Uncertainty and Climate Change and its effect on Generalization and Prediction abilities by creating Diverse Classifiers and Feature Section Methods using Information Fusion

Amit Ganatra & Y. P. Kosta
ypkostaresearch@yahoo.com ypkostaresearch@yahoo.com

U & P U. Patel Department of Computer Engineering,
Charotar University of Science & Technology-CHARUSAT
Gujarat, India

Abstract

The model forecast suggests a deterministic approach. Forecasting was traditionally done by a single model - deterministic prediction, recent years has witnessed drastic changes. Today, with Information Fusion (Ensemble) technique it is possible to improve the generalization ability of classifiers with high levels of reliability. Through Information Fusion it is easily possible to combine diverse & independent outcomes for decision-making. This approach adopts the idea of combining the results of multiple methods (two-way interactions between them) using appropriate model on the testset. Although uncertainties are often very significant, for the purpose of single prediction, especially at the initial stage, one dose not consider uncertainties in the model, the initial conditions, or the very nature of the climate (environment or atmosphere) itself using single model. If we make small changes in the initial parameter setting, it will result in change in predictive accuracy of the model. Similarly, uncertainty in model physics can result in large forecast differences and errors. So, instead of running one prediction, run a collection/package/bundle (ensemble) of predictions, each one kick starting from a different initial state or with different conditions and sequentially executing the next. The variations resulting due to execution of different prediction package/model could be then used (independently combining or aggregating) to estimate the uncertainty of the prediction, giving us better accuracy and reliability. In this paper the authors propose to use Information fusion technique that will provide insight of probable key parameters that is necessary to purposefully evaluate the successes of new generation of products and services, improving forecasting. Ensembles can be creatively applied to provide insight against the new generation products yielding higher probabilities of success. Ensemble will yield critical features of the products and also provide insight to forecasting ultimately improving the predicative skills & capabilities. This is accomplished by creative selection of multiple predicators and combining the same to crack down the complexity. Diversity can be achieved from different algorithms, or algorithm parameters.

Keywords

Information Fusion, Boosting, Collection, Ensemble, Predictions

I. INTRODUCTION

In a Real-world, data exist in complexity, knowledge discovery processes typically involves pre-processing, learning, evolving, evaluation, and visualization. Therefore, typically, data mining platform must be capable of; addressing and handling complex model chains, provide transparency, flexibility and scalability in data handling, optimization of critical parameters and

DOI : 10.5121/ijdms.2010.2407

an easy-to-use environment. Depending on the type and nature of task, a user can interactively explore different domain-chain relating to knowledge discovery, inspect intermediate results, or even execute a highly automated data mining processes in batch mode. Thus an ideal data mining platform offers the above mentioned functions as a single tool-set, with cross validation capabilities [1].

While we discuss the rationale behind multiple classifiers, limitations of individual classifiers cannot be ignored. The rationale behind the growing interest in multiple classifier systems (MCSs) is that the classical approach to designing a system, which focuses on the search for the best individual classifier, has some serious drawbacks. The primary being that the best individual classifier for the classification task at hand is very difficult to identify, and requires deep prior knowledge, in addition, a single classifier find it extremely difficult to exploit the complementary discriminatory information that other classifiers may be encapsulating [4]. A set of independent classifiers are handpicked and creatively bundled to produce or construct a high performance MCS system.

Multiple classifier systems is built & based on the combination of outputs from different but independent classifiers that consolidate/combine to form/make a set producing a high performance out, in the process making a high performance classification system. Generally most independent classifiers produce errors that depend upon the type of classifier. The challenge here is to reduce the error dependency of the individual classifier in an MCS. Our approach is to design MCS that is independent of errors. Based on the past work, we can say that the efficiency of multiple classifiers depends on how they explore independent errors. Previously single classifier with different parameter values was used and then through experiments parameters were optimized to meet the performance. The performance of single classifier system was limited. If we apply the same concept to the diverse fields then it may fail to meet the performance criteria even if all parameters and architectures have been fully optimized. In such situations, using multiple classifiers, one of the most significant advances in classification in recent years, proves to be better in terms of performance enhancement and improving generalization and prediction accuracies. [4].

