Data Classification Using Variation of Genetic Programming Fitness Function

SAFEENA RAZAQ*, MUHAMMAD WAQAR ASLAM, SARISH ABID, BASHARAT MANZOOR

Computer System Engineering Department, Mirpur University of Science and Technology (MUST) Mirpur, AJK, Pakistan

**Corresponding author's email: safeena.cse@must.edu.pk*

Abstract

Genetic Programming (GP) is a technique that deals with evolving computer programs using biologically inspired methods. GP is a set of instruction and a fitness function to evaluate the best solution. The objective of GP is to find a computer program capable of solving a predefined problem. GP has capability to select the useful features for the new generation and discard the unwanted features during evolution. In this paper, GP is used for real world classification problems. Five real world problems are used to evaluate the GP performance. In this paper, Gaussian Distribution Criteria, Standard Accuracy Method, Average Class Accuracy Method and Artificial Neural Networks (ANN) are used for the evaluation of fitness function for binary classification problems. A number of experiments are carried out to evaluate and compare the results obtained from GP. Results prove that GP (ANN) provide a better accuracy as compared to others methods.

Keywords: Genetic Programming (GP), Artificial Neural Networks (ANN), Binary classification

INTRODUCTION

Data classification is receiving increasing interest for their applicability in various real-life domains, such as medical diagnosis, image recognition and decision making. It is a difficult task because of uncertainty and unpredictability of variable data. Genetic Programming (GP) is a relatively new and fast developing algorithm, based on the process of biological evolution. It has been specially designed for automatically generating and developing computer programs. GP randomly generates initial population of individuals. Fitness values for individuals is calculated and the best individual (in term of fitness) will be selected for the next generation. GP has the intrinsic ability to select valuable features and discard others. Preprocessing step is not required for GP. This flexible and interesting technique has been applied to solve complex problem like classification. GP was found to be successful for classification problems and has emerged as a powerful tool for classifier evolution [1].

In the past, GP has been applied for some classification problems [2]–[5]. A survey was given on the application of genetic programming for classification purpose [2]. Four fitness functions to deal with the shortcomings of standard GP function and two approaches for binary classification with the unbalanced data were presented in [6] by Bhowan *et al*. It was reported that the evolved classifier performed poorly using standard GP fitness as compared to the improved fitness functions.

Genetic programming has also been successfully applied to real world applications, such as Optical Character Recognition (OCR) [7] and image and signal processing [8]. Handley [9] used GP to predict the shape of proteins. He was able to evolve programs which, using the protein's chemical composition, were able to predict whether each part of protein would have a particular geometric shape or not. Andre [10] used GP to evolve programs that were capable of storing a representation of their environment (map-making), and then using that representation to control a simulated robot in that environment. In his experiments, he was able to evolve solutions yielding 100% fitness. Handley [11] used GP to generate plans for a simulated mobile robot. Das *et al.*, [12] uses genetic programming to generate sounds and three-dimensional shapes.

An impressive GP-supported image processing algorithm was presented by Daida *et al.*, in [13]. Design of electrical circuits using GP was presented by Koza [14] where GP was successfully evolved for a large number of circuits with impressive results. Resistors, capacitors, inductors and functions for making parallel or series connections were used in the function pool. Many human competitive solutions have been synthesized using this system [15]. Some other popular examples are speech processing, communications in general, DNA pattern recognition, weather prediction, etc.

Nandi suggested a method for the diagnosis of breast cancer using the feature generated by GP [16] . A method for the classification of diabetes using a Modified Artificial Immune Recognition System2 (MAIRS2) is proposed in [17]. A number of techniques for handling imbalanced data sets using various data sampling methods and Meta Cost learners on six open-source data sets are presented in [18]. In algorithmic level methods, new algorithms are created which are adapted to the nature of imbalanced data sets. Gravitational Fixed Radius Nearest Neighbor Algorithm (GFRNN) is an algorithmic level method developed with the aim of enhancing k nearest neighbor classifier to acquire the ability of dealing with imbalanced data sets in [19]. Different aspects of imbalanced learning such as classification, clustering, regression, mining data streams and big data analytics, providing a thorough guide to emerging issues in these domains were discussed in [20].

