the out of the bag error rate [10].
- Each tree votes for a certain class giving a classification of their own. The forest decides the classification on the basis of aggregation of votes from all the trees. The vote will be Yes or No, for a binary dependent variable [10]. Counting the number of Yes votes will give the RF score of that classification. In Regression trees, the average of the number of votes is taken for calculating the predicted probability.
- Random Forests uses multiple techniques like Bagging, Bootstrap sampling etc. [11].

E. Background

Random Forests aren't good at generalising with completely new data. E.g. if 1 candy costs $1, 2 candies costs $2, 3 candies costs $3 etc., 10 candies should cost $10. But random forest cannot solve this problem however, linear regression can. Also if a variable is a categorical variable with multiple levels, Random Forest becomes biased [12].

The forest error rate depends on two things:

- Correlation between any two trees in the forest. This means increasing the correlation between trees increases the forest error rate [13].
- The strength of each individual tree in the forest. This means, a tree with low error rate is a strong classifier. By increasing the strength of the individual tree, the forest error rate gets decreased [10].

1) How to Fine Tune the Random Forest: Random Forest is very sensitive to the training data and redundancy in the trees or a bad choice of number of random variables used can adversely affect the strength of the whole forest and its classification [12]. Thus it becomes extremely important to overcome this demerit of random forest by picking the optimal set of trees for classification and prediction [11].

This paper tries to attempt to solve this particular problem by combining one of the optimising strategies of Genetic Algorithm [14] with Random Forests.

2) Genetic Algorithm: The Genetic Algorithm is a different type of algorithm all together which simulates the biological phenomena of the survival of the fittest which is used in the field of Artificial Intelligence [15]. GA involves some of the processes like crossover, mutation and selection [16].

II. LITERATURE REVIEW

Some of the research work presented by other authors in the area of machine learning and GAs are as follows:

Table I. Literature Review of Machine Learning and GAs Techniques

S No.	Title and Publication	Author(s)	Technique
1.	"Fault diagnosis in spur gears based on genetic algorithm and random forest", Mechanical Systems and Signal Processing, Science Direct, 70-71 (2016) 87–10	MarielaCerrad, GroverZurita, DiegoCabrera, René-VinicioSánchez, Mariano Artés, ChuanLi	In this paper [17], Genetic Algorithm based feature selection is combined with Random Forest models for diagnosis in gear fault. In this paper, a two-step approach is proposed for designing a fault diagnosis system for spur gears.

2.	"A Guided Hybrid Genetic Algorithm for Feature Selection with Expensive Cost Functions", Procedia Computer Science, SciVerseScienceDirect, Volume 18, 2013, Pages 2337-2346	Martin Junga, JakobZscheischlera	In this paper [18], the author tries to propose a guided hybrid genetic algorithm, to minimize cost function evaluation. To make the stochastic backward search of the genetic algorithm more efficient, a guided variable elimination is used.
3.	"GARF: Towards Self-optimised Random Forests"	Mohamed Bader-El-Den and Mohamed Gabe	The paper [19] proposes an ensemble based machine learning approach that makes use of a number of classifiers. The paper suggests an idea of having a self-optimized Random Forest algorithm which is able to dynamically change trees in the forest using Genetic Algorithm. The modified algorithm is known as GARF, i.e. Genetic Algorithm based Random Forest.
4.	"Random forests classifier for machine fault diagnosis", Journal of Mechanical Science and Technology, Springer, 22 (2008) 1716~1725	Bo-Suk Yang, Xiao Di and Tian Han	The paper [20] suggests a hybrid method combining both genetic algorithm and random forest to improve the classification accuracy in machine fault diagnosis. GA is used to evaluate the best suited parameters for RF.
5.	"Modified Decision Tree Algorithm Based on Genetic Algorithm for Mobile User Classification Problem", The Scientific World Journal Volume 2014, Article ID 468324, 11 pages	Dong-sheng Liu and Shu-jiang Fan	The paper [21] aims to modify a decision tree algorithm using genetic algorithm in order to overcome the limitations of previous classification methods and use it for mobile user classification problem in order to offer better services to mobile customers.

III. PROPOSED SOLUTION

Following is the graph showing the steps of the proposed algorithm. For the ease of understanding and use, Random Forest will be referred to as RF and Genetic Algorithm will be referred to as GA.