Specifically, we want to know whether generating accurate and diverse classifiers for ensembles, which prove to work well for accuracy improvement, guarantees that an ensemble performs better. More importantly, we need innovations on how effectively we design to design an ensemble and fusion of them for diverse classification problems. This paper is an attempt towards addressing some of these issues [7].

II. COMPOSITE CLASSIFIER DESIGN CRITERIA

There are three primary criteria that have been applied in creating multiple classifiers:

1. **Accuracy** of the component classifiers,
2. **Diversity** of the component classifiers, and
3. **Efficiency** of the entire composite classifier.

The **accuracy** criterion arises from a desire to make the component classifiers independently accurate. The accuracy of the component classifiers has been regarded as the most important of the three criteria, and a great deal of effort typically goes to training the individual components to be highly accurate as independent classifiers.

If we had access to a classifier with perfect generalization performance, there would be no need to resort to ensemble techniques. The realities of noise, outliers and overlapping data distributions, however, make such a classifier an impossible proposition. At best, we can expect the classification accuracies to be higher. The methodology in multiple classifiers is to create and combine many classifiers and their outputs such that the overall combination improves the performance. This requires, however, that individual classifiers make errors on different instances. The intuition is that if each classifier makes different errors, then a strategic combination of these classifiers can reduce the total error. The overarching principal in ensemble systems is therefore to make each classifier as unique as possible, particularly with respect to misclassified instances. Specifically, we need classifiers with decision parameters which are adequately different from others i.e. diverse in some sense.

The diversity is important because if all the classifiers make the same errors cannot lead to any improvement in the accuracy and predictive accuracy of the combine prediction. There are many methods to generate diverse individual classifiers for ensembles.

The **efficiency** criterion is less often considered in composite classifier work, but arises from the general requirement that a classifier should use only reasonable amounts of time and memory for training and application.

Avoiding prohibitively expensive classifiers has been cited previously as a design goal.

One ought to prefer

(a) Fewer component classifiers over more, and

(b) Computationally inexpensive ones over more expensive [4] [5] [7].

III. ISSUES REGARDING CLASSIFICATION AND PREDICTION

1. Pre-processing of the data in order to have quality decisions from quality data.
2. Attribute relevance analysis-Remove the irrelevant or redundant attributes
3. How to make data suitable to mining? i.e. through Data transformation-Generalize and/or normalize data and
4. Mining on which kinds of attributes-Numerical attribute income Þ categorical {low, medium, high}
5. How to normalize when all numerical attributes are there and how to transform them into the range between [0, 1].

Along with these following are important requirements which should be met.

1. Generalization ability and Predictive accuracy
2. Time Complexity (Speed)
   Time to construct the model
   Time to use the model
3. Robustness
   Handling noise and missing values
4. Scalability- The algorithm should be scalable
   Efficiency in disk-resident databases
5. Human Understanding and Interpretability:
   Understanding and insight provided by the model

6. Quality and Goodness of rules ()
   Compactness of classification rules

IV. FEATURE SELECTION

When we have a dataset with various attributes, we would like to check, whether all of these attributes are really relevant and required, or whether we can get a better model by removing some of the original attributes. This task is called feature selection and the Genetic Algorithm is one of the approach that can solve it [6].

In feature selection process we try to identify and remove as much irrelevant and redundant features as possible. This reduces the dimensionality of the data and enables learning algorithms to operate faster and more effectively. Generally, features are characterized as:

1. Relevant: These are features which are required for classification and have an impact on the output and their role can not be assumed by the rest.
2. Irrelevant: Irrelevant features are those by removing them, have no impact and influence on the output, and whose values are generated at random for each example.
3. Redundant: A redundancy exists whenever a feature can take the role of another (perhaps the simplest way to model redundancy)[8].

