

# An Integrated Feature Selection and Classification Scheme

Yi Peng<sup>1</sup>, Gang Kou<sup>1</sup>, Daji Ergu<sup>2</sup>, Wenshuai Wu<sup>1</sup>, Yong Shi<sup>3,4</sup>

<sup>1</sup> School of Management and Economics,  
University of Electronic Science and Technology of China,  
No.4, Sec 2, North Jianshe Rd, Chengdu, 610054, China,  
kougang@yahoo.com

<sup>2</sup> Southwest University for Nationalities,  
Chengdu, 610200, China,  
ergudaji@163.com

<sup>3</sup> College of Information Science & Technology,  
University of Nebraska at Omaha,  
Omaha, NE 68182, USA,  
yshi@unomaha.edu

<sup>4</sup> CAS Research Center on Fictitious Economy and Data Sciences,  
Beijing 100080, China,  
yshi@gucas.ac.cn

**Abstract :** Irrelevant and redundant features may not only deteriorate the performances of classifiers, but also slow the prediction process. Another problem in prediction is the availability of a large number of classification models. How to choose a satisfactory classifier is an important yet understudied task. The goal of this paper is to propose an integrated scheme for feature selection and classifier evaluation in the context of prediction. It combines traditional feature selection techniques and multi-criteria decision making (MCDM) methods in an attempt to increase the accuracies of classification models and identify appropriate classifiers for different types of data sets.

**Keywords:** multi-criteria decision making (MCDM); feature selection; classification.

## 1. Introduction

Classification is one of the most important tasks in data mining [1] and is a commonly used approach in prediction. Research results reported on the performances of classification models diverge considerably [2-4]. How to choose a satisfactory classifier is an important yet understudied task.

Including all these attributes in the model-building process can deteriorate the performances of classifiers. In addition, high dimensionality can slow down the prediction process. Thus feature subset selection, which aims at selecting the most relevant and representative attributes to increase accuracy rates, is an essential step in the process of prediction.

This paper focuses on two issues in prediction: feature subset selection and classification algorithm evaluation. It proposes a research scheme that integrates traditional feature selection methods and multi-criteria decision making (MCDM) methods to improve the accuracy and reliability of prediction models.

The rest of this paper is organized as follows: section 2 reviews related works. Section 3 describes the research methodologies, including

the research design, feature selection methods, and MCDM methods. Section 4 summarizes the paper.

## 2. Related Works

### Feature subset selection techniques

Feature subset selection techniques can be broadly put into two categories. The first category is called a filter approach, which performs feature selection based on the characteristics of the dataset. The second category, on the other hand, assesses features by wrapping the learning algorithms into the selection procedure and is called a wrapper approach [5]. A filter approach is generally computationally more efficient than a wrapper approach because a wrapper approach needs to run the learning algorithms on the datasets to evaluate feature subsets.

### Classification algorithm evaluation

The problem of algorithm selection appears in diverse situations and has been investigated by many disciplines. Rice [6] formalized algorithm selection as abstract models with the problem space, the feature space, the criteria

space, the algorithm space, and the performance measures. The algorithm selection was presented as a learning task by the machine learning community [7]. Data mining field suggested multi-criteria based metrics to compare classification algorithms [8]. Rokach [9] suggested that the algorithm selection can be considered as a multiple criteria decision making (MCDM) problem and MCDM methods can be utilized to systematically choose the appropriate algorithm. Peng et al. [4] applied a set of MCDM methods to rank classification algorithms for the task of software defect detection.

### 3. Research Methodology

This section presents the research scheme and the major components of the scheme, including feature selection methods, MCDM methods, and classification algorithms.

Based on the findings of Myrtveit et al. [2], this study designs the research scheme with careful consideration of these three factors. First, multiple datasets, representing different sizes and domains, are selected for the experimental study. Second, five accuracy indicators are used to evaluate classifiers. Third, 10-fold cross-validation technique is applied to the sample datasets to select features. The research scheme is summarized in Figure 1.