Mohamad *et al.*, used the technique of Artificial Neural Network using back-propagation algorithm for classification of banknote authentication dataset [21]. Chen *et al.*, proposed a new feature selection method based on permutation to select features for high dimensional symbolic regression using GP [22]. In [23] Aslam and Nandi used GP for the diabetes classification. They used a modified version of GP called Comparative Partner Selection (CPS) for diabetes detection. They used Pima Indian Diabetes Dataset (PIDD) for this classification. Their results suggest that GP based classifiers can help in the diagnosis of diabetes disease.

Muni *et al.* [24] presented an online feature selection algorithm using GP. Feature Selection (FS) is a process to select the best features necessary to solve a problem. They used fitness function to achieve the readability of the trees extracted by the system. Through the output of the algorithm, they obtained a ranking of the features. A detailed analysis and comparison between different fitness functions in terms of performance and computational complexity are explained in this paper. Aslam *et al*. proposed an algorithm that uses GP with KNN classifier for automatic modulation classification [25]. This algorithm was used to identify BPSK, QPSK, 16QAM and 64QAM modulation.

In this paper, we have evaluated different fitness functions for few classification problems. The purpose of this study is to give a comparative analysis between different fitness functions in terms of classification accuracy and computational complexity.

FITNESS EVOLUTION METHODS

The fitness function is very important parameter of an individual to judge its position in population. The goal of fitness function is to help GP find, which individuals should be given chance to reproduce and multiply and which individuals should be removed from the population. In this section we will discuss different fitness function that can be used for binary classification in GP.

1. Gaussian Distribution

For getting the best possible GP solution for binary class, threshold value (overlapping point between two distributions) must be classified for each GP solution so that the accuracy can be maximized. In this technique, the probability distribution function for each class is found. The classification task is modeled as Gaussian distribution; its equation is given as follows:

$$\varphi(\mu c, \sigma c, x) = \frac{1}{\sigma c\sqrt{2\pi}} \exp(\frac{-(x-\mu c)^2}{2\sigma c^2}) \tag{1}$$

Where μc implies the mean of class and σc is the standard deviation of that class and x is the output of GP individual when evaluated on any input instance. Using the value of φ, all the samples lying on the wrong side of distribution will be classified as wrong distribution.

The steps for classifying the class labels are:

i. Firstly, using training data, calculate the output of GP individuals for classes.
ii. Calculate the mean and the standard deviation of each class from GP individual's output.

iii. Using corresponding mean and standard deviation, calculate the two φ values for every sample for each class.

iv. At last the class having larger value of φ, will be the class of this sample.

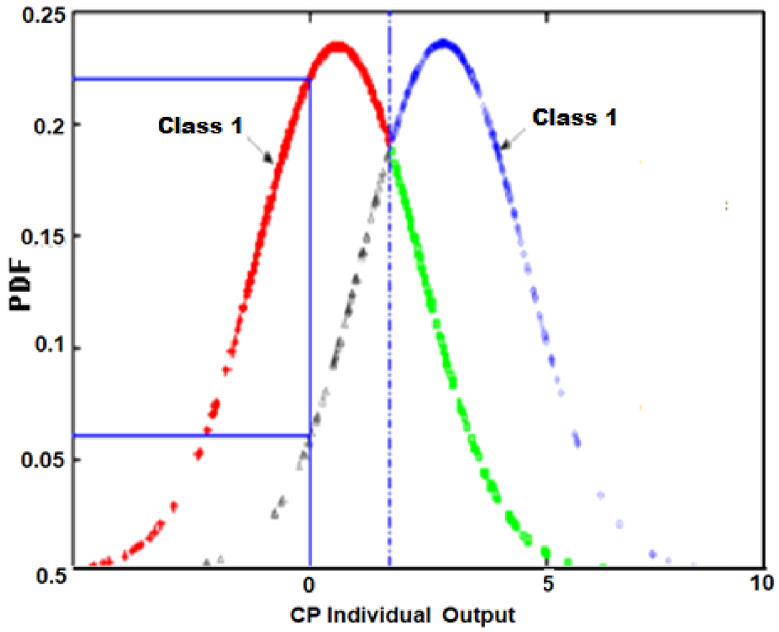

Figure 1: Gaussian distribution

Standard GP Fitness Function

Standard GP fitness function takes overall classification accuracy as fitness function of GP individual. This is simply the number of examples for all classes that are correctly classified by a classifier as a fraction of the total number of training examples for the classification problems. The classification accuracy is defined as