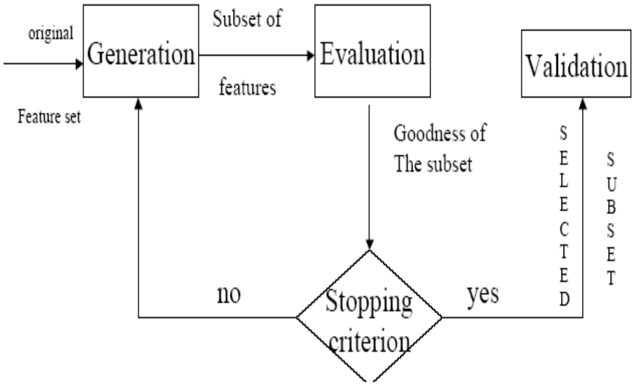

Fig. 1

General Characteristics of Feature Selection Methods:

Feature selection aims to search the relevant features in the feature space From the point of view of heuristic search, Blum and Langley argue that the following four issues, which affect the nature of the search, can characterize any feature selection method.

1. Depending on the starting point, the search direction will vary.

2. The organization of the search procedure. Obviously, if the number of features is too large, the exhaustive search of all the feature subspace is prohibitive, as there are 2N possible combinations for N features. Recommended method is heuristic search rather than exhaustive search.

3. The evaluation strategy. How feature subsets are evaluated is an important problem. As for classification, the ideal feature subset should have the best separation of the data.

4. The criterion for stopping the search. Various criteria to be decided e.g. performance, no. of generations, no change in error or performance. [6] [8].

V. DESIGNING NN ENSEMBLES

More systematic and efficient methods for designing classifier ensembles is still expected. Neural network ensembles, and other classifier ensembles alike, are typically designed heuristically in two steps: first generating individual classifiers, and then combining the outputs of the individual classifiers, for example, by simply averaging. While generating the classifiers, one has to focus on accuracy and diversity of the individual classifier as far as possible. The underlying hypothesis is that as long as the individual classifiers are reasonably accurate and diverse, the resulting ensemble will be more accurate. Although general relationship between ensemble accuracy and the diversity is still not clear, such design approach (generating accurate and diverse individual classifiers) can often time yield an ensemble that is more accurate than individual classifiers. Simply generating diverse individual classifiers may not result in an ensemble that optimally meets the performance requirements. Instead, we argue that the individual classifiers need to be designed by directly targeting to meet the performance requirements in addition to having the normal properties, i.e., reasonable accuracy and high possible diversity. To achieve this, we combine performance requirements into the actual process of feature selection to ensure better performance with individual classifiers [8] [13].

Algorithm:
Step 1

i. Use Feature Selection based on Genetic Algorithm from training data
ii. For random sample with replacement n times
iii. Run GA based feature selection
iv. Save the selected feature set
v. End for loop

Step 2

i. Train the Ensemble (NNs) after feature selection

Step 3

ii. Testing data
iii. Test the Ensemble (NNs) after training and aggregate the result from base learners (NNs)

Step 4

i. Produce the result as accuracy, ROC curve.

Three important diversity measures in designing individual neural networks:

1) Each neural network in the ensemble should be checked with different parameters and architectures i.e. with a different number of hidden neurons in the hidden layer, with a different learning rate and activation functions.
2) Each neural network in the ensemble should be trained with a dataset that is randomly sampled from the original training data.
3) Each neural network in the ensemble should be checked with different subset of features, which is selected using one generation (iteration) of genetic algorithms (GA) based feature selection.

Fitness function is the actual driver of the genetic algorithm and unlike GA-based feature selection for ensembles which was used previously where the fitness function is either accuracy or some combination of accuracy and diversity of the classifiers, we set the fitness function of GA min FPR subject to a TPR during feature selection process to be exactly match up with the performance requirement,.

The design consists of three main steps: 1) GA-based feature selection, 2) network training, and 3) classifier testing. In the feature selection step, we run GA-based feature selection *different number of times* to obtain *different sets* of features, each of which is used for one individual classifier (neural network). Each GA run uses a different set of data along with a different random initial population. The mechanism for obtaining data sets are obtained by random sampling with replacement strategy to obtain the original data set *n* times, where *n* is the number of examples in the training set [6].

