ROBUST QAM CLASSIFICATION USING GENETIC PROGRAMMING AND FISHER CRITERION

Muhammad Waqar Aslam, Zhechen Zhu and Asoke Kumar Nandi

The University of Liverpool,
Department of Electrical Engineering and Electronics,
Brownlow Hill, Liverpool, L69 3GJ, U.K.
{m.w.aslam, z.zhu, a.nandi}@liverpool.ac.uk

ABSTRACT

Automatic modulation recognition has seen increasing demand in recent years. It has found many applications in wireless communications, including both civilian and military applications. It is a scheme to identify automatically the modulation type of received signal by observing data samples of received signals in the presence of noise. In this paper a combination of genetic programming (GP) and Fisher criterion is proposed for classification of QAM modulation schemes for the first time. This method appears to be both efficient and robust. Due to an increase in importance of QAM modulations schemes in recent times we have used QAM for classification purpose. The modulations considered here are QAM16 and QAM64. Simulations and results show that the performance achieved using GP are better than other methods presented so far.

1. INTRODUCTION

Automatic modulation recognition (AMR) is a digital signal processing technique that automatically tells us about the modulation type of an incoming signal. It is an intermediate step between signal detection and demodulation. It has various civilian and military applications [1]. For civilian applications it is used for signal confirmation, spectrum management and interference identification. It is also used for electronic warfare, surveillance and threat analysis, target acquisition, homing and jamming for military applications.

Software defined radio (SDR) [2]-[3] has also seen an increasing demand in recent times. It allows implementation of creative transceiver designs, which can adapt to communication channel requirement. As SDR has to deal with a variety of communication systems so blind recognition of modulation of incoming signal is required. AMR system can be used at the front end of SDR to solve this problem. Cognitive radio (CR) [4] is also another emerging technology that makes use of AMR system. CR achieves efficient spectrum utilization by making use of empty frequency bands. Dobre et al. [5] gave a survey on different modulation classification techniques available in the literature. Two steps are involved in designing a modulation classifier: signal preprocessing and selection of modulation classifier. Regarding the second step, the process of AMR can be divided in two categories, likelihood based (LB) [6]-[10] approach and feature-based (FB) [11]-[20] approach. The LB approach is based on likelihood function of an incoming signal and decision is made comparing the likelihood ratio against a threshold. Although this method is optimal in Bayesian sense, it is computationally complex. Nandi and Azzouz [6] used LB approach and Artificial Neural Network (ANN) for classification of wide range of modulation types. They reported the classification accuracy of 94% at an SNR of 15 dB. Hong and Ho [7] used Bayes technique for classification of BPSK and QPSK signals without a priori knowledge of the received signal level. Wong and Nandi [8] used Maximum Likelihood (ML) approach with estimation of SNR, presenting an estimated ML (EsML) classifier. They used minimum distance classifier to reduce the complexity of classifier. They also used blind source separation technique for rectifying carrier offset problem. Wei and Mendel [9] used ML method for classification of digital quadrature modulations. They reported classification accuracy of 83% at 5 dB SNR between 16-point V.29 and QAM16. Wong, Ting and Nandi [10] used Naïve Bayes classifier for classification of BPSK, QPSK, QAM16 and QAM64. In FB approach features are extracted from an incoming signal and decision is made based on the values of these features. This method may not be optimal but reduces computational complexity and can produce optimal results if designed properly. Swami and Sadler [12] used fourth order cumulants for classification of digital modulation schemes. The method presented was robust to carrier phase and frequency offset. Chaithanya and Reddy [13] used combination of moments for classification of a big set of modulation techniques. In [14] Hong used Haar Wavelet transform technique for QAM classification and reported 94% classification accuracy at an SNR of 8 dB. Mirarab and Sobhani [15] used higher order cumulants for classification of PSK, ASK and QAM signals. The method presented was robust to frequency offset. In [16] Dobre, Ness and Su used higher order cyclic cumulants for classification of QAM, PSK and ASK signals. They achieved classification accuracy of 90% at an SNR of 10 dB.

In this study we use GP [21] for classification of QAM signals. GP has been used previously for classification purposes. Espejo, Ventura, and Herrera [22] gave a survey about the application of GP for classification problems, showing different ways in which GP can be used for developing accurate classifiers. Eggermont, Eiben and Hemert [23] presented a comparative study on different variations of GP for binary data classification problems. Kishore et al. [24] used GP for multi category pattern classification. They presented the idea of dividing an *n*-class classification problem into *n* two-class classification problems. Zhang, Jack and Nandi [25] used GP for feature generation and K-nearest neighbour (KNN) for classification purpose.