$$Acc = \frac{TP+TN}{TP+FN+TN+FP} \tag{2}$$

When calculating the overall accuracy in unbalanced datasets, this measure consider all examples as equally important, does not take into account that the number of examples in the minority class can be much smaller than in the majority class. Biased classifiers which have poor accuracy on the minority class but high majority class accuracy, can also have a high overall accuracy due to the influence of majority class examples. For example, in imbalance problem only 10% of samples belong to minority class, classifiers without discrimination capacity can give a high fitness by classifying all the cases as the member of the majority class. Accuracy is commonly used to train classifiers, and it can be applied to both binary and multi class classifiers. However, it is sensitive to class size distribution, and generally a poor choice when there is a significant class size variation [26].

Average Class Accuracy in Fitness

Average classification accuracy Ave, uses weighted average of the majority and minority class accuracy in fitness function.

$$Ave = \mathrm{W} \times \frac{TP}{TP+FN} + (1 - \mathrm{W}) \times \frac{TN}{TN+FP} \tag{3}$$

In Ave, the weighting factor controlling the contribution of any class in the fitness function. The value of W varies between 0 and 1. When W is 0.5, then the accuracy of both classes is considered as equally important. When $\mathrm{W} > 0.5$, minority class accuracy will contribute more in the fitness function than majority class. Similarly, majority class will contribute more when $\mathrm{W} < 0.5$. Many practitioners have applied equal weighting (i.e. $\mathrm{W} = 0.5$) in [27]–[29]. In this paper, the weighting configuration for W is between 0.1 to 0.9 with interval of 0.2 and its effect on the classification accuracy is investigated.

Artificial Neural Networks

ANNs, also known as neural networks is an information system paradigm that is inspired by a mathematical model of biological neural networks [30], [31]. In ANN we have a network of simple processing elements which are connected to each other via weighted links. Inputs are fed to the input unit and as a result of computation done in this unit, the outputs are produced.

The ANN has two modes of operations, training and testing mode. The training dataset has to be a representative collection of input-output examples. A highly favored algorithm known as back propagation algorithm is used to do the training of algorithm in which error function is used as a cost function for the modification of the weights of neurons [32]. Back propagation training is a gradient decent algorithm. It attempts to bring improvements in the performance of the neural net while decreasing the error along with its gradients. Sometimes, a validation phase is also required. A validation error is calculated using the validation data. If the validation error remains persistent for a predefined period, regardless of the fact that training error is reducing, the network is considered to be over-fitting the test data. If such situation is encountered, the training process is interrupted and the parameters of the network are reverted to the value that gave the minimum validation error. After completing the training and validation phase, trained network's performance is tested by making use of test data. A well-trained classifier should perform well for all data sets (training, validation and testing).

In this paper machine learning classifiers ANN is used with GP for evaluating the fitness of the individual. The inverse of the number of errors made by ANN is taken as fitness of individual where higher fitness means better individual. The best individual returned by GP

(ANN) is tested by ANN during testing phase. The results of different datasets obtained by using ANN with GP are mentioned in next section.

DATASETS

Pima Indian Diabetes Dataset

The Pima dataset contains 768 instances with two labels: diabetic or non-diabetic. This dataset consists of 768 patient's medical report. In total 768 individuals, all are female patients in which 500 individuals are non- diabetic and 268 individuals are diabetic. The dataset of 768 individuals is divided in two sets. One set is for training and the second set is for testing. The dataset consists of 8 attribute values. The result will be based on two possible outcomes, whether the patient is tested positive for diabetes or negative for non- diabetes.

Indian Liver Patient Dataset

The ILP dataset contains 583 instances with class distribution: liver patient and non-liver patient. In this dataset 416 instances are liver patient and 167 instances are non-liver patient, out of these 524 instances are used for training and 59 instances for testing. Liver patients are considered true positives and non-liver patient as true negative.

Ionosphere Dataset

The courtesy of system in Goose Bay, Labrador from which the radar data was collected. There is phase array of 16 high frequency antennas in this system. The total transmitted power of this system is of the order of 6.4 kilowatts. The target of this system was to hit free electron in the ionosphere. If some evidence of existence of some type of structure in the ionosphere is found it is stated as "Good" radar returns and for those that do not are "Bad" returns. Auto correlation functions are used to deal with received signals. There are 17 numbers of pulses for the system. Two attributes are used to describe instances in this dataset i.e., two attributes per pulse number. There are total of 351 instances for this dataset, in which the Bad class consists of 126 individuals and the Good class consists of 225 cases.