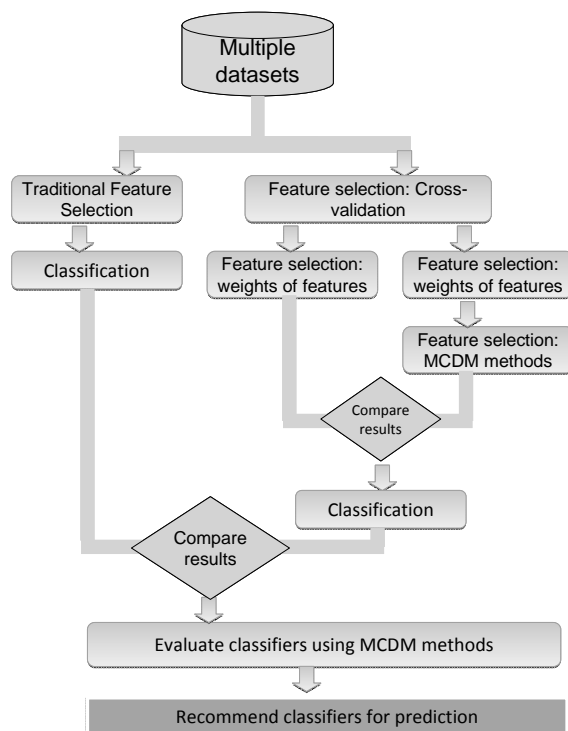


Figure 1. Research scheme

The datasets are handled by two different approaches. The first approach applies traditional feature selection and classification algorithms to the datasets to get prediction results. In the second approach, feature selection and classification are conducted in four steps. First, feature selection is conducted using traditional techniques. Features are then ranked using the proposed feature selection method. The third step employs MCDM methods to evaluate feature selection techniques and choose the better performed techniques. In the last step, the selected features are used in the classification. The classification results of the first and second approaches are compared to examine whether the proposed feature selection and MCDM methods can improve the prediction accuracy. The performances of classifiers are also evaluated using MCDM methods and a recommendation of classifiers for prediction is made based on their accuracy and reliability.

#### Proposed feature selection methods

As discussed in section 2, filter and wrapper are two main categories of feature subset selection techniques. The proposed feature selection approach makes use of both types of techniques. Weka is used throughout this study to implement feature selection and classification tasks [5]. Four types of feature selection evaluators are provided by Weka: CfsSubsetEval, ConsistencySubsetEval, ClassifierSubsetEval, and WrapperSubsetEval. CfsSubsetEval selects attributes that are highly correlated with the class label and have low inter-correlation. The second method projects training data onto attribute set to measure the level of consistency in the class values. The goal is to find the smallest subset with the highest consistency. The third and fourth methods belong to wrapper approach and they both use a classifier to evaluate attributes. The difference is that ClassifierSubsetEval measures attribute sets on the training data and WrapperSubsetEval uses cross-validation.

The proposed feature selection approach is based on the results of traditional feature selection methods. The importance of feature  $a_i$  is measured by a weight  $W_{a_i}$ , which is calculated as:

$$W_{a_i} = \frac{\sum_{j=1}^n \text{count}(b_{ij} > 0)}{n} \times \frac{\sum_{j=1}^n (b_{ij})^2}{\sum_{j=1}^n b_{ij}}, i = 1, 2, \dots, m$$

where  $n$  is the number of feature selection techniques,  $m$  is the number of features, and  $b_j$  is the numeric value generated by each feature selection technique. A feature with a high weight indicates that it is chosen by many feature selection methods and the variations of values generated by different feature selection techniques for this feature are low. Therefore features can be sorted according to their weights.

