

Model Selection and Feature Ranking for Financial Distress Classification

Srinivas Mukkamala, *A. S. Vieira, Andrew H. Sung

Department of Computer Science, New Mexico Tech, Socorro, NM 87801

Institute for Complex Additive Systems and Analysis, Socorro, NM 87801

*ISEP and Computational Physics Centre, University of Coimbra, Coimbra, Portugal

Abstract—In this paper we apply several learning machine techniques to the problem of financial distress classification of medium-sized private companies. Financial data was obtained from Diana, a large database containing financial statements of French companies. Classification accuracy is evaluated with Artificial Neural Networks, Classification and Regression Trees (CART), TreeNet, Random Forests and Linear Genetic Programs (LGPs). We analyze both type I (bankrupt companies misclassified as healthy) and type II (healthy companies misclassified as bankrupt) errors on two datasets containing balanced and unbalanced class distribution. LGPs have the best performance accuracy in both balanced data and unbalanced dataset. Our results demonstrate the potential of using learning machines, with respect to discriminant analysis, in solving important economics problems such as bankruptcy detection.

We also address the related issue of ranking the importance of input features, which is itself a problem of great interest. Elimination of the insignificant inputs leads to a simplified problem and possibly faster and more accurate classification of financial distress. Experiments on Diana dataset have been carried out to assess the effectiveness of this criterion. Results show that using significant features gives the most remarkable performance and performs consistently well over financial datasets we used.

Keywords: Classification and Regression Trees, Multivariate Regression Splines Random Forests, TreeNet, Neural Networks, Bankruptcy Detection., feature selection, feature ranking, financial distress classification.

1 INTRODUCTION

Financial distress prediction is of great importance to banks, insurance firms, creditors and investors. The problem is stated as follows: given a set of parameters (mainly of financial nature) describing the situation of a company over a given period, predict the probability that the company may become bankrupted in a near future, normally during the following year.

There has been considerable interest in using financial ratios for predicting financial distress in companies since the seminal work of Beaver [1] using univariate analysis and Altman approach with multiple discriminant analysis [2]. Despite its limitations [3], Multiple Discriminant Analysis (MLD) is still largely used as a standard tool for bankruptcy prediction. Non-linear models, such as the Logit [4] and Probit [5], are used with caution as they only slightly improve the accuracy of MLD and may be sensitive to exceptions, common in this problem.

Bankruptcy prediction is a very hard classification problem as it is high-dimensional, most data distribution is non-Gaussian and exceptions are common [6]. A nonlinear classifier should be superior to a linear approach due to saturation effects and multiplicative factors in the relationships between the financial ratios. For example, an increase of the earnings to total assets ratio from -0.1 to 0.1 is more relevant than an increase from 1.0 to 1.2. One the other the potential for default of a firm with negative cash flow is more amplified if it has large liabilities.

ANNs, implemented by multilayer perceptrons, have been increasingly used to default prediction as they generally outperform other existing methods [7-9]. Recent methods,

such as Support Vector Machines, Genetic Algorithms and Genetic Programming have also been applied in this problem with success. In general all these approaches outperform Multiple Discriminant Analysis. However, in most cases the datasets used are very small (sometimes with less than 100 cases) often highly unbalanced which does not allow a fair comparison [10].

In this work we compare the efficiency of four machine learning approaches on bankruptcy detection using a large database of French private companies. This database is very detailed as it contains a wide set of financial ratios spanning over a period of three years, corresponding to more than one thousand healthy and distressed companies. The approaches used are: Classification and Regression Trees (CART), TreeNet, Random Forest, Linear Genetic Programs (LGPs) and Artificial Neural Networks in two versions: multilayer perceptrons and Hidden Layer Learning Vector Quantization.

This paper is organized as follows: Section 2 presents the Artificial Neural Networks; Section 3 introduces Classification and Regression Trees (CART). A brief introduction to TreeNet is given in section 4. Random forests are described in section 5. Section 6 describes Linear Genetic Programs (LGPs). Section 7 describes the dataset used for analysis. Feature selection and ranking are described on section 8. Section 9 presents the results and discussion. Finally, Section 10 presents the conclusions.