SPECT Heart Dataset

Single Photon Emission Computed Tomography (SPECT) heart contain 267 records from SPECT images. Two classes in this dataset, one is normal and the second is abnormal. There are 267 SPECT images set of patient's features in the dataset. The counts for continuous binary feature pattern is 22. All attributes have continuous integer values from 0 to 100. No missing values in this dataset. Out of 267 cases 55 cases were classified as Class 1, and 212 cases were classified in the second class. The total dataset is divided into 90:10 for training and testing datasets. The SPECT Heart dataset contains 267 instances with class distribution: Normal and Abnormal, where 55 instances are abnormal and 212 instances are normal.

Banknote Authentication Dataset

The data were obtained from the images taken from specimens of banknotes as genuine and fake. An industrial camera was used for its digitization. The dimensions of the concluded images were 400x 400 pixels. Resolution of these images was almost 660 dpi due to distance to the investigated object and object lens. Features were extracted using wavelet transform. The source data used 1372 samples with two problems to check class, whether the banknote is genuine or forged for authentication. Class 1 contains 610 cases and the rest of 762 cases are under class 2. The Banknote Authentication dataset contains 1372 individuals with class distribution: Forged and Genuine, where 610 instances are genuine and 762 instances are forged, out of these 1235 instances are used for training and 137 instances for testing. Forged are considered as true positive and genuine as true negative.

EXPERIMENTS AND RESULTS

In this section, results are explained using different fitness function on different dataset. Performance of each classifier is evaluated using three statistical measure i.e. classification accuracy, specificity and sensitivity. Specificity is specialized for majority class and sensitivity is for minority class.

Analysis of Average Class Accuracy Fitness Function

In this section the results of different datasets are presented through Average Class Accuracy Fitness Function. Average Class Accuracy Fitness Function is explained in previous section. The weighting factor W control the contribution of specificity and sensitivity in fitness evaluation. The weighting configuration for W is between 0.1 and 0.9 with interval of 0.2. The results of different datasets achieved through this method are presented in Table 1. As expected, sensitivity is high for $W > 0.5$ and specificity is high for $W < 0.5$. Based on these results it can be said that a weight favouring the majority class results in best overall classification accuracy.

Analysis of other GP Fitness Function on Test Data

This section focuses on the comparison of different fitness function using datasets from the UCI machine learning dataset repository [33]. Fitness function tries to separate the two classes. An overview of the datasets used is given in previous section. The experimental results depending on their best accuracy, sensitivity, specificity and average running time are specified in the Table 2. All the datasets are divided into training and testing data by the ratio of 90% and 10%. The 90% data is used to train the classifiers. After training the classifier GP algorithm is applied on test dataset to check its accuracy on testing data. All the results are calculated over 20 runs for 100 generation and 70 individuals. Table 2 shows the average accuracy with standard deviation for 20 runs, as well as the best accuracy, sensitivity and specificity for each fitness function.

For Pima dataset, using the GP (ANN) as fitness function obtained the highest accuracy of 94.8052% whereas Gaussian distribution and standard accuracy achieved 88% and 88.31%

respectively. GP (ANN) is also best suitable for ILPD dataset. Through this method we achieved the best accuracy of 94 %. Although GP (ANN) produced the highest accuracy for both datasets, but it does not provide a good balance between sensitivity and specificity. Specificity that is related to majority class is higher than the sensitivity.

Table 1: Analysis of Average Class Accuracy Fitness Function on Test Data Over 20 Runs

