Creating classification rules using grammatical evolution

Ioannis G. Tsoulos

Department of Computer Engineering,
School of Applied Techhnology,
Technological Educational Institute of Epirus,
47100 Arta, Greece
Email: itsoulos@teiep.gr

Abstract: A genetic programming based method is introduced for data classification. The fundamental element of the method is the well-known technique of Grammatical Evolution. The method constructs classification programs in a C – like programming language in order to classify the input data, producing simple if-else rules. The paper introduces the method as well as the conducted experiments on a series of datasets against other well known classification methods.

Keywords: genetic algorithm; data classification; grammatical evolution; stochastic methods.

Reference to this paper should be made as follows: Tsoulos, I.G. (2020) 'Creating classification rules using grammatical evolution', *Int. J. Computational Intelligence Studies*, Vol. 9, Nos. 1/2, pp.161–171.

Biographical notes: Ioannis G. Tsoulos received his BS degree in Computer Science from the University of Ioannina, Greece in 1998 and PhD on Global Optimization in 2006. He has worked on several research programs. Also, he has served several years as a visiting Lecturer in Universities and Technological Educational Institutes. He has published almost 60 peer-reviewed manuscripts. His research interests include global optimisation, neural networks, genetic algorithms, genetic programming, biomedical signal and image processing. He holds the position of Assistant Professor at the Department of Computer Engineering of Technological Educational Institute (T.E.I.) of Epirus.

1 Introduction

The paper proposed a new method for data classification that is based on Grammatical evolution and it can create classification programs in a C – like programming language.

Data classification is a very challenging problem that finds many applications on a series of practical problems from areas such as data produced from water experiments (Guler et al., 2002), drug classification (Byvatov et al., 2003), water quality problems (Singh et al., 2009), data for cancer classification (Guyon et al., 2002), classification of images from macromolecules (Marabini and Carazo, 1994), classification for data

that represent economic series (Kaastra and Boyd, 1996), prediction analysis for bank data (Leshno et al., 1996), classification of astronomic data (Folkes et al., 1996), classification of biological datasets (Cai et al., 2003) etc. During the past years many methods have been proposed to problems of this category such as neural networks (Hornik, 1989), deep neural networks for image classification (Krizhevsky et al., 2012), convolution neural networks (Kim, 2012), probabilistic neural networks (Spech, 1990), fuzzy neural networks (Simpson, 1992), self organising neural networks (Ultsch, 1993), radial basis functions networks (Buhmann, 2003), tree based rbf neural networks (Gillespie and Nguyen, 2005), support vector machines (Steinwart and Christmann, 2008; Zhu and Blumberg, 2002), Naive Bayes and J48 algorithms (Patil and Sherekar, 2013) etc. Grammatical evolution is an evolutionary process that has been applied with success in many areas such as music composition (Ortega et al., 2002), economics (O'Neill et al., 2001), symbolic regression (O'Neill and Ryan, 2003), robot control (Collins and Ryan, 2000), caching algorithms (O'Neill and Ryan, 1999), water quality problems (Chen et al., 2008), construction of neural networks (Tsoulos et al., 2008), game programming (Perez et al., 2011) etc. The proposed method constructs classification programs in human readable form and does not require additional information from the objective problem (such as derivatives). The method requires small amount of memory (only the objective problem) and it can be applied to any dataset without restrictions to the dimensionality of the problem.

The rest of this article is organised as follows: in Section 2 the basic steps of the method are outlined. In Section 3 a series of experiments on some well-known classification datasets are demonstrated. In Section 4 conclusions are presented.

2 Method description

The procedure of grammatical evolution has two major requirements:

- The context free grammar (CFG) of the target language as expressed in Backus Naur form (BNF).
- The associated fitness function.

The CFG grammar G is defined as $G = (N, T, S, P)$, where N is a set of non terminal symbols, T is a finite set of terminal symbols with the constraint $N \cap T = \emptyset$. The terminal symbol S is named start symbol of the grammar and P is a finite set of production rules in the form $A \rightarrow a$ or $A \rightarrow aB,\ A, B \in N,\ a \in T$.