In the training step, each neural network with different number of hidden neurons, learning rate and activation functions is trained independently using the entire training data. The number of inputs for each individual neural network is different feature sets that GA selected. The trained networks are then tested using the testing data set. The outputs of the networks for each case are combined by some way like majority voting or simple averaging to arrive at the output of the ensemble. By varying decision threshold and applying against the ensemble outputs, a set of TPR-FPR pairs is obtained, thus the ROC curve can be generated.[2] [6] [8] [9] [11].
The main difference between the proposed suggested method and the past methods is the use of the different fitness functions for GA feature selection. In the conventional design, the fitness function of GA is the training set accuracy and here we suggest some diversity measures for GA: TPR, the kappa statistic, and the fail/non-fail disagreement.

VI. DATASETS

Iris Plants Dataset

This Iris Plants Database was created by Creator: R.A. Fisher. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. It contains 150 (50 in each of three classes) instances in its basic form. There are 4 numeric predictive attributes and the class.

TABLE I
IRIS PLANTS DATASET INFORMATION

	Attribute Name	**Type**	**Description**
1	sepal length	Numerical	It describes length of sepal in Iris Plants
2	sepal width	Numerical	It describes width of sepal in Iris Plants
3	petal length	Numerical	It describes length of petal in Iris Plants
4	petal width	Numerical	It describes length of petal in Iris Plants
5	Class	Catagorical	Class of Iris Plants

TABLE II
IRIS PLANTS DATASET'S VALUE DETAIL

No.	**FIELD NAME**	**Possible Values**					**Distinct Values**
		MIn	Max	Mean	SD	Class Correlation	
1	sepal length	4.3	7.9	5.84	0.83	0.7826	-
2	sepal width	2.0	4.4	3.05	0.43	-0.4194	-
3	petal length	1.0	6.9	3.76	1.76	0.9490	-
4	petal width	0.1	2.5	1.20	0.76	0.9565	-
5	Class	Iris Setosa, Iris Versicolour, Iris Virginica					3

VII. PERFORMANCE ANALYSIS (RESULTS)

The results on the IRIS data classification problem suggest that the results are acceptable and are good; however the scalability of the algorithm needs to be proved with more difficult classification problems.

Details of IRIS Data:

1. Number of Instances in Training set: 90
2. Number of Instances in testing set: 60
3. Number of Attributes: 4 input+1 class attribute
4. For Each Attribute: all input attributes are numeric and class attribute is categorical.
5. The output attribute is the class code 0..2

PERFORMANCE ANALYSIS BASED ON DIFFERENT PARAMETERS

Testing the performance is done on a Personal Computer with 1 GB RAM & 2.8 GHz speed processor. Testing set and Training set are separated as per hold out method.
Performance analysis based on learning rate.
Other parameters are same for below throughout training [5] [7] [10][14-23].

TABLE III
PERFORMANCE ANALYSIS BASED ON LEARNING RATE

No of Iterations:	600
No of Hidden Layers:	1
No of Hidden Nodes:	10
Error Tolerance:	90%
Mutation Rate:	0.2
Cross Over Rate:	0.3
Generations:	25

Population:	300

Learning Rate	Accuracy %	Time In Sec
0.3	98	1.49
0.4	98	1.24
0.5	98	1.44
0.6	78	1.45

Performance analysis based on Iterations

Other parameters are same for below throughout the training.

TABLE IV
PERFORMANCE ANALYSIS BASED ON ITERATION

Learning Rate:	0.4
No of Hidden Layers:	1
No of Hidden Nodes:	10
Error Tolerance:	90%
Mutation Rate:	0.2

Cross Over Rate:	0.3
Generations:	25
Population:	300

Iterations	Accuracy %	Time in sec
100	96	0.17
200	96	0.28
300	97	0.38
500	97	0.59
1000	98	1.08

Performance analysis based on number of Hidden nodes

Other parameters are Learning Rate =0.4, Error Tolerance =80%,
Iterations =300, No of Hidden layers = 1, No of Hidden Nodes = 10,
Mutation Rate = 0.2, Cross Over Rate = 0.3, Generation = 25, Population = 300 .