Dataset	Weight W	Average Accuracy	Best Accuracy	Sensitivity	Specificity	Training time
Pima	0.1	94.8563±0.6231	95.9778	62.96	89.40	2.6hrs
	0.3	87.9533±0.9740	89.7111	67.78	96.60	1hr 50min
	0.5	85.7963±1.1924	86.5926	82.59	89.00	1hr 38min
	0.7	84.2207±1.0071	85.4047	89.63	71.60	1.5hrs
	0.9	91.1067±3.0290	93.8000	98.52	24.40	3hrs
ILPD	0.1	98.3333±0.3723	98.0952	83.33	100	1hr 40min
	0.3	94.9286±1.0885	95.7143	83.09	100	1.5hr
	0.5	92.3389±0.9631	93.4874	87.62	97.06	1hr 13min
	0.7	91.3137±3.0546	94.9020	92.38	88.83	37min
	0.9	95.1261±3.5796	97.6471	98.33	65.30	3hrs
Ionosphere	0.1	100±0	100	100	100	6hrs
	0.3	99.3182±1.1409	100	97.73	100	5hrs
	0.5	96.9697±3.9800	100	93.94	100	4.5hrs
	0.7	95.6643±3.3654	100	95.45	96.15	3hrs
	0.9	96.4161±2.6295	100	97.73	84.62	3.5hrs
SPECT Heart	0.1	89.1191±4.1164	93.3333	35.00	94.76	1hr 20min
	0.3	84.8333±4.9348	96.6667	78.33	87.62	1hr 20min
	0.5	88.4524±5.8604	97.6190	95.00	81.90	2.3hrs
	0.7	91.5476±4.7847	95.7143	98.33	75.71	1.8hrs
	0.9	97.9524±1.7250	99.0476	100	79.52	2.5hrs
Banknote	0.1	91.0326±1.5806	91.9672	20.98	98.81	31min
	0.3	86.2532±6.4236	99.0789	86.72	86.05	8min
	0.5	86.4420±10.3486	98.0263	93.28	79.61	10 min
	0.7	85.4761±6.6701	98.8185	87.54	80.66	5min
	0.9	89.7591±5.0515	95.6299	93.61	55.13	6min

Standard accuracy method and Gaussian distribution method are best for this dataset because these methods provide a desirable stability between sensitivity and specificity. Minority class is accurately classified through these methods, and the obtained accuracy through these methods is also good that is 91%. For Ionosphere dataset, we achieved the best accuracy that is 100% through GP (ANN) and also using Average Accuracy method. It accurately classified the minority class because the sensitivity that is related to minority class is also 100%. Gaussian distribution obtained an accuracy of 96.57% with the best value (from the 20 runs) of 100% for this dataset.

From Table 2, it is clear that Average Accuracy method and GP (ANN) are best suitable for SPECT Heart dataset. We achieved the best accuracy through GP (ANN) that is 96.2963% and 99.0476 % through Average Accuracy method for W = 0.9. For this dataset, sensitivity is always higher than the specificity. GP (ANN) is best suitable for Banknote Authentication dataset. We achieved 100% accuracy through GP (ANN) method. Sensitivity and specificity for this data is also 100%. Gaussian distribution is also best for this dataset. Both methods provide a good balance between sensitivity and specificity. These result shows that the best accuracy, sensitivity and specificity are achieved by using GP (ANN) with 10 hidden layers.

Table 2: Analysis of GP Fitness Function on Test Data Over 20 Runs

Dataset	Fitness Function	Average Accuracy	Best Accuracy	Sensitivity	Specificity	Training Time
Pima	Gaussian	86.8831±1.4292	88.3117	77.04	92.20	1hr 21min
	Acc	86.4935±1.8569	88.3117	74.81	92.60	55min
	ANN	93.5065±0.8214	94.8052	71.60	91.67	3hrs
ILPD	Gaussian	91.6949±4.0307	96.6102	91.91	91.18	1hr 22min
	Acc	91.5254±2.2599	93.2203	91.43	91.77	1hr 10min
	ANN	92.6553±1.7505	94.9153	66.67	91.27	3hrs
Ionosphere	Gaussian	96.5714±3.9955	100	95.91	97.69	1.4hrs
	Acc	94.8572±4.4263	100	95.00	94.62	1hr
	ANN	100±0	100	100	100	4hrs
SPECT Heart	Gaussian	82.9630±1.9126	85.1852	71.67	86.19	40min
	Acc	84.4445±3.4035	88.8889	71.67	88.57	52min
	ANN	95.6790±1.5120	96.2963	90.48	78.97	3hrs
Banknote	Gaussian	94.8905±3.9682	97.8102	96.06	93.95	1.5hrs
	Acc	86.8613±4.2030	95.6204	85.90	84.21	30min
	ANN	100±0	100	100	100	3.5hrs

Complexity Analysis

GP classifier may take a long training time on the order of hours or even days for classification. All the fitness function takes a lot of training time. As our results show that the Gaussian method takes approximately one and half hour to train the datasets. Standard accuracy fitness function takes one hour for training the datasets, whereas GP with ANN and Average class accuracy fitness function take approximately three to four hours for training. This training time is for 20 runs. Based on the results GP with ANN and Average accuracy methods take a lot of training time whereas the Standard accuracy fitness function takes less time to train the datasets. The final solution produced by GP depends on the training time. The complexity of the classifier depends on the GP returned solution and the complexity of the classifier used for the testing phase.