Grammatical evolution is a genetic programming procedure, but the chromosomes are not expressed in the usual for of parse trees but as vectors of integers. Each element of the vector stands for a production rule from the given BNF grammar. The procedure initiates from the start symbol of the grammar and iteratively produces the program string, by replacing non terminal symbols with the right hand of the selected production rule. The selection is performed in two steps:

- The next element from the vector is taken (denoted as V).
- The production rule is selected using the scheme rule = V mod R, where R is the number of production rules for the current non-terminal symbol.

The selection is executed iteratively until either a valid expression is produced or the end of chromosome is reached. For the second case the chromosome is considered as invalid and we can start over (wrapping event) from the beginning of the chromosome. If the maximum number of wrapping events has been reached, then the chromosome is considered as invalid. The BF grammar used to create a classification program for a problem with two classes (0 and 1) is shown in Figure 1. The parameter D determines the dimensionality of the objective problem. The numbers in parentheses denote the sequence number of the corresponding production rule to be used in the chromosome production procedure. In the proposed technique the chromosomes are represented as sets of integers rather than in binary form. This representation was chosen to increase speed of the evolution. Each gene g_i is defined as: $g_i \in [0..255]$. The upper limit 255 means that the maximum number of production rules in the grammar is 255 but this limit can easily change. For better understanding of the production procedure consider the chromosome $C = [10, 65, 12, 31, 28, 9, 8, 6, 10, 6, 2, 0, 1]$. In Table 1 we list the steps of producing a valid classification program using the grammar of two classes in Figure 1. The dimensionality of the input problem is considered as D = 2.

Figure 1 The grammar used by GenClass for a problem with two classes (0 and 1)

```
<S>::=if(<BEXPR>) CLASS=0 else CLASS=1 (0)
<BEXPR>::=<XLIST><BOOLOP><EXPR>            (0)
         |!(<BEXPR>)                       (1)
         |<XLIST><BOOLOP><EXPR>&<BEXPR>    (2)
         |<XLIST><BOOLOP><EXPR>|<BEXPR>    (3)
<BOOLOP>::= >                              (0)
          | >=                             (1)
          | <                              (2)
          | <=                             (3)

<EXPR>::= (<EXPR><BINARYOP><EXPR>)         (0)
        | <FUNCTION>(<EXPR>)               (1)
        | <TERMINAL>                       (2)
<BINARYOP>::= +                            (0)
            | -                            (1)
            | *                            (2)
            | /                            (3)
<FUNCTION>::= sin | cos | exp | log         (0-3)
<TERMINAL>::= <XLIST>                      (0)
            | <DIGITLIST>.<DIGITLIST>      (1)
            | (-<DIGITLIST>.<DIGITLIST>)   (2)
 <XLIST>::= x1 | x2 | ...|xD               (0-D-1)
<DIGITLIST>::=<DIGIT>                      (0)
             | <DIGIT><DIGIT>              (1)
             | <DIGIT><DIGIT><DIGIT>       (2)
<DIGIT>::= 0 | 1 | 2 | 3 | 4 |5 |6 |7 |8 |9    (0-9)
```

Notes: The parameter D in the determination of non-terminal symbol XLIST specifies the dimensionality of the objective problem. The numbers in parentheses denote the sequence number of the corresponding production rule to be used in the chromosome production procedure.