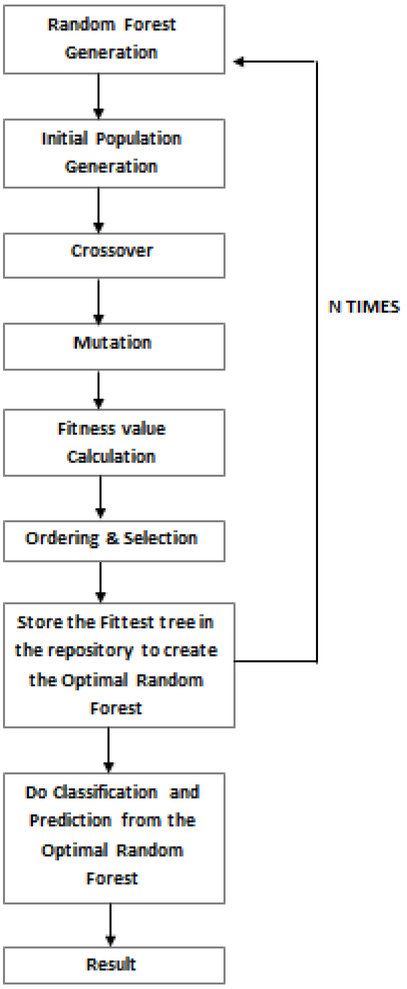
Random Forest Generation
Initial Population Generation
Crossover
Mutation
Fitness value Calculation
Ordering & Selection
Store the Fittest tree in the repository to create the Optimal Random Forest
N TIMES
Do Classification and Prediction from the Optimal Random Forest
Result

Figure 3. Proposed Algorithm

A. Random Forest Generation

A population of Random Forest is generated with multiple decision trees in it. It was created using an ensemble method including many techniques such as Bagging, Bootstrap Sampling, and Removing out of the bag error rate, etc. [10].

B. Initial Population Generation

The random forest generated in the previous step, is converted into an initial population for the Genetic Algorithm. This process of conversion involves the following steps for each tree in the random forest**:**

1) Starting from the root of the tree, assign codes to the branches and the leaf nodes. While the branch to the left of a node including the root node is assigned as a zero and the branch to the right of the node is assigned as 1, just like encoding in Huffman Tree [22].
2) For the leaf nodes, the classes are the labels. Assign all the classes their unique decimal number and convert them into their binary formats.
3) Chromosome Building for GA population: To create the chromosome, the tree is stored in an array in the form of Depth First Search (DFS). Each branch's code is separated from their corresponding branch's or leaf node's binary code by inserting a padding, say, '-1'.

4) The process is repeated to convert all the trees in the Random Forest as chromosomes in the initial population required for GA.
5) Following is the diagram showing the process of Encoding and Conversion of a sample decision tree:

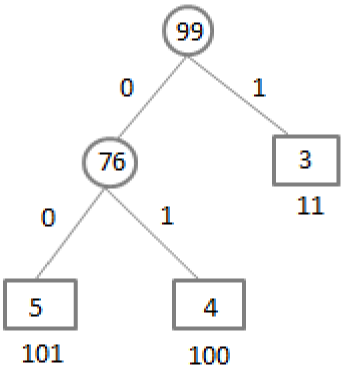

Figure 4. Decision Tree

0	-1	0	-1	1	0	1	-1	1	-1	1	0	0	-1	1	-1	1	1

Figure 5. Chromosome Building for GA population

After the initial population of trees in the form of chromosomes is generated, the population undergoes various GA processes to achieve randomization and diversification in the population. Each chromosome's length is the number of genes it contains [15].

C. Crossover

Crossover is applied to recombine multiple chromosomes and create an offspring from them [15]. It is applied on two chromosomes which are randomly chosen based on a certain crossover probability, also called as the crossover rate [21]. The number of crossovers is calculated from the below formula [16]:

num_of_crossover = crossover_rate * a * b/100,
Where a = Chromosome Length, b= Number of Chromosomes.

After a random crossover point is generated, the genes of the first parent are copied to the offspring up till the crossover point and the rest of the genes are taken from the second parent. The resultant offspring is different than its parent [15].

D. Mutation

Mutation is applied to achieve diversity in the population by mutating the genes in the population of chromosomes [23]. A low mutation rate is chosen which is used to find out the number of mutations that needs to be carried out in the population [18]. In our proposed approach, we have made use of the bit inversion mutation operator using the following formula [16]:

num_of_mutation = mutation_rate * a * b/100,
Where a = Chromosome Length, b= Number of Chromosomes.

Mutation prevents the population from becoming too similar to each other which helps evolution by trying to avoid the local minima [24].

E. Fitness Value Calculation

Fitness Function is a mathematical function used to define the fitness of the chromosomes in the GA population [17]. In the proposed algorithm, all the binary values are first converted into their corresponding decimal value and their fitness value is calculated. The formula used for fitness value calculation is as follows [16]:

$F = 1 / (1 + e^{(-\alpha)})$,
Where $\alpha = X_1\alpha_1 + X_2\alpha_{2,}$
α_1 = Frequency Test, α_2 = Gap Test,
$X_1 = X_2 = 1$

1) Frequency Test: This test is performed to check frequency of each chromosome and the number times it has been repeated [25].