2 NEURAL NETWORKS

The Hidden Layer Learning Vector Quantization (HLVQ) is an algorithm recently proposed for classification of high dimensional data [12,13]. HLVQ is implemented in three steps. First, a multilayer perceptron is trained using back-

propagation. Second, supervised Learning Vector Quantization is applied to the outputs of the last hidden layer to obtain the code-vectors \vec{w}_{c_i} corresponding to each class c_i in which data are to be classified. Each example, \vec{x}_i , is assigned to the class c_k having the smallest Euclidian distance to the respective code-vector:

$$k = \min_j \left\| \vec{w}_{c_j} - \vec{h}(\vec{x}) \right\| \quad (1)$$

where \vec{h} is a vector containing the outputs of the hidden layer and $\|\cdot\|$ denotes the usual Euclidian distance. In the third step the MLP is retrained but with two differences regarding conventional multilayer training. First the error correction is not applied to the output layer but directly to the last hidden layer being the output layer ignored from now on. The second difference is in the error correction backpropagated to each hidden node:

$$E = \frac{1}{2} \sum_{i=1}^{N_h} \left(\vec{w}_{c_k} - \vec{h}(x_i) \right)^2 \quad (2)$$

where N_h is the number of hidden nodes. After retraining the MLP a new set of code-vectors,

$$\vec{w}_{c_i}^{new} = \vec{w}_{c_i} + \Delta \vec{w}_{c_i} \quad (3)$$

is obtained according to the following training scheme:

$$\Delta \vec{w}_{c_i} = \alpha(n) (\vec{x} - \vec{w}_{c_i}) \text{ if } \vec{x} \in \text{class } c_i,$$

$$\Delta \vec{w}_{c_i} = 0 \quad \text{if } \vec{x} \notin \text{class } c_i \quad (4)$$

The parameter α is the learning rate, which should decrease with iteration n to guarantee convergence. Steps two and three are repeated following an iterative process. The stopping criterion is met when a minimum classification error is found.

The distance of given example \vec{x} to each prototype is:

$$d_i = \left\| \vec{h}(\vec{x}) - \vec{w}_{c_i} \right\| \quad (5)$$

which is a proximity measure to each class?

After HLVQ is applied, only a small fraction of the hidden nodes is relevant for the code-vectors. Therefore HLVQ simplifies the network thus reducing the risk of overfitting.

3 CART

CART builds classification and regression trees for predicting continuous dependent variables (regression) and categorical predictor variables (classification) [14,15,16].

The decision tree begins with a root node t derived from whichever variable in the feature space minimizes a measure of the impurity of the two sibling nodes. The

measure of the impurity at node t , denoted by $i(t)$, is as shown in the following equation:

$$i(t) = - \sum_{j=1}^k p(w_j | t) \log p(w_j | t) \quad (8)$$

Where $p(w_j | t)$ is the proportion of patterns x_i allocated to class w_j at node t . Each non-terminal node is then divided into two further nodes, t_L and t_R , such that p_L , p_R are the proportions of entities passed to the new nodes t_L , t_R respectively. The best division is that which maximizes the difference given in:

$$\Delta i(s, t) = i(t) - p_i L(t_L) - p_i R(t_R) \quad (9)$$

The decision tree grows by means of the successive subdivisions until a stage is reached in which there is no significant decrease in the measure of impurity when a further additional division s is implemented. When this stage is reached, the node t is not sub-divided further, and automatically becomes a terminal node. The class w_j associated with the terminal node t is that which maximizes the conditional probability $p(w_j | t)$. No of nodes generated and terminal node values for each class are for the sample data set obtained from Diana, a database containing financial statements of about 780,000 French companies described in section VII, are presented in Table 1.