Table 1 Steps of producing an example valid classification program

String	*Chromosome*	*Operation*
if(<BEXPR>) CLASS=0 else CLASS=1	10, 65, 12, 31, 28, 9, 8, 6, 10, 6, 2, 0, 1	
if(<XLIST><BOOLOP><EXPR>) CLASS=0 else CLASS=1	65, 12, 31, 28, 9, 8, 6, 10, 6, 2, 0, 1	10%4 = 0
if(x1<BOOLOP><EXPR>) CLASS=0 else CLASS=1	12, 31, 28, 9, 8, 6, 10, 6, 2, 0, 1	65%2 = 1
if(x1><EXPR>) CLASS=0 else CLASS=1	31, 28, 9, 8, 6, 10, 6, 2, 0, 1	12%4 = 0
if(x1><EXPR><BINARYOP> <EXPR>) CLASS=0 else CLASS=1	28, 9, 8, 6, 10, 6, 2, 0, 1	31%3 = 1
if(x1><FUNCTION>(<EXPR) <BINARYOP><EXPR>) CLASS=0 else CLASS=1	9, 8, 6, 10, 6, 2, 0, 1	28%3 = 1
if(x1>cos(<EXPR>)<BINARYOP> <EXPR>) CLASS=0 else CLASS=1	8, 6, 10, 6, 2, 0, 1	9%4 = 1
if(x1>cos(<TERMINAL><BINARYOP> <EXPR) CLASS=0 else CLASS=1	6, 10, 6, 2, 0, 1	8%3 = 2
if(x1>cos(<XLIST>)<BINARYOP> <EXPR) CLASS=0 else CLASS=1	10, 6, 2, 0, 1	6%3 = 0
if(x1>cos(x0)<BINARYOP> <EXPR>) CLASS=0 else CLASS=1	6, 2, 0, 1	10%2 = 0
if(x1>cos(x0)*<EXPR>) CLASS=0 else CLASS=1	2, 0, 1	6%4 = 2
if(x1>cos(x0)*<TERMINAL>) CLASS=0 else CLASS=1	0, 1	2%3 = 2
if(x1>cos(x0)*<XLIST>) CLASS=0 else CLASS=1	1	0%3 = 0
if(x1>cos(x0)*x1) CLASS=0 else CLASS=1		1%2 = 1

The main steps of the algorithm are:

1. *Initialisation* step.
   a Read the train data.
   b Set N_G as the maximum number of generations.
   c Set N_C as the number of chromosomes in the population.
   d Set P_S as the selection rate.
   e Set P_M as the mutation rate.
   f Initialise the chromosomes of the population. The chromosomes are initialised randomly as vectors of integers.

2 *Genetic* step

a **For** $i = 1, \ldots, N_g$ **do**

- *Create* for every chromosome in the population a classification program using the previous procedure of grammatical evolution. Denote this classification program as C_i.
- *Calculate* the fitness f_i for every chromosome of the population. The fitness is calculated according to the following equation

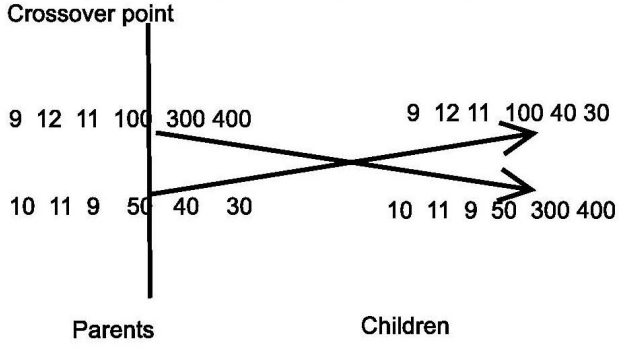
$$f_i = \sum_{i=1}^{M} \left(C_i\left(x_i\right) - t_i\right)^2 \tag{1}$$

where M denotes the number of input patterns in train data, x_i is the i train pattern and t_i is the desired output.

- *Apply* the selection procedure: The chromosomes are sorted in descending order according to their fitness value. The first $(1 - P_s) \times N_c$ chromosomes are copied intact to the next generations. The rest of the chromosomes are produced using the crossover procedure. For every new chromosome two chromosomes (parents) are selected from the old population using tournament selection. The procedure of tournament selection has as follows: a set of $N > 1$ randomly selected chromosomes is produced and the chromosome with the best fitness value in this set is selected and the others are discarded. Each new individual is produced from the two selected parent using the so-called one point crossover. During one point crossover the parent chromosomes are cut at a randomly selected point and their right-hand side subchromosomes are exchanged as show in Figure 2.
- *Apply* the mutation procedure: For every element in each chromosome a random number r in range $[0, 1]$ is produced. If $r \leq P_M$ then the corresponding element is randomly changed.

b **EndFor**

Figure 2 An example of the one point crossover procedure



3 *Evaluation* step

   a Create a classification program for the best chromosome in the population.

   b Apply the previous program to test set and report the induced error.