TABLE V
PERFORMANCE BASED ON NO OF HIDDEN NODES

Hidden Nodes	Accuracy %	Time In sec
8	95	4.25
9	95	5.15
10	97	6.05
11	95	6.50

Performance analysis based on number of Hidden layers

Other parameters are Learning Rate =0.4, Error Tolerance =80%,
Iterations = 200, No of Hidden layers = 1,
Mutation Rate = 0.2, Cross Over Rate = 0.3, Generation = 25, Population = 300.

TABLE VI
PERFORMANCE BASED ON NO OF HIDDEN LAYERS

Hidden Nodes in layer1	Hidden Nodes in layer 2	Accuracy %	Time in sec
6	3	96	13
7	2	96	14
8	1	95	14
6	4	95	15
5	5	97	15

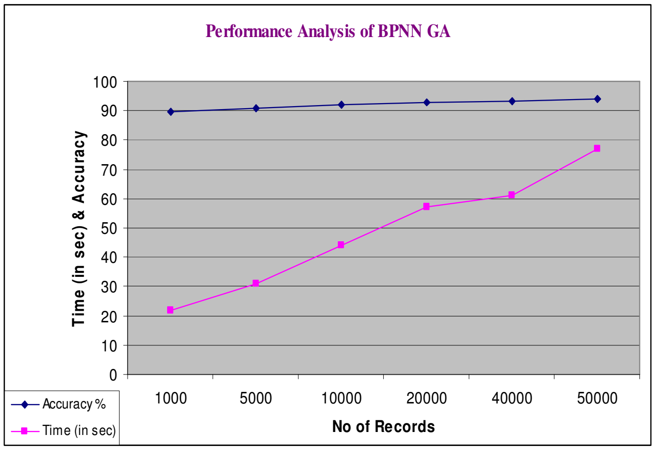

Fig. 2 Performance Analysis of BPNN GA for AdaBoost

VIII. Feature Selection

Comparison of Classification Methods using meta classfier AdaBoost in WEKA:

Relation: weather
Instances: 14
Attributes: 5 Outlook, Temperature, Humidity, Windy, Play, Test mode:
Evaluated: on training data
Output Class: outlook [2].

TABLE VII
COMPARISONS OF NN WITH DECISION TREE INDUCTION (ID3) AS A BASE CLASSIFIER OF ADABOOST

Characteristics	**MLP(NN)**	**Decision Tree**
Approach used	Non-parametric	Parametric
Classification Accuracy with cross validation	71.4286%	71.4286%
Unclassified region	More if training set is not sufficient	Less compared to NN when training set is not sufficient
Training set accuracy	100%	92.8571%
Test site accuracy	85.7143%	92.8571%
Time taken to build model in seconds	0.25 sec	0.02 sec

TABLE VIII
COMPARISONS OF ADABOOST (MULTICLASS CLASSIFIER) WITH DIFFERENT BASE CLASSIFIERS USING WEKA

Classifier\ Parameters	Correctly Classified Instances	Incorrectly classified Instances	Root mean squared error	Time taken to build model in seconds
MLP	100	0	0.0454	0.19
SMO	78.57	21.42	0.4021	2.95
bayes.net	78.57	21.42	0.4557	0.09
NBTree	78.57	21.42	0.3864	0.08
ZeroR	64.28	35.71	0.4795	0
Filtered Classifier	85.71	14.28	0.344	0.05
Bagging	85.71	14.28	0.3383	0.14

IX. CONCLUSION

In our paper, we have considered measure of diversity in the architecture and parameters of the classifier and its relation with accuracy. As diversity increases the generalization abilities and predictive accuracy is expected to be improved. We have also used genetic algorithm as a search strategy for ensemble feature selection. It was shown in experiments that it results in better ensembles having greater accuracy across the domains, especially data sets with large number of features. Accuracy on the training set is increased for AdaBoost with MLP as a base classifier compared to ID3 while the accuracy of the test set is consistent. Diversity can be used to measure the growth and the potential for improvement on the overall combination of the classifiers.