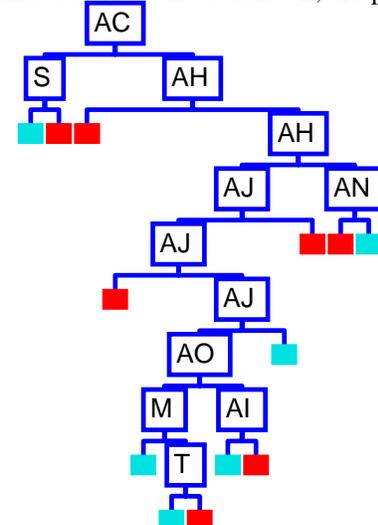


Figure 1. Tree for classifying normal vs. bankruptcy for unbalanced dataset

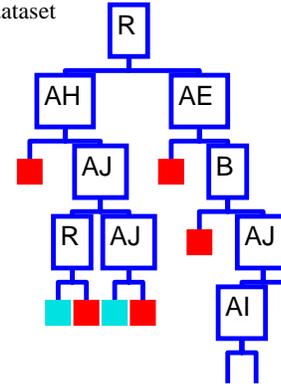


Figure 2. Tree for classifying normal vs. bankruptcy for balanced dataset

Table 1. SUMMARY OF TERMINAL NODES

| Class | No of Nodes | Terminal Node Value |
|--------------------|-------------|---------------------|
| Balanced Dataset | 38 | 0.38 |
| Unbalanced Dataset | 52 | 0.30 |

Figure 1 represents a classification tree generated for data set described in section VII for classifying normal vs. bankrupt for unbalanced dataset. Figure 2 represents a classification tree generated for data set described in section VII for classifying normal vs. bankrupt for balanced dataset. Each of the terminal node describes a data value; each record is classified into one of the terminal node through the decisions made at the non-terminal node that lead from the root to that leaf.

4 TREENET

In a TreeNet model classification and regression models are built up gradually through a potentially large collection of small trees. Typically consist from a few dozen to several hundred trees, each normally no longer than two to eight terminal nodes. The model is similar to a long series expansion (such as Fourier or Taylor's series) - a sum of factors that becomes progressively more accurate as the expansion continues. The expansion can be written as [14,17]:

$$F(X) = F_0 + \beta_1 T_1(X) + \beta_2 T_2(X) \dots + \beta_M T_M(X) \quad (10)$$

Where T_i is a small tree

Each tree improves on its predecessors through an error-correcting strategy. Individual trees may be as small as one split, but the final models can be accurate and are resistant to overfitting.

5 RANDOM FORESTS

A random forest is a classifier consisting of a collection of tree structured classifiers $\{h(x, \Theta_k), k=1, \dots\}$ where $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class of input X .

The common element in random trees is that for the k th tree, a random vector Θ_k is generated, independent of the past random vectors $\Theta_1, \dots, \Theta_{k-1}$ but with the same distribution; and a tree is grown using the training set and Θ_k , resulting in a classifier $h(x, \Theta_k)$ where x is an input vector. For instance, in bagging the random vector Θ is generated as the counts in N boxes resulting from N darts thrown at random at the boxes, where N is number of examples in the training set. In random split selection Θ consists of a number of independent random integers between 1 and K . The nature and dimensionality of Θ

depends on its use in tree construction. After a large number of trees are generated, they vote for the most popular class [14,18].

The random forest error rate depends on two things:

- ✓ The *correlation* between any two trees in the forest. Increasing the correlation increases the forest error rate.
- ✓ The *strength* of each individual tree in the forest. A tree with a low error rate is a strong classifier. Increasing the strength of the individual trees decreases the forest error rate.

6 LINEAR GENETIC PROGRAMS

Linear Genetic Programming (LGP) is a variant of the genetic programming technique that acts on linear genomes [10]. The linear genetic programming technique used for our current experiment is based on machine code level manipulation and evaluation of programs. Its main characteristic, in comparison to tree-based GP, is that the evolvable units are not the expressions of a functional programming language (like LISP); instead, programs of an imperative language (like C) are evolved [19,20,21].

In the automatic induction of machine code by GP, individuals are manipulated directly as binary code in memory and executed directly without passing through an interpreter during fitness calculation. The LGP tournament selection procedure puts the lowest selection pressure on the individuals by allowing only two individuals to participate in a tournament. A copy of the winner replaces the loser of each tournament. The crossover points only occur between instructions. Inside instructions the mutation operation randomly replaces the instruction identifier.

In GP an intron is defined as part of a program that has no influence on the fitness calculation of outputs for all possible inputs. Fitness F of an individual program p is calculated as

$$F(p) = \frac{1}{nm} \sum_{j=1}^n (o_{ij}^{pred} - o_{ij}^{des})^2 + \frac{w}{n} CE = MSE + wMCE \quad (10)$$

i.e., the mean square error (MSE) between the predicted

output (o_{ij}^{pred}) and the desired output (o_{ij}^{des}) for all n training samples and m outputs. The classification error (CE) is defined as the number of misclassifications. Mean classification error (MCE) is added to the fitness function while its contribution is determined by the absolute value of weight (w) [20].