2) Gap Test: This test is done to check the interval and the gap between two recurring numbers and its implications [26].

F. Ordering & Selection

The resulting dataset after applying above GA operators are arranged in the descending order of their fitness value. The topmost chromosome which has the highest fitness value is selected [14] [5].

G. Storing of Best Fit Tree in the Repository

The fittest chromosome is then stored in the repository and the whole process is repeated 'n' times [14].

H. Applying Classification and Prediction from the Optimal Random Forest Created

Each chromosome in the repository is then converted to its tree form to construct a well optimized Random Forest. This Random Forest is then analysed and the trees in the forest are allowed to vote [27]. The mode of all the votes is the classification given by the newly generated Random Forest. GA operators were used to achieve randomization in the forest since decrease in correlation means decrease in forest error rate. Since the trees used here are optimal, therefore the strength of the whole forest also increases and hence the produced classifier will be more accurate [13].

IV. RESULTS AND DISCUSSIONS

This paper aims to find the optimally fit forest used in random forest algorithm to generate better classifications and predictions. The trees used in random forest are least correlated and are random in nature. Since decreasing the correlation between trees decreases the forest error rate, hence the strength of the whole forest is increased. This randomness and fitness of trees is achieved using Genetic Algorithm. For GA, the chromosome is constructed having a fixed gene length of 30, with extra padding of 0 bits added at the end if required. The correlation of trees is tested against the crossover rate of 2.5, the mutation rate of 0.5 and the fitness formula using Gap Test and Frequency Test. The process is then repeated 'n' number of times where 'n' is taken as 10 until the whole forest is grown. The classification and prediction is done using this fit forest. For validating the test results, out of the bag error is calculated. Following are some of the observations shown in a graphical manner:

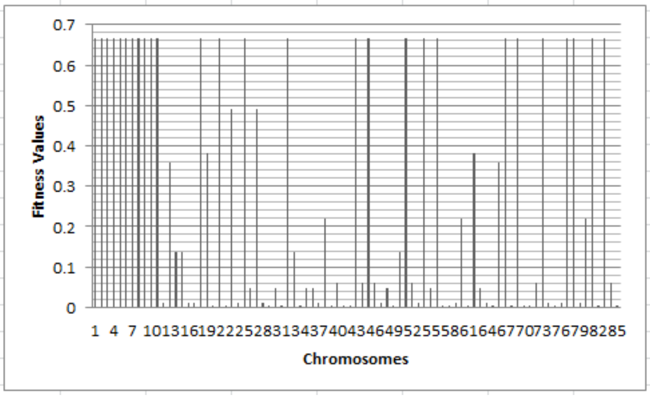

Figure 6. Fitness Graph of decision trees in random forest after GA is applied

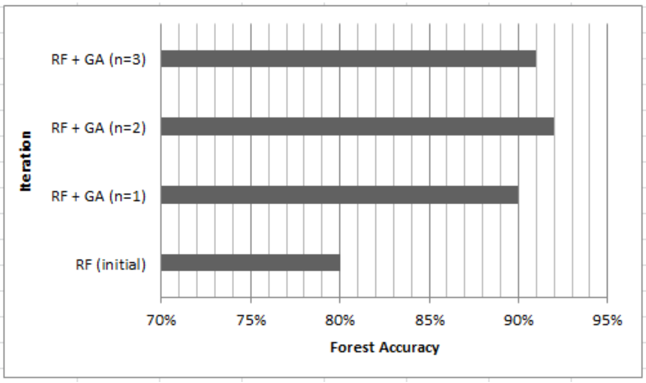

Figure 7. Comparative analysis of the accuracy of the Random Forest Algorithm before and after GA was applied

V. SUMMARY AND CONCLUSIONS

This paper puts forward a new approach to enhance the classifier algorithm called Random Forest. The modified algorithm will increase the efficiency in terms of the classification produced and give more accurate result. It uses the concept of Genetic algorithms to increase the randomness of the forest and pick the least correlated tree.

Modifying the classification algorithms along with the optimization algorithm is done to perform classification in a more accurate manner. The modification in random forest algorithm proposed in this paper can not only overcome the existing classification issues but also has an edge over the randomization of the forest involved using complex methodologies.

Moreover, our proposed approach can be expanded further for solving more classification problems by trying to create new ensemble models using different classification algorithms.