3 Experimental results

In order to measure the efficiency of the proposed method a series of experiments were conducted on some common classification problems as well as two real world problems (EEG and regions). For all the experiments we have used 10-fold and they were conducted 30 times using different seed for the random generator each time and averages were taken. In all experiments we have used the parameters shown in Table 2. The following datasets were used. The software used in the experiments is freely available from the relevant URL https://github.com/itsoulos/GenClass.

1 Wine dataset: The wine recognition dataset contains data from wine chemical analysis. It contains 178 examples of 13 features each that are classified into three classes.

2 Glass dataset: The dataset contains glass component analysis for glass pieces that belong to six classes. The dataset contains 214 examples with ten attributes each.

3 Pima dataset: The Pima Indians diabetes dataset contains 768 examples of eight attributes each that are classified into two categories: healthy and diabetic.

4 Ionosphere dataset: The ionosphere dataset (ION in the following tables) contains data from the Johns Hopkins Ionosphere database. The two-class dataset contains 351 examples of 34 features each.

5 Eeg dataset: As an real word example, consider an EEG dataset described in Andrzejak et al. (2001) is used here. The dataset consists of five sets (denoted as Z, O, N, F and S) each containing 100 single-channel EEG segments each having 23.6 sec duration. Sets Z and O have been taken from surface EEG recordings of five healthy volunteers with eye open and closed, respectively. Signals in two sets have been measured in seizure-free intervals from five patients in the epileptogenic zone (F) and from the hippocampal formation of the opposite hemisphere of the brain (N). Set S contains seizure activity, selected from all recording sites exhibiting ictal activity. Sets Z and O have been recorded extracranially, whereas sets N, F and S have been recorded intracranially.

6 Spiral artificial data: The spiral artificial dataset (SPIRAL) contains 1,000 two-dimensional examples that belong to two classes (500 examples each). The number of the features is 2. The data in the first class are created using the following formula: $x_1 = 0.5t\cos(0.08t)$, $x_2 = 0.5t\cos\left(0.08t + \frac{\pi}{2}\right)$ and the second class data using**:** $x_1 = 0.5t\cos(0.08t + \pi)$, $x_2 = 0.5t\cos\left(0.08t + \frac{3\pi}{2}\right)$

7 Wisconsin diagnostic breast cancer: The Wisconsin diagnostic breast cancer dataset (WDBC) contains data for breast tumors. It contains 569 training examples of 30 features each that are classified into two categories.

8 Fertility dataset (FERT): 100 volunteers provide a semen sample analysed according to the WHO 2010 criteria. Sperm concentration are related to socio-demographic data, environmental factors, health status, and life habits. It contains 100 examples of 10 features each.

9 Regions dataset: Regions dataset is created from liver biopsy images of patients with hepatitis C (Giannakeas et al., 2015). From each region in the acquired images 18 shape-based and colour-based features were extracted, while it was also annotated form medical experts. The resulting dataset includes 600 samples belonging into six classes.

10 Thyroid dataset: Thyroid disease records (Quinlan et al., 1986) with 7,200 patterns of 21 features each.

11 Parkinsons dataset: This dataset (Little et al., 2009) is composed of a range of biomedical voice measurements from 31 people, 23 with Parkinson's disease.

12 Abalone dataset: A dataset to predict the age of abalone from physical measurements (Nash et al., 1994).

13 Satellite image dataset (Satimage): The database consists of the multi-spectral values of pixels in 3×3 neighbourhoods in a satellite image, and the classification associated with the central pixel in each neighbourhood.The dataset contains 6,635 patterns.