7 DATA SET

We used a sample obtained from Diana, a database containing financial statements of about 780,000 French companies. The initial sample consisted of financial ratios

of 2,800 industrial French companies, for the years of 1998, 1999 and 2000, with at least 35 employees. From these companies, 311 were declared bankrupted in 2000 and 272 presented a restructuring plan (“Plan de redressement”) to the court for approval by the creditors. We decided not to distinguish these two categories as both signal companies in financial distress. The sample used for this study has 580 financial distressed firms, most of them small to medium size, with a number of employees from 35 to 400, corresponding to the year of 1999 - thus we are making bankruptcy prediction one year ahead.

This dataset includes companies from a wide range of industrial sectors with 30 financial ratios defined by COFACE¹ and included in the Diana database.

Error analysis

There are two types of errors for this classification problem: type I error and type II error. Type I error is the number of cases classified as healthy when they are really bankrupted, N_{01} , divided by the number of bankrupt companies N_1 :

$$e_I = \frac{N_{01}}{N_1} \quad (11)$$

Type II error is the number of companies classified as bankrupt when in reality they are healthy, N_{10} , divided by the total number of healthy companies, N_0 :

$$e_{II} = \frac{N_{10}}{N_0} \quad (12)$$

The total error is just:

$$e_{Total} = \frac{N_{10} + N_{01}}{N_0 + N_1} \quad (13)$$

For a balanced dataset with $N_0 = N_1$, the total error is average of both errors. The accuracy is defined as $1 - e_{Total}$.

Most companies on the verge of bankruptcy have heterogeneous patterns which are difficult to identify by any learning machine. Therefore type I error is in general higher than type II. Since the cost associated with this type of error is in general higher, in real applications global accuracy may not be the best performance indicator of the algorithm.

To study the effect of unbalanced datasets, we randomly added healthy companies in order to get the following ratios of bankrupted to healthy firms: dataset 1 (50/50), dataset 2 (36/64) and dataset 3 (28/72). Lower ratios put stronger bias towards healthy firms, deteriorating the generalization capabilities of the network and increasing type I error which is undesirable.

¹ Coface is a French credit risk provider

8 FEATURE SELECTION AND RANKING

The feature ranking for financial distress classification is similar in nature to various engineering problems that are characterized by:

- Having a large number of input variables $\mathbf{x} = (x_1, x_2, \dots, x_n)$ of varying degrees of importance to the output \mathbf{y} ; i.e., some elements of \mathbf{x} are essential, some are less important, some of them may not be mutually independent, and some may be useless or irrelevant (in determining the value of \mathbf{y})
- Lacking an analytical model that provides the basis for a mathematical formula that precisely describes the input-output relationship, $\mathbf{y} = \mathbf{F}(\mathbf{x})$
- Having available a finite set of experimental data, based on which a model (e.g. neural networks) can be built for simulation and prediction purposes
- Excess features can reduce classifier accuracy
- Excess features can be costly to collect
- If real time classification is important, excess features can reduce classifier operating speed independent of data collection
- If storage is important, excess features can be costly to store

Table 2. KEY FEATURES IDENTIFIED BY LGPS

| LGPs | Key Features |
|---|---|
| Top 6 features identified by LGP feature ranking algorithm | <ul style="list-style-type: none"> • Debt ratio • Financial autonomy • Collection period • Interest to sales (%) • Sales kEUR • Financial equilibrium ratio • Equity to Stable Funds • Working capital to current assets • Inventory days of sales |

9 RESULTS

We applied all methods to two datasets, balanced dataset (580 healthy companies and 580 companies with financial distress) and unbalanced dataset (1470 healthy companies and 580 companies with financial distress). In balanced dataset 500 randomly selected samples are used for training and 660 samples are used for testing. In unbalanced dataset 950 randomly selected samples are used for training and 1110 samples are used for testing. Detection rates and false alarms are evaluated for Diana, a database containing financial statements of about 780,000 French companies and the obtained results are used to form the ROC curves. The point (0,1) is the perfect classifier, since it classifies all positive cases and negative cases correctly. Thus an ideal system will initiate by identifying all the positive examples and so the curve will

rise to (0,1) immediately, having a zero rate of false positives, and then continue along to (1,1).

Figures 3 to 4 show the ROC curves of the detection models of LGP, Random forests and TreeNet.