CONCLUSION

The experimental results have shown that different fitness function used for classification purpose behave differently on different datasets. Gaussian method and standard accuracy method perform poorly for Pima Indian diabetes and SPECT heart dataset, resulting in low sensitivity and higher specificity. Although the overall accuracy achieved by Gaussian method and standard accuracy method is good, but the low sensitivity makes it trivial. Both methods perform well for remaining datasets because it provides a balance between specificity and sensitivity. The classification technique which has shown the high accuracy rate and sensitivity for a dataset has been chosen as the best technique for that dataset. From results, it can be concluded that GP with ANN is most suitable for given tasks. Overall ANN has achieved remarkable performance with highest accuracies, and provides a good balance between sensitivity and specificity for most of the datasets. Average accuracy method is close to second. In all respect GP with ANN is performing well; Hence GP with ANN is recommended for binary classification irrespective of dataset. As far as the complexity is concerned, GP with ANN takes more training time, and ANN complexity increases with increasing hidden layers size.

REFERENCES

[1] H. Jabeen and A. R. Baig, “Review of classification using Genetic Programming,” *Int. J. Eng. Sci. Technol.*, vol. 2, no. 2, pp. 94–103, 2010.

[2] P. G. Espejo, *et al.*, “A survey on the application of Genetic Programming to classification,” *IEEE Trans. Syst. Man, Cybern. Part C Appl. Rev.*, vol. 40, no. 2, pp. 121–144, 2010.

[3] J. Eggermont, *et al.*, “A comparison of genetic programming variants for data classification,” in *Proc. Third Symposium on Advances in Intelligent Data Analysis*, 1999, pp. 281–290.

[4] C. Gathercole and P. Ross, “Dynamic training subset selection for supervised learning in Genetic Programming,” in *Parallel Problem Solving from Nature — PPSN III. PPSN 1994. Lecture Notes in Computer Science*, Y. Davidor, H. P. Schwefel, and R. Männer, Eds. Springer, Berlin, Heidelberg, 1994, pp. 312–321.

[5] H. Gray, "Genetic Programming for classification of medical data," in *Late Breaking Papers at the 1997 Genetic Programming Conf.*, 1997, pp. 291–297.

[6] U. Bhowan, *et al.*, "Developing new fitness functions in Genetic Programming for classification with unbalanced data," *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 42, no. 2, pp. 406–421, 2012.

[7] D. Andre, "Automatically defined features: The simultaneous evolution of 2-dimensional feature detectors and an algorithm for using them," in *Advances in Genetic Programming, Chapter 23*, vol. 1, MIT Press Cambridge, MA, 1994, p. 477.

[8] W. A. Tackett, "Genetic programming for feature discovery and image discrimination," in *Proc. 5th Int. Conf. Genetic Algorithms, ICGA-93,* Morgan Kaufmann Publishers, 1993, pp. 303–309.

[9] S. Handley, "Automatic learning of a detector for alpha-helices" in *Proc. 5th Int. Conf. Genetic Algorithms, ICGA-93,* Morgan Kaufmann Publishers, 1993, pp. 271–278.

[10] D. Andre, "Evolution of mapmaking: learning, planning, and memory using Genetic Programming," in *Proc. First IEEE Conf. Evolutionary Computation. IEEE World Congress on Computational Intelligence*, 1998, pp. 250–255.

[11] S. G. Handley, "The Automatic generations of plans for a mobile robot via Genetic Programming with automatically defined functions," in *Advances in Genetic Programming*, vol. 18, Kenneth E. Kinnear, Ed. MIT Press, Cambridge, MA, 1994, pp. 391–407.

[12] S. Das, *et al.*, "A Genetic Programming application in virtual reality," in *Proc. 1994 IEEE 3rd Int. Fuzzy Systems Conf.* 1994, pp. 480–484.