14 Dermatology dataset: The aim is used for the Eryhemato-Squamous disease. The dataset contains 366 patterns with 33 features each.

Table 2 Parameters for the experiments

Parameter	*Value*
N_C	200
N_G	500
P_S	90%
P_M	5%

The results from the experiments are displayed in Table 3. The column DATASET denotes the name of the dataset. The column NEURAL stands for the average test error from the application of neural network to the corresponding dataset. The number of weights (hidden nodes) for the neural network was set to 10 and a BFGS variant due to Powell (1989) was used to train the network. The column RBF denotes the average test error from the application of a Radial Basis Function network to the dataset. In RBF network the hidden-layer weights are estimated using the K-means algorithm and the pseudo-inverse method is used to derive the output-layer weights The number of hidden nodes for this network was also set to 10. Finally, the column GENCLASS denotes the average test error from the application of the proposed method to the dataset. As it can be deduced from the results, the proposed can improve classification accuracy in the majority of the used datasets.

Table 3 Experimental results

Dataset	*Neural*	*RBF*	*GenClass*
Wine	49.55%	31.41%	8.72%
Glass	53.68%	50.16%	37.72%
Pima	27.52%	25.78%	27.53%
Ionosphere	17.08%	16.06%	9.09%
EEG	36.49%	66.56%	28.60%
Spiral	43.23%	44.87%	41.03%
Wdbc	21.03%	7.27%	6.64%
Fert	19.10%	12.00%	14.10%
Regions	34.72%	25.78%	21.02%
Thyroid	2.50%	10.82%	2.74%
Abalone	60.47%	58.64%	48.54%
Parkinsons	17.79%	17.16%	13.48%
Satimage	59.71%	40.36%	22.85%
Dermatology	65.60%	60.23%	42.20%

4 Conclusions

The article presents a novel method designed for data classification. The new method is based on the well-known grammatical evolution technique and it does not require additional information from the objective problems such as high class derivatives that costs in evaluation time and memory. It can create after a series of generations classification problems in a C – like language which can be used in real C programs without many modifications as an example consider the classification program outlined in Algorithm 1. As it can be seen the method constructs programs with the command if–else of the C repertoire and the features are considered as variables for the program.

Algorithm 1 An example of classification problem produced by the new method for the wine dataset

```
1  if(x7>=x3&&x10>(((-463.9)-(log(sin(x1))-sin(x10/x4*x4)))*x6))
2    CLASS=0.00
3  else  if(x12>=x2)
4    CLASS=1.00
5  else
6    CLASS=2.00
```

The proposed method has been with success to a series of classification datasets from the relevant literature. The produced results are compared against the results produced by a neural network and an RBF neural network and the are very promising.

Future research may include more advanced stopping rules such as those demonstrated in Lagaris and Tsoulos (2008) and Kaelo and Ali (2007). Also the method can be easily parallelised using the Open Mpi framework (Graham et al., 2006) for example as it is based on genetic algorithms.

5 Compliance with ethical standards

- Funding: This study was not funded by any grant.
- Conflict of interest: Author Ioannis G. Tsoulos declares that he has no conflict of interest.
- Ethical approval: This article does not contain any studies with human or animal participants performed by any of the authors.

References

Andrzejak, R.G., Lehnertz, K., Mormann, F., Rieke, C., David, P. and Elger, C.E. (2001) 'Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: dependence on recording region and brain state', *Phys. Rev. E*, Vol. 64, No. 6, pp.1–8.

Buhmann, M. (2003) *Radial Basis Functions: Theory and Implementations*, Cambridge Monographs on Applied and Computational Mathematics, pp.1–6, Cambridge University Press, Cambridge.

Byvatov, E., Fechner, U., Sadowski, J. and Schneider, G. (2003) 'Comparison of support vector machine and artificial neural network systems for drug/nondrug classification', *J. Chem. Inf. Comput. Sci.*, Vol. 43, No. 6, pp.1882–1889.