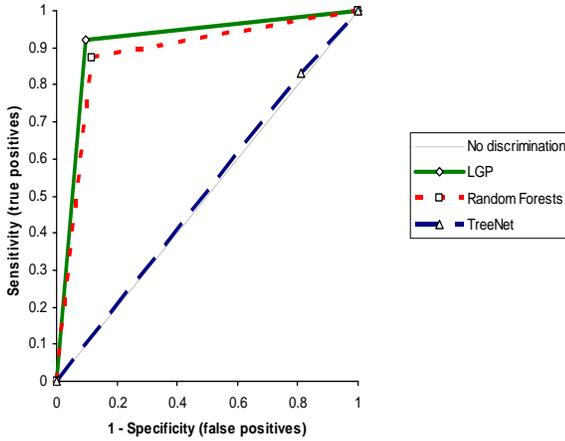


Figure 3. Classification accuracy for balanced dataset

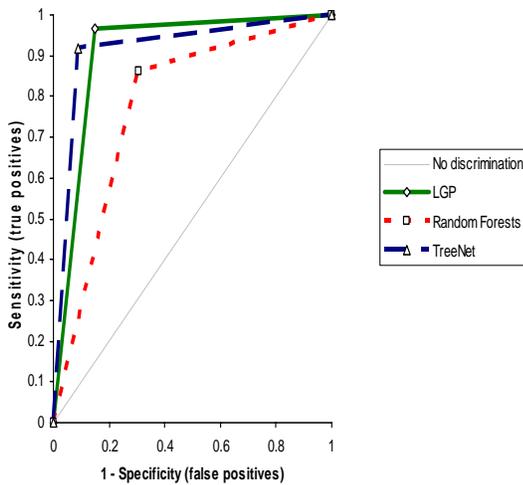


Figure 4. Classification accuracy for unbalanced dataset

In each of these ROC plots, the x-axis is the false positive rate, calculated as the percentage of normal companies considered as bankrupt; the y-axis is the detection rate, calculated as the percentage of bankrupt companies detected. A data point in the upper left corner corresponds to optimal high performance, i.e. high detection rate with low false alarm rate. Overall classification accuracies are given in table 3. Summary of number of false positives and false negatives is given in table 4.

Table 3. SUMMARY OF CLASSIFICATION ACCURACY

| | Balanced Dataset | Unbalanced Dataset |
|-------------|------------------|--------------------|
| CART | 83.3 | 87.7 |

| | | |
|-----------------------|-------------|-------------|
| TreeNet | 83.0 | 91.8 |
| Random Forests | 87.3 | 86.1 |
| LGP | 92.1 | 96.6 |
| HLVQ | 77.03 | 80.94 |
| MLP | 75.20 | 78.85 |

Table 4. SUMMARY OF FALSE POSITIVES (FP) AND NEGATIVES (FN)

| | Balanced Dataset | | Unbalanced Dataset | |
|-----------------------|------------------|-----------|--------------------|-----------|
| | FP | FN | FP | FN |
| CART | 0 | 55 | 0 | 95 |
| TreeNet | 0 | 56 | 0 | 57 |
| Random Forests | 0 | 42 | 0 | 97 |
| LGP | 0 | 26 | 0 | 24 |

Neural Networks

Multilayer Perceptrons (MLP) containing a single hidden layer from 5 to 20 nodes were tested in this problem. The best performing set was a hidden layer of 15 neurons trained by backpropagation with a learning rate of 0.1 and a momentum term of 0.25.

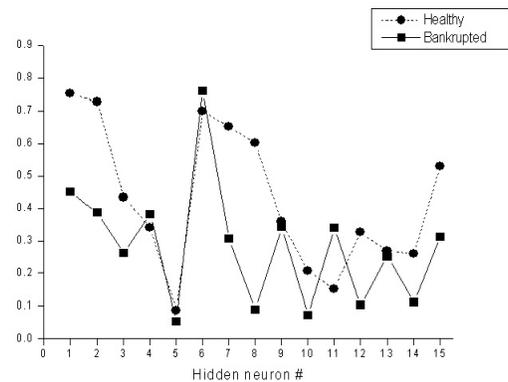


Fig. 5: HLVQ code-vectors

HLVQ was applied upon this MLP with a very fast convergence - only 8 iterations. Results obtained with MLP and HLVQ are presented in table 3.