VI. REFERENCES

[1] "A Course on Machine Learning", by Stanford University, provided by Coursera. [Online]. Available: https://www.coursera.org/learn/machine-learning

[2] "A Course on Machine Learning: Classification", by University of Washington, provided by Coursera. [Online]. Available: https://www.coursera.org/learn/ml-classification

[3] E. Alpaydın, "Introduction to Machine Learning (Adaptive Computation and Machine Learning)," MIT Press, 2004.

[4] "Essentials of Machine Learning Algorithms", by Analytics Vidhya,[Online]. Available: https://www.analyticsvidhya.com/blog/2015/08/common-machine-learning-algorithms/

[5] T. Mitchell, "Machine Learning", 1st edition. New York: McGraw-Hill, 1997.

[6] Hadi Hormozi, Elham Hormozi and Hamed Rahimi Nohooji, "The Classification of the Applicable Machine Learning Methods in Robot Manipulators", International Journal of Machine Learning and Computing, Vol. 2, No. 5, October 2012.

[7] "Dave and Decision Trees for NGS", by Dan Koboldt. Available: http://massgenomics.org/2008/10/dave-and-decision-trees-for-ngs.html.

[8] "A Complete Tutorial on Tree Based Modeling", by Analytics Vidhya,[Online]. Available:https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-python/

[9] Sunil Bhatia, Pratik Sharma, Rohit Burman, Santosh Hazari, Rupali Hande, "Credit Scoring using Machine Learning Techniques", International Journal of Computer Applications (0975 – 8887) Volume 161 – No 11, March 2017.

[10] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, "An Introduction to Statistical Learning: with Applications in R", Springer Texts in Statistics, Corr. 6th printing 2016 Edition.

[11] Max Kuhn, Kjell Johnson, "Applied Predictive Modeling", Springer, 2013th Edition.

[12] Trevor Hastie, Robert Tibshirani, Jerome Friedman, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction", Springer Series in Statistic, 2nd Edition.

[13] "Random Forests", by Leo Breiman and Adele Cutler, [Online]. Available: https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm

[14] Sania Jawaid, AnamSayeda and Naba Suroor, "Selection of Fittest Key Using Genetic Algorithm and Autocorrelation in Cryptography", Journal of Computer Sciences and Applications, 2015, Vol. 3, No. 2, 46-51

[15] S.N. Sivanandam, S.N. Deepa, "Introduction to Genetic Algorithms", Springer, ISBN 978-3-540-73189-4

[16] Sania Jawaid and Adeeba Jamal, "Generating the Best Fit Key in Cryptography using Genetic Algorithm", International Journal of Computer Applications (0975 – 8887) Volume 98 – No.20, July 2014

[17] Mariela Cerrada, Grover Zurita, Diego Cabrera, René-Vinicio Sánchez, Mariano Artés, ChuanLi, "Fault diagnosis in spur gears based on genetic algorithm and random forest", Mechanical Systems and Signal Processing, Science Direct, 70-71 (2016) 87–10

[18] Martin Junga, JakobZscheischlera, "A Guided Hybrid Genetic Algorithm for Feature Selection with Expensive Cost Functions", Procedia Computer Science, SciVerseScienceDirect, Volume 18, 2013, Pages 2337-2346

[19] Mohamed Bader-El-Den and Mohamed Gabe, "GARF: Towards Self-optimised Random Forests

[20] Bo-Suk Yang, Xiao Di and Tian Han, "Random forests classifier for machine fault diagnosis", Journal of Mechanical Science and Technology, Springer, 22 (2008) 1716~1725

[21] Dong-sheng Liu and Shu-jiang Fan, "Modified Decision Tree Algorithm Based on Genetic Algorithm for Mobile User Classification Problem", The Scientific World Journal, Volume 2014, Article ID 468324, 11 pages.

[22] Joseph Lee, "Huffman Data Compression", MIT Undergraduate Journal of Mathematics, May 23, 2007.

[23] R. Gil-Pita and X. Yao, "Using a Genetic Algorithm for Editing k-Nearest Neighbor Classifiers"

[24] Hamid Parvin, BehrouzMinaei, AkramBeigi, and HodaHelmi, "Classification Ensemble by Genetic Algorithms", Adaptive

and Natural Computing Algorithms, Springer, Volume 6593 of the series Lecture Notes in Computer Science pp 391-399

[25] OdedGoldreich, Foundations of Cryptography, Volume 1: Basic Tools, Cambridge University Press, 2001, ISBN 0-521-79172-3

[26] Harsh Bhasin and Nakul Arora, “Key Generation for Cryptography using Genetic Algorithm”

[27] Eric Bauer and Ron Kohavi, “An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants”, Machine Learning, Springer, July 1999, Volume 36, Issue 1, pp 105–139.