Cai, C.Z., Wang, W.L. and Chen, Y.Z. (2003) 'Support vector machine classification of physical and biological datasets', *Int. J. Mod. Phys. C*, Vol. 14, No. 1, p.575.

Chen, L., Tan, C-H., Kao, S-J. and Wang, T-S. (2008) 'Improvement of remote monitoring on water quality in a subtropical reservoir by incorporating grammatical evolution with parallel genetic algorithms into satellite imagery', *Water Research*, Vol. 42, No. 1, pp.296–306.

Collins, J.J. and Ryan, C. (2000) *Proceedings of AROB 2000, Fifth International Symposium on Artificial Life and Robotics*.

Folkes, S.R., Lahav, O. and Maddox, S.J. (1996) 'An artificial neural network approach to the classification of galaxy spectra', *Montly Notices of the Royal Astronomical Society*, Vol. 283, No. 2, pp.651–665.

Giannakeas, N., Tsipouras, M.G., Tzallas, A.T., Kyriakidi, K., Tsianou, Z.E., Manousou, P., Hall, A., Karvounis, E.C., Tsianos, V. and Tsianos, E. (2015) 'A clustering based method for collagen proportional area extraction in liver biopsy images', *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, November, No. 7319047, pp.3097–3100.

Gillespie, W.J. and Nguyen, D.T. (2005) 'Video classification using a tree-based RBF network', *IEEE International Conference on Image Processing ICIP 2005*.

Graham, R.L., Woodall, T.S. and Squyres, J.M. (2006) 'Open MPI: a flexible high performance MPI', in Wyrzykowski, R., Dongarra J., Meyer, N. and Waśniewski, J. (Eds.): *Parallel Processing and Applied Mathematics. PPAM 2005. Lecture Notes in Computer Science*, Vol. 3911, Springer, Berlin, Heidelberg.

Guler, C., Thyne, G.D., McCray, J.E. and Turner, K.A. (2002) 'Evaluation of graphical and multivariate statistical methods for classification of water chemistry data', *Hydrogeology Journal*, Vol. 10, No. 4, pp.455–474.

Guyon, I., Weston, J., Barnhill, S., Vapnik, V. and Gene (2002) 'Selection for cancer classification using support vector machines', *Machine Learning*, Vol. 46, pp.389–422.

Hornik, K. (1989) 'Multilayer feedforward networks are universal approximators', *Neural Networks*, Vol. 2, No. 5, pp.359–366.

Kaastra, I. and Boyd, M. (1996) 'Designing a neural network for forecasting financial and economic time series', *Neurocomputing* Vol. 10, No. 3, pp.215–236.

Kaelo, P. and Ali, M.M. (2007) 'Integrated crossover rules in real coded genetic algorithms', *European Journal of Operational Research*, Vol. 176, No. 1, pp.60–76.

Kim, Y. (2014) 'Convolutional neural networks for sentence classification', *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*.

Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012) 'ImageNet classification with deep convolutional neural networks', *Advances in Neural Information Processing Systems*, Vol. 1, pp.1097–1105.

Lagaris, I.E. and Tsoulos, I.G. (2008) 'Stopping rules for box-constrained stochastic global optimization', *Applied Mathematics and Computation*, Vol. 197, No. 2, pp.622–632.

Leshno, M. and Spector, Y. (1996) 'Neural network prediction analysis: the bankruptcy case', *Neurocomputing*, Vol. 10, No. 2, pp.125–147.

Little, M.A., McSharry, P.E., Hunter, E.J. and Ramig, L.O. (2009) 'Suitability of dysphonia measurements for telemonitoring of Parkinson's disease', *IEEE Transactions on Biomedical Engineering*, Vol. 56, No. 4, pp.1015–1022.

Marabini, R. and Carazo, J.M. (1994) 'Pattern recognition and classification of images of biological macromolecules using artificial neural networks', *Biophysical Journal*, Vol. 66, No. 6, pp.1804–1814.