We have used previously GP with KNN for classification of different modulation schemes [17]-[18]. In that study GP was used for evolution of features and KNN was used for fitness evaluation of individuals. With the use of KNN, GP was able to do multi-class classification. The modulations used were BPSK, QPSK, QAM16 and QAM64. In this study we are extending that research but instead of doing four-class classification with GP-KNN, only three-class classification is done with GP-KNN and later two class classification is done with a different fitness criteria. It is demonstrated through extensive simulation that using this different fitness criterion for two-class classification improves the classification accuracy. GPLab toolbox has been used here for simulation purpose (http://gplab.sourceforge.net/).

The paper is organised as follows: Section 2 explains the signal model and features used. Section 3 explains basics of genetic programming. It also explains our proposed method. Section 4 is about the experiments and results and conclusion is drawn in Section 5.

2. SIGNAL MODEL AND FEATURES USED

2.1 Signal Model

AMR algorithm proposed uses information extracted from the received modulated signal. A general expression for the received baseband waveform can be written as

$$y(n) = s(n) + g(n) \qquad (1)$$

where $y(n)$ is complex baseband envelope of the received signal, $g(n)$ is additive white and Gaussian noise (AWGN) and $s(n)$ can be written as

$$s(n)=Ae^{j(2\pi f_o nT+\theta_n)} \sum_{l=-\infty}^{\infty} x(l)h(nT - lT + \epsilon_T T) \qquad (2)$$

where $x(l)$ is input symbol sequence, A is unknown amplitude, f_o is constant carrier frequency offset, T is the symbol spacing, θ_n is the phase jitter, h *(.)* is residual baseband channel effects and ϵ_T is the timing error. The channel is assumed to be equalized and residual channel effect is negligible. The conditions are assumed to be ideal with the presence of AWGN only.

2.2 Features Used

Feature based approach has been used here for AMR. Fourth and sixth order cumulants of received signals have been used as the underlying features. As cumulants are made up of moments, so various moments have been used as features too. For a complex valued stationary signal the cumulants can be defined as shown below [15]

$$C_{40} = cum(y(n),y(n),y(n),y(n)) = M_{40} - 3{M_{20}}^2$$
$$C_{41} = cum(y(n),y(n),y(n),y^*(n)) = M_{40} - 3M_{20}M_{21}$$
$$C_{42} = cum(y(n),y(n),y^*(n),y^*(n)) = M_{42} - |M_{20}|^2 - 2{M_{21}}^2$$
$$C_{60} = cum(y(n),y(n),y(n),y(n),y(n),y(n))$$
$$= M_{60} - 15M_{20}M_{40} + 30{M_{20}}^3$$
$$C_{61} = cum(y(n),y(n),y(n),y(n),y(n),y^*(n))$$
$$= M_{61} - 5M_{21}M_{40} - 10M_{20}M_{41} + 30{M_{20}}^2M_{21}$$
$$C_{62} = cum(y(n),y(n),y(n),y(n),y^*(n),y^*(n))$$
$$= M_{62} - 6M_{20}M_{42} - 8M_{21}M_{41} - M_{22}M_{40} + 6{M_{20}}^2M_{22} + 24{M_{21}}^2M_{20}$$
$$C_{63} = cum(y(n),y(n),y(n),y^*(n),y^*(n),y^*(n))$$
$$= M_{63} - 9M_{21}M_{42} + 12{M_{21}}^3 - 3M_{20}M_{43} - 3M_{22}M_{41} + 18M_{20}M_{21}M_{22} \qquad (3)$$

M_{pq} represents the moment of a signal which is defined as

$$M_{pq} = E[y(k)^{p-q}(y^*(k))^q] \qquad (4)$$

3. THE PROPOSED METHOD

3.1 Genetic Programming

GP is a machine learning methodology inspired by the biological evolution to find computer programs that perform user-defined task [21]. The following are the main steps in GP evolution.

(1) GP starts with a population of randomly generated individuals.

(2) The individual are given a fitness value which is the ability of an individual to solve the problem.