X REFERENCES

[1] Robert G. Reynolds, Bin Peng, and Raja' S.Alomari, Cultural Evolution of Ensemble Learning for Problem Solving, 2006 IEEE Congress on Evolutionary Computation Sheraton Vancouver Wall Centre Hotel, Vancouver,BC, Canada July 16-21, 2006.

[2] H Chouaib, O Ramos Terrades,S. Tabbone,F Cloppet,N Vincent Laboratoire CRIP5(EA 2517),University Paris Descartes,France,LORIA-University Nancy2 ,Campus Scientique BP 239 Nancy,France, Feature Selection Combining Genetic Algorithm and Adaboost Classfiers, 978-1-4244-2175-6/08/ 2008 IEEE.

[3] Weizhong Yan and Kai F Goebel, Designing Classifier Ensembles with Constrained Performance Requirements.

[4] Albert Hung-Ren Ko,Roberts Sabourin, and Alceu de Souza Britto, Jr., Combining Diversity and Classficiation Accuracy for Ensemble Selection in Random Subspaces.

[5] Lesedi Masisi, Fulufhelo V. Nelwamondo and Tshilidzi Marwala,School of Eletrical and Information Engineering, University of the Witwatersrand, Private Bag 3 2050 South Africa, The Effect of Sturctural Diversity of an Ensemble of Classifiers on Classification Accuracy.

[6] Li Shuo,Jing Zheng, Gang Wang, Xia Li,Bin Ai,Junping Qian,School of Geography and Planning Sun Yat-sen University Guangzhou 510275,China.Guangdong Climate Centre, Guangzhou 510080,China,School of Geographical Seiences ,Guangzhuo University,Guangzhuo,China, A GENETIC ALGORITHM BASED WRAPER FEATURE SELECTION METHOS FOR CLASSIFICATION OF HYPERSPECTRAL IMAGES USING SUPPORTS VECTOR MACHINE.

[7] R. Musehne, F. Netshiongolwe, F.V Nelwamondo, L Masisi and T. Marwala, School of Electrical and Information Engineering ,University of the Witwatersrand, Private Bag-3 ,2050 Johannesburg , South Africa, Relationship between Diversity and Performance of multiple Classifiers for Decision Support.

[8] V.Landassuri-Moreno,John A. Bullinaria,School of computer science University of Birmingham ,Birmingham,B15 2TT UK,V.Landassuri_Moreno@cs.bham.ac.uk,J.A.Bullinaria@cs.bham.ac.uk, Feature Selection in Evolved Artificial NeuralNetworks using the evolutionary Algorithm EPNet.

[9] Alexy Tsymbal,Mykola Pechenizkily,Padring Cunningham,of Computer Ascience ,Trinity College Dublin,Dublin 2, Ireland,Dept. of CS &Iss,University of Jyvaskyla P.O. Box 35, Finaland-40351,Dept. of Computer Ascience Trinity College Dublin,Dublin 2, and, tsymbalo@cs.tcd.ie, mpechen@cs.jyu.fi, Padraig.Cunningham@cs.tcd.ie, Sequential Genetic Search for Ensemble Feature Selection.

[10] John Maindonald, Centre for Bioinformation Science, Australian National University, Canberra ACT 0200 Australia, The Role of Models in Predictive Validation.

[11] A Chitra, S N SIVANANRAM, Genetic Based Output Controlled Feed Forward Neural Network

[12] Md. Monirul Islam,Xin,Yao,Fellow,IEEE,S.M.Shahriar Nirjon,Muhammad Asiful Islam and Kazuyuki Murase, Bagging and Boosting Negatively Correlated Neural Network, IEEE TRANSACTIONS ON SYSTEMS,MAN,AND CYBERNETICS-PARTB:CYBERNETICS,VOL.38,NO.3,JUNE 2008.