Fig 5 presents the code vectors obtained by HLVQ corresponding to the two categories: healthy and bankrupt companies. Note that of the total 15 components, five are very similar, thus redundant. The remaining ten

components are the effective features used by HLVQ to classify data.

10 DISCUSSION AND CONCLUSIONS

Although the performance of the six methods is comparable in all datasets, we found that CART and Random Trees achieved consistently same results. Hidden Layer Learning Vector Quantization algorithm did not perform well when compare to CART, TreeNet, Random Forests and LGPs.

LGP performed the best on both the datasets, unbalanced dataset with an overall accuracy of 91.5 (0 false positives 24 false negatives), balanced dataset with an overall accuracy of 91.2 (0 false positives and 26 false negatives).

For unbalanced samples the overall accuracy improves. However, error type I, the most costly for banks, degrades in all machine learning methods applied. Therefore unbalance samples should be avoided.

Bankruptcy prediction is an important, interesting but difficult problem and further investigation is still needed. As a future work we plan to use a more complete data set including annual variations of important ratios from two or more years. As more inputs are added, feature selection will have to follow a more stringent scrutiny.

References

- [1] W. K. Beaver, Financial Ratios as Predictors of Failure, *Empirical Research in Accounting: Selected Studies*, 1966, supplement to volume 5, *Journal of Accounting Research* (1996) 71-102.
- [2] Altman, E. I. Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy, *Journal of Finance*, 23 (1968) 589-609.
- [3] R. A. Eisenbeis, Pitfalls in the Application of Discriminant Analysis in Business, Finance and Economics, *Journal of Finance*, 32 (3), June, (1977) 875-900.
- [4] D. Martin, Early Warning of Bank Failure: A Logit Regression Approach, *Journal of Banking and Finance*, 1 (1977) 249-276.
- [5] C. Tan, and H. A. Dihadjo, Study on Using Artificial Neural Networks to Develop an Early Warning Predictor for Credit Union Financial Distress with Comparison to the Probit Model, *Managerial Finance*, 27 (4), (2001) 56-77.
- [6] C. Zavgren, The Prediction of Corporate Failure: The State of the Art, *Journal of Accounting Literature*, 2 (1983) 1-38.
- [7] P.K. Coats and L.F. Fant, Recognising Financial Distress Patterns Using a Neural Network Tool, *Financial Management* (Autumn), (1996) 142-155.
- [8] F. Atiya, Bankruptcy prediction for credit risk using neural networks: A survey and new results, *IEEE Trans. Neural. Net.*, 4 (2001) 12-16.
- [9] G. Udo Neural Network Performance on the Bankruptcy Classification Problem, *Computers and Industrial Engineering*, 25 (1993) 377-380.
- [10] J. S. Grice and M. T. Dugan, The limitations of bankruptcy prediction models: Some cautions for the researcher, *Rev. of Quant. Finance and Account.*, 17 no. 2 (2001) 151.
- [11] F. Cucker and S. Smale, On the mathematical foundations of learning, *Bulletin of the American Mathematical Society* 39, n 1 (2001) 1-49.
- [12] A. Vieira and N. P. Barradas, A training algorithm for classification of high dimensional data *Neurocomputing*, 50C, (2003) 461-472.
- [13] A. Vieira, P. Castillo and J. Merelo, Comparison of HLVQ and GProp in the problem of bankruptcy prediction, *IWANN03 - International Workshop on Artificial Neural Networks*, L.J. Mira, ed., Springer-Verlag (2003) 665-662.
- [14] Salford Systems. *TreeNet, CART, Random Forests Manual*.
- [15] T. Hastie, R. Tibshirani, and J. H. Friedman, *The elements of statistical learning: Data mining, inference, and prediction*. Springer, 2001.
- [16] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees*. Wadsworth and Brooks/Cole Advanced Books and Software, 1986.
- [17] J. H. Friedman, Stochastic Gradient Boosting. *Journal of Computational Statistics and Data Analysis*, Elsevier Science, Vol. 38, PP. 367-378, 2002.
- [18] L. Breiman. Random Forests. *Journal of Machine Learning*, Vol. 45, pp. 5-32, 2001.
- [19] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, Cambridge, MA: The MIT Press, 1992.
- [20] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Reading, MA: Addison-Wesley, 1989.
- [21] AIM Learning Technology, <http://www.aimlearning.com>.