(3) GP produces new individuals in every generation using genetic operators on current individuals. These new individuals are created either by swapping the parent individuals (crossover) or by mutating a parent individual (mutation).

(4) The individuals with better fitness values go to the next generation.

(5) Step (3) and (4) are repeated till we get 100% correct result or maximum number of generations set by the user is reached. In the end fittest of all the individuals is the solution of the problem. This process can be represented by an equation

$$g_{t+1} = g_o(f(g_t)) \qquad (5)$$

where $g_t$is current generation, function f chooses the fitter individual in the current generation, g_o produces new individuals using genetic operators and g_{t+1} is the new generation being produced. The computer programs (individuals) can be represented by different ways. One of the common representations is tree representation which has been used here as well. A tree has three main parts; leaf nodes, intermediate nodes and root. The leaf nodes represent the inputs given to the tree, intermediate nodes represent functions operating on inputs and root represents the output of the tree. Some of the basic terms of GP are explained below.

3.2 Some basic terms of GP

3.2.1 Fitness Function

This is the most important parameter of an individual. A fitness function is an attribute of an individual representing the strength of an individual to solve the given problem. It is a user-defined function and depends on the nature of the problem. Whenever a new generation is created fitness values are calculated and a decision is made based on this value that which individuals will go to the next generation.

3.2.2 Function Pool
Function pool contains all the functions used by individuals. These functions are the operations done on input values. A function pool can contain any type of function depending on the nature of problem. For logical problems AND, OR, etc. are used and for non-linear problems non-linear functions could be used. The function pool used is given in Table 1.
3.3.3 Genetic Operators
Genetic operators operate on a generation of individuals to produce next generation. There are many types of genetic operators and some of them are given below.
Crossover
In this genetic operator two parents are crossed with each other to make child. A random node is chosen on the parents and the sub-trees downwards from that node are swapped with each other to make child.
Mutation
This genetic operator operates on a single parent. It chooses a random node on the parent tree and alters the sub-tree downward from that node in a random way to produce off-spring.
Reproduction
This is similar to cloning. This genetic operator just copies the individual in next generation without any change.

3.3 System Used and Fitness Evaluation

Initially all the features (cumulants and moments) are given as input to GP. GP will try to find the perfect combination of features and different functions to make a new feature good enough to distinguish between the modulation schemes. The process is guided by a fitness function defined by user. In each passing generation GP will try to improve the performance of individuals. Prior to the classification of QAM16 and QAM64 using this method we did classification of these modulations using a different technique mentioned in our research [17]-[18]. In that research we used four modulations BPSK, QPSK, QAM16 and QAM64. These were divided into three groups as BPSK, QPSK and QAM16/QAM64. At first stage training for three class-classification was done using GP-KNN. At the second stage training for QAM16 and QAM64 was also done using GP-KNN.

In this study we have used the same training strategy for first stage but for second stage instead of using GP-KNN we have used GP with different fitness criterion shown in equation 6. A block diagram of proposed system is shown in Fig. 1. The purpose of introducing KNN at first stage was to make GP a multi-class classifier and as second stage involves only two classes we do not need KNN. We can do that classification using simple fitness criterion rather than using KNN for fitness evaluation which complicates the fitness evaluation. Once the training is finished the best tree is tested with KNN. For testing only KNN is used. The results show that using this different fitness evaluation criterion improves the classification results. As there are two modulations at second stage so there will be two distributions produced by GP-tree, one for each modulation. Aim is to increase the distance between these modulations (i.e increase the inter class variance) and decrease the distance between points within each modulation (i.e decrease the intra class variance).

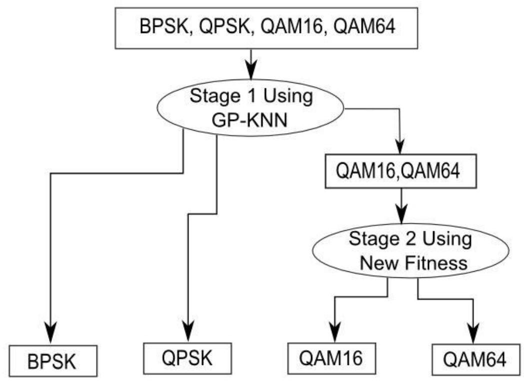

Fig. 1. System block diagram for training

So we adopted a simple fitness function based on fisher criterion which serves this purpose. The fitness function used is given in equation 6.

$$\text{Fitness} = \left[\frac{|\ m_1 - m_2|}{\sqrt{(\sigma_1{}^2 + \sigma_2{}^2)}}\right] \tag{6}$$

where m_1, m_2 are the means of two distributions for two modulations and σ_1, σ_2 represent standard deviations of two modulations. This fitness function tries to increase the distance between the means of two classes while minimizing the variance of two classes. Example outputs of a GP-tree for these two modulation schemes are shown in Fig. 2.