[13] Albert Hung-Ren Ko,Roberts Sabourin, and Alceu de Souza Britto, Jr., Combining Diversity and Classficiation Accuracy for Ensemble Selection in Random Subspaces.

[14] Goldberg, D.E. "Genetic algorithms in search, optimization, and machine learning", Pearson Education, 2001.

[15] R. K. Gupta, A. K. Bhunia, "An Application of real-coded Genetic Algorithm for integer linear programming", AMO-Advanced Modeling and Optimization, Volume 8, Number 1, 2006.

[16] BL Mak, H Sockel – Info. & Mgmt, A confirmatory factor analysis of IS employee motivation and retention, 2001

[17] K. Hornik,M. Stnchcombe and H White,Multilayer Feedforward Networks are Universal Approximators, Neural Network,2:pg359- 366,1989

[18] Vijyalakshmi Pai G.A. & Rajasekaran S. Neural networks, fuzzy logic and genetic algorithms, Synthesis and applications. Reading, Prentice-Hall of India, 2004

[19] H. Akaike, 'A new look at the statistical model identification', IEEE Transactions on Automatic Control, Vol. 19, No. 6, pp. 716–723, 1974

[20] C. M. Hurvich and C. Tsai, 'Regression and Time Series Model Selection in Small Samples', Biometrika, Vol. 76, pp. 297–307, 1989

[21] J. E. Cavanaugh, 'Unifying the Deriviations for the Akaike and Corrected Akaike Information Criteria', Statistics & Probability Letters, Vol. 33, pp. 201–208, 1997

[22] Mengjie Zhang , "Artificial Neural Networks" Victoria University of Wellington,

[23] V.P. Plagianakos, G.D. Magoulas, M.N. Vrahatis, "Learning rate adaptation in stochastic gradient descent", ,Department of Mathematics, University of Patras,

[24] Wen Jin-Wei Zhao, Jia-Li Luo Si-Wei and Han Zhen " The Improvements of BP Neural Network Learning Algorithm", Department of Computer Science & Technology,Northem Jiaotong University ,BeiJing, 100044, P.R.China,

[25] M. Petronil, A.S. Malowanyl, C.C. Johnston2, B.J.Stevens3, "A Comparison of Neural Network Architectures for the Classification of Three Wpes of Infant Cry Vocalizations".

[26] Dudoit, S. and Fridlyand, J. (2003) Classification in microarray experiments. In Speed, T.P. (Ed.). *Statistical Analysis of Gene Expression Microarray Data*, Chapman & Hall/CRC, Boca Raton, FL, pp. 93–158.

[27] Gentleman, **R**.C., et al. (2004) Bioconductor: open software development **for** computational biology and bioin**for**matics. *Genome Biol*, . **5**, **R**80[CrossRef][Medline].

[28] Goldberg, D.E. *Genetic* ***Algorithms*** *in Search, Optimization, and Machine Learning*, (1989) , **R**eading, MA Addison-Wesley **Vol. xiii**, , pp. 412.

Lee, J.W., et al. (2005) An extensive comparison of **r**ecent classification tools applied to microarray data. *Comput. Stat. Data Anal*, . **48**, 869–885[CrossRef].

[29] Li, L.P., et al. (2001) Gene selection **for** sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. *Bioin**for**matics*, **17**, 1131–1142[Abstract/Free Full Text].

[30] Ooi, C.H. and Tan, P. (2003) Genetic **algorithm**s applied to multi-class prediction **for** the analysis of gene expression data. *Bioin**for**matics*, **19**, 37–44[Abstract/Free Full Text].

[31] Sha, N., et al. (2004) Bayesian variable selection in multinomial probit models to identify molecular signatures of disease stage. *Biometrics*, **60**, 812–819[CrossRef][Web of cience][Medline].

[32] Gary G. Yen and Haiming Lu. Hierarchical genetic algorithm based neural network design. IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks, pages 168{175, 2000.

[33] Chunkai Zhang, Huihe Shao, and Yu Li. Particle swarm optimisation for evolving artificial neural network. IEEE International Conference on Systems, Man and Cybernetics, 2000, 4:2487-2490, 2000.