Nash, W.J., Sellers, T.L., Talbot, S.R., Cawthorn, A.J. and Ford, W.B. (1994) *The Population Biology of Abalone (Haliotis species) in Tasmania. I. Blacklip Abalone (H. Rubra) from the North Coast and Islands of Bass Strait, Sea Fisheries Division*, Technical Report No. 48, ISSN: 1034-3288.

O'Neill, M. and Ryan, C. (1999) in Miettinen, K., Mkel, M.M., Neittaanmki, P. and Periaux, J. (Eds.): *Evolutionary Algorithms in Engineering and Computer Science*, pp.127–134, Jyvskyl, Finland.

O'Neill, M. and Ryan, C. (2001) 'Grammatical evolution', *IEEE Trans. Evol. Comput.*, Vol. 5, No. 4, pp.349–358.

O'Neill, M and Ryan, C. (2003) 'Grammatical evolution: evolutionary automatic programming in a arbitary language', *Genetic Programming*, Kluwer Academic Publishers, Dordrecht.

O'Neill, M., Brabazon, A., Ryan, C. and Collins, J.J. (2001) 'Evolving market index trading rules using grammatical evolution', *Applications of Evolutionary Computing Volume 2037 of the Series Lecture Notes in Computer Science*, pp.343–352.

Ortega, A., Sanchez, R. and Moreno, M.A. (2002) 'Automatic composition of music by means of grammatical evolution', *APL'02 Proceedings of the 2002 Conference on APL: Array Processing Languages: Lore, Problems, and Applications*, pp.148–155.

Patil, T.R. and Sherekar, S.S. (2013) 'Performance analysis of Naive Bayes and J48 classification algorithm for data classification', *International Journal of Computer Science and Applications*, Vol. 6, pp.256–261.

Perez, D., Nicolau, M., O'Neill, M. and Brabazon, A. (2011) 'Evolving behaviour trees for the Mario AI competition using grammatical evolution', in Di Chio, C. et al. (Eds.): *Applications of Evolutionary Computation. EvoApplications 2011. Lecture Notes in Computer Science*, Vol. 6624, Springer, Berlin, Heidelberg.

Powell, M.J.D. (1989) 'A tolerant algorithm for linearly constrained optimization calculations', *Mathematical Programming*, Vol. 45, No. 1, p.547.

Quinlan, J.R., Compton, P.J., Horn, K.A. and Lazurus, L. (1986) 'Inductive knowledge acquisition: a case study', in *Proceedings of the Second Australian Conference on Applications of Expert Systems*, Sydney, Australia.

Simpson, P.K. (1992) 'Fuzzy min-max neural networks', in *IEEE Transactions on Neural Networks Classification*, Vol. 3, No. 5, pp.776–786.

Singh, K.P., Basant, A., Malik, A. and Jain, G. (2009) Artificial neural network modeling of the river water quality – a case study', *Ecological Modelling*, Vol. 220, No. 6, pp.888–895.

Spech, D.F. (1990) 'Probabilistic neural networks', *Neural Networks*, Vol. 3, No. 1, pp.109–118.

Steinwart, I. and Christmann, A. (2008) 'Support vector machines', *Information Science and Statistics*, Springer-Verlag New York.

Tsoulos, I., Gavrilis, D. and Glavas, E. (2008) 'Neural network construction and training using grammatical evolution', Vol. 72, No. 1, pp.269–277.

Ultsch, A. (1993) 'Self-organizing neural networks for visualisation and classification', in Opitz, O., Lausen, B. and Klar, R. (Eds.): *Information and Classification. Studies in Classification, Data Analysis and Knowledge Organization*, Springer, Berlin, Heidelberg.

Zhu, G. and Blumberg, D.G. (2002) 'Classification using ASTER data and SVM algorithms: the case study of Beer Sheva, Israel', *Remote Sensing of Environment*, Vol. 80, No. 5, pp.233–240.