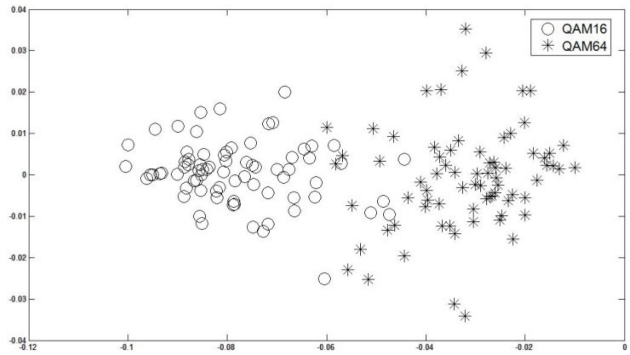

Fig. 2. Feature Distribution for QAM16 and QAM64 at an SNR of 10 dB and 1024 number of samples as returned by a GP-tree.

4. RESULTS AND DISCUSSIONS

4.1 Experiments and Results

The number of generations used during GP evaluation is 100 and the number of candidates used is 50. The fitness function used is given in equation 6. Rest of the parameters used by GP are given in Table 1.

All the moments and cumulants are given as input to GP. GP tries different combination of these inputs in each generation in search of best combination. This process is guided by fitness function which helps GP to choose those individuals which have the capability to make two modulations distributions apart. After reaching 100th generation GP returns the best tree produced during evolution process. That best tree is then tested with test data using KNN classifier and results are presented in Table 2, 3 and 4.

Table 1
Parameters used for the experimental work.

Parameter	Standard value
No. of Generations	100
Population Size	50
Function Pool	{+, -, *, Sin, Cos, Log}
Terminal Pool	Moments and cumulants
Genetic Operators	{Crossover, mutation}
Operator Probabilities	{0.9,0.1}
Tree Generation	Ramped half-n-half
Initial Maximum Depth	6
Maximum Depth	28
Selection Operator	Roulette
Elitism	Half-elitism

The classifier used here for testing is KNN. Results of the 3-class classification BPSK, QPSK and QAM16/QAM64 are the same as given in [18]. MATLAB/GPLab has been used here for experiments. The number of training experiments done was 25 for each SNR and number of samples combination. As there were 25 individuals in each generation so 625 trees were produced. Out of these 625 trees the best tree was chosen for evaluating test data. The SNRs and number of samples chosen for testing were 5 dB, 10 dB, 15 dB, 20 dB and 512, 1024, and 2048 respectively. The number of test samples for each modulation is 1000 for each modulation. We have used 100 test runs to get the average classification accuracy so total number of test sample becomes 100,000. Results for QAM16 and QAM64 using new fitness are given in Table 2. One can see the standard deviation is really low for all cases which shows the robustness of method proposed.

Table 2
Accuracy of performance results for GP-tree with 1-standard deviation in 2-class classification (QAM16 and QAM64) using different SNRs and number of samples.

SNRs	512	1024	2048
5 dB	68.1 ± 1.0%	74.6 ± 0.9%	86.8 ± 0.7%
10 dB	89.7 ± 0.7%	96.4 ± 0.4%	99.3 ± 0.2%
15 dB	97.7 ± 0.3%	99.7 ± 0.1%	100.0 ± 0.0%
20 dB	99.9 ± 0.1%	100.0 ± 0.0%	100.0 ± 0.0%

The performance of GP-tree at 10 dB SNR and at 1024 number of samples is given in Table 3 in the form of confusion matrix. It is clear from the Table that BPSK and QPSK are easy to classify as compared to QAM16 and QAM64. The performance of GP-tree at 1024 number of samples and at different SNRs is given in Table 4. One can see from Table 4 that performance of GP-tree is always above 95% from SNR 10 dB to 20 dB.