[13] J. M. Daida, *et al.*, "Computer-Assisted Design of image classification algorithms: Dynamic and Static Fitness Evaluations in a Scaffolded Genetic Programming Environment," in *Genetic Programming 1996: Proc. First Annual Conf.* 1996, pp. 279–284.

[14] J. R. Koza and J. P. Rice, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.

[15] J. R. Koza, *et al.*, *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*. Kluwer Academic Publishers, 2003.

[16] H. Guo and A. K. Nandi, "Breast cancer diagnosis using Genetic Programming generated feature," *Pattern Recognit.*, vol. 39, no. 5, pp. 980–987, 2006.

[17] Y. A. Christobel and P. Sivaprakasam, "A new Classwise k Nearest Neighbor (CKNN) Method for the classification of diabetes dataset," *Int. J. Eng. Adv. Technol.*, vol. 2, no. 3, pp. 396–400, 2013.

[18] R. Malhotra and M. Khanna, "An empirical study for software change prediction using imbalanced data," *Empir. Softw. Eng.*, vol. 22, no. 6, pp. 2806–2851, 2017.

[19] B. Nikpour, *et al.*, "Proposing new method to improve gravitational fixed nearest neighbor algorithm for imbalanced data classification," in *2nd Conf. Swarm Intelligence and Evolutionary Computation (CSIEC)*, 2017, pp. 6–11.

[20] B. Krawczyk, "Learning from imbalanced data: Open challenges and future directions," *Prog. Artif. Intell.*, vol. 5, no. 4, pp. 221–232, Nov. 2016.

[21] N. S. Mohamad, *et al.*, "Banknote authentication using Artificial Neural Network," *Sci. Int.*, vol. 26, no. 5, pp. 1865–1868, 2014.

[22] Q. Chen, *et al.*, "Feature selection to improve generalization of Genetic Programming for high-dimensional symbolic regression," *IEEE Trans. Evol. Comput.*, vol. 21, no. 5, pp. 792–806, Oct. 2017.

[23] M. W. Aslam and A. K. Nandi, "Detection of diabetes using Genetic Programming," in *18th European Signal Processing Conf. (EUSIPCO-2010)*, 2010, pp. 1184–1188.

[24] D. P. Muni, *et al.*, "Genetic Programming for simultaneous feature selection and classifier design," *IEEE Trans. Syst. Man Cybern. part B*, vol. 36, no. 1, pp. 106–117, Feb. 2006.

[25] M. W. Aslam, *et al.*, "Automatic digital modulation classification using Genetic Programming with K-Nearest Neighbor," *Mil. Commun. Conf. 2010 - Milcom 2010*, pp. 1731–1736, 2010.

[26] G. Patterson and M. Zhang, "Fitness functions in Genetic Programming for classification with unbalanced data," in *AI 2007: Advances in Artificial Intelligence. AI 2007. Lecture Notes in Computer Science*, vol. 4830, M. A. Orgun and J. Thornton, Eds. Berlin, Heidelberg: Springer, 2007, pp. 769–775.

[27] H. F. Gray, *et al.*, "Genetic Programming for classification of brain tumours from Nuclear Magnetic Resonance biopsy spectra," in *Proc. First Annual Conf. Genetic Programming*, MIT Press, Cambridge, MA, 1996, pp. 424–430.

[28] J. Doucette and M. I. Heywood, "GP classification under imbalanced data sets: Active sub-sampling and AUC approximation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4971 LNCS, 2008, pp. 266–277.

[29] S. M. Winkler, *et al.*, "Using enhanced Genetic Programming techniques for evolving classifiers in the context of medical diagnosis," *Genet. Program. Evolvable Mach.*, vol. 10, no. 2, pp. 111–140, Jun. 2009.

[30] S. S. Haykin, *Neural networks: A comprehensive foundation*, 1st ed. Prentice Hall, 1994.

[31] R. Rojas, *Neural Networks- A Systematic Introduction*. Springer, 1996.

[32] D. E. Rumelhart and J. L. Mcclelland, "Learning internal representations by error propagation," in *Parallel distributed processing: Explorations in the microstructure of cognition.* vol. I, J. A. Feldman, P. J. Hayes, and D. E. Rumelhart, Eds. MIT Press,Cambridge, MA, 1986.