4.2 Discussions

In [10] Wong, Ting and Nandi used Naïve Bayes classifier for detection of BPSK, QPSK, QAM16 and QAM64. They reported a classification accuracy of 94.4% at an SNR of 10 dB and 1024 number of samples for their bigger set of modulations. For the same settings we achieved a percentage of 96.4% for QAM16 and QAM64. Again it is to be noted that classification of BPSK and QPSK is relatively easy as compared to QAM16 and QAM64 modulations. Swami and Sadler [12] reported a performance of 90% for QAM16 and QAM64 but the signals were assumed to be noiseless and also the number of symbols used was more than 10,000. We have achieved 100% performance at 15 dB and 2048 number of samples.

Table 3
Confusion matrix for 1024 number of samples and at 10 dB SNR for 4-class classification.

Modulations	BPSK	QPSK	QAM16	QAM64
BPSK	100,000	0	0	0
QPSK	0	100,000	0	0
QAM16	0	0	97,949	5,227
QAM64	0	0	2,051	94,773

Mirarab and Sobhani [15] reported 94% accuracy at an SNR of 15 dB and 2000 number of samples for a bigger set of modulations. For the same SNR and 2048 number of samples we achieved 100% classification accuracy for our smaller set. Dobre, Ness and Su [16] reported 70% classification accuracy using 2000 samples and at 10 dB SNR for classification of QAM16 and QAM64. For the same settings we achieved 96.4% accuracy. Orlic and Dukic [19] reported accuracy of 70% for QPSK, QAM16 and QAM64 at 15 dB SNR and using 2000 number of samples in multipath fading channel. We have achieved 100% accuracy at the same SNR and 2048 number of samples in AWGN channel.

Table 4
Performance of GP-Tree at 1024 number of samples and at different SNRs.

SNR	Modulation Type	QAM16	QAM64
5 dB	QAM16	78.3%	29.1%
	QAM64	21.7%	70.9%
10 dB	QAM16	97.9%	5.2%
	QAM64	2.1%	94.8%
15 dB	QAM16	99.9%	0.5%
	QAM64	0.1%	99.5%
20 dB	QAM16	100.0%	0.0%
	QAM64	0.0%	100.0%

Lanjun and Canyan [20] reported classification accuracy of 70% at an SNR of 10 dB using 8192 number of samples for QAM16 and QAM64. They also reported that classification between these two modulations was difficult due to their similar constellation diagram. For the same setting of SNR and at 2048 number of samples we have achieved 99.3% classification accuracy. Chaithanya and Reddy [13] reported classification accuracy of 90% using 1000 samples at 10 dB SNR. For the same SNR and 1024 number of samples we achieved 96.4% classification accuracy.

5. CONCLUSION

This paper addresses the problem of AMR in the presence of AWGN. A machine learning methodology has been used for classification. We used GP with Fisher criterion for this purpose. GP automates the process of combining the existing features to make new feature. Due to importance of QAM modulation schemes QAM16 and QAM64 have been chosen for classification. Recently published results have reported the difficulty in classifying QAM16 from QAM64 due to their similar shapes. The results from this study demonstrate that GP solves this problem. GP combines existing features to make new feature which is good enough to distinguish between these two modulations. The results reported here are better than other results published so far.

6. ACKNNOWLEDGEMENTS

Muhammad Waqar Aslam would like to acknowledge the financial support of The University of Azad Jammu & Kashmir Pakistan. Zhechen Zhu would like to thank the University of Liverpool (U.K.) for financial support.

REFERENCES

[1] E. E. Azzouz and A.K. Nandi, Automatic Modulation Recognition of Communication Signals. Boston, MA: Kluwer, 1996.

[2] K. E. Nolan, L. Doyle, P. Mackenzie, and D. O'Mahony, "Modulation scheme classification for 4G software radio wireless networks", in *Proc. IASTED International Conference on Signal Processing, Pattern Recognition,and Applications*, 2002, pp. 25-31.

[3] N. Alyaoui, H. Ben Hnia, A Kachouri, M. Samet, "The modulation recognition approaches for software radio", *2nd International Conference on Signals, Circuits and Systems (SCS)*, 2008, pp. 1-5.

[4] B. Ramkumar, "Automatic modulation classification for cognitive radios using cyclic feature detection", *IEEE Circuits and Systems Magazine*, vol. 9, 2009, pp. 27-45.

[5] Octave A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, " A survey of automatic modulation classification techniques: classical approaches and new trends", *IET Commun*., vol. 1, 2007, pp. 137-156.

[6] A. K. Nandi and E. E. Azzouz, "Algorithms for automatic modulation recognition of communication signals", *IEEE Transactions on Communications*, vol. 46, 1998, pp. 431-436

[7] L. Hong and K. C. Ho, "Classification of BPSK and QPSK signals with unknown signal level using the Bayes technique", in *Proc. IEEE ISCAS,* vol. 4, 2003, pp. IV.1-IV.4.

[8] M. L. D. Wong and A. K. Nandi, "Semi-blind algorithms for automatic classification of digital modulation types", *Digital Signal Processing*, vol. 18, 2008, pp. 209-227.

[9] W. Wei and J. M. Mendel, "Maximum-likelihood classification for digital amplitude-phase modulations", *IEEE Trans. Commun*., vol. 48, 2000, pp. 189-193.

[10] M. L. D. Wong, S. K. Ting and A. K. Nandi, "Naïve Bayes classification of adaptive broadband wireless modulation types with higher order cumulants", in *Proc. of the International Conference on Signal Processing and Communication Systems*, 2008.

[11] M. L. D. Wong and A. K. Nandi, "Automatic digital modulation recognition using artificial neural network and genetic algorithm", Signal Processing, 84, 2004, pp. 351-36.

[12] A. Swami and B. M. Sadler, "Hierarchical digital modulation classification using cumulants", *IEEE Trans. Commun*., vol. 48, 2000, pp. 416-429.

[13] V. Chaithanya, V. U. Reddy, "Blind modulation classification in the presence of carrier frequency offset", *International Conference on Signal Processing and Communications (SPCOM), 2010* , 18-21 July 2010, pp.1-5.

[14] L. Hong , "Low-complexity identifier for M-ary QAM signals", *SOUTHEASTCON '09. IEEE*, pp. 164-168, 2009.

[15] M. R. Mirarab, M. A. Sobhani, "Robust modulation classification for PSK/QAM/ASK using higher order cumulants", in *Proc. of the Sixth International Conference on Information, Communications and Signal Processing*, 2007.

[16] O. A. Dobre, Y. Bar-Ness, and W. Su, "Higher-order cyclic cumulants for high order modulation classification," in *Proc. IEEE MILCOM*, , 2003 pp. 112-117.

[17] Z. Zhu, M. W. Aslam, and A. K. Nandi, "Augmented genetic programming for automatic digital modulation classification", *Proceedings of the IEEE International workshop on Machine Learning for Signal Processing (MLSP-2010)*, Kittila, Finland, 29 August–01 September 2010, pp. 391-396.

[18] M. W. Aslam, Z. Zhu, and A .K. Nandi, "Automatic digital modulation classification using genetic programming with K-nearest neighbour", *Proceedings of the Military Communications Conference (MILCOM-2010), San Jose, CA, USA*, 31 October–03 November 2010, pp. 512-517.

[19] V. D. Orlic, M. L. Dukic, "Multipath channel estimation algorithm for automatic modulation classification using sixth-order cumulants", *Electronics Letters* , vol.46, no.19, Sept. 16 2010, pp.1348-1349.

[20] Q. Lanjun, Z. Canyan, "Modulation classification based on cyclic spectral features and neural network", *3rd International Congress on Image and Signal Processing (CISP), 2010*, vol.8, 16-18 Oct. 2010, pp.3601-3605.

[21] J. R. Koza. Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, 1992.

[22] P. G. Espejo, S. Ventura, F. Herrera, "A Survey on the Application of Genetic Programming to Classification", *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews,* vol. 40, 2010, pp. 121-144.

[23] J. Eggermont, A. E. Eiben and J. I. van Hemert, "A comparison of genetic programming variants for data classification", *In Proc. on the Third Symposium on Intelligent Data Analysis (IDA*), 1999.

[24] J. K. Kishore, L. M. Patnaik, V. Mani, and V. K. Agrawal, "Application of genetic programming for multi-category pattern classification", *IEEE Transactions on Evolutionary Computation*, vol. 4, 2000, pp. 242-258.

[25] L. Zhang, L. B. Jack and A. K. Nandi, "Extending genetic programming for multi-class classification by combining k-nearest neighbour", *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2005, pp. 349-352.