This study chooses WrapperSubsetEval, CfsSubsetEval, and ConsistencySubsetEval, as feature subset evaluators. WrapperSubsetEval uses nine classifiers as base learners for estimating the accuracy of subsets. The classifiers are described in a later section. Each classifier represents one feature selection method. Thus there are total eleven feature selection methods.

### MCDM methods

Multiple criteria decision making (MCDM) aims at solving decision problems with multiple objectives and often conflictive constraints [10-13]. Algorithm evaluation or selection normally needs to examine more than one criterion and can be modeled as MCDM problems [9]. Two types of algorithm evaluation tasks are considered: the evaluation of feature selection techniques and classification algorithms.

This section provides an overview of five MCDM methods, and explains how they are used in the experimental study to evaluate algorithms.

### Data envelopment analysis (DEA)

The original DEA model presented by Charnes et al [14] is called "CCR ratio model", which uses the ratio of outputs to inputs to measure the efficiency of DMUs. Assume that there are  $n$  DMUs with  $m$  inputs to produce  $s$  outputs.  $x_{ij}$  and  $y_{rj}$  represent the amount of input  $i$  and output  $r$  for  $DMU_j$  ( $j=1,2,\dots,n$ ), respectively. Then the ratio-form of DEA can be represented as:

$$\max h_0(u, v) = \sum_r u_r y_{ro} / \sum_i v_i x_{io}$$

subject to

$$\sum_r u_r y_{rj} / \sum_i v_i x_{ij} \leq 1 \text{ for } j = 1, \dots, n,$$

$$u_r, v_i \geq 0 \text{ for all } i \text{ and } r.$$

where the  $u_r$ 's and the  $v_i$ 's are the variables and the  $y_{ro}$ 's and  $x_{io}$ 's are the observed output and input values of the DMU to be evaluated (i.e.,  $DMU_o$ ), respectively (Cooper et al. 2004). The equivalent linear programming problem using the Charnes-Cooper transformation is [15]:

$$\max z = \sum_{r=1}^s \mu_r y_{ro}$$

subject to

$$\sum_{r=1}^s \mu_r y_{rj} - \sum_{i=1}^m v_i x_{ij} \leq 0$$

$$\sum_{i=1}^m v_i x_{io} = 1$$

$$\mu_r, v_i \geq 0$$

Comparing with the CCR model, a constraint

$$\sum_{j=1}^n \lambda_j = 1$$

is added in the BCC model.

These models can be solved using the simplex method for each DMUs. DMUs with value of 1 are efficient and others are inefficient.

Nakhaeizadeh and Schnabl [8] proposed to use DEA approach in data mining algorithms selection. They argued that in order to make an objective evaluation of data mining algorithms that all the available positive and negative properties of algorithms are important and DEA models are able to take both aspects into consideration. Positive and negative properties of data mining algorithms can be considered as output and input components in DEA, respectively. For example, the overall accuracy rate of a classification algorithm is an output component and the computation time of an algorithm is an input component. Using existing DEA models, it is possible to give a comprehensive evaluation of feature selection and classification algorithms.

### ELimination and Choice Expressing Reality (ELECTRE)

ELECTRE stands for *ELimination Et Choix Traduisant la REalité* (ELimination and Choice Expressing the REality) and was first proposed by Roy [16] to choose the best alternative from a collection of alternatives. ELECTRE III is chosen in this paper because it is appropriate for the sorting problem. The procedure can be summarized as follows [16]:

**Step 1:** define a concordance and discordance index set for each pair of alternatives  $A_j$  and  $A_k$ ,  $j, k = 1, \dots, m$ ;  $i \neq k$

**Step 2:** add all the indices of an alternative to get its global concordance index  $C_{ki}$

**Step 3:** define an outranking credibility degree  $\sigma_s(A_j, A_k)$  by combining the discordance indices and the global concordance index.

**Step 4:** define two outranking relations using descending and ascending distillation. Descending distillation selects the best alternative first and the worst alternative last. Ascending distillation selects the worst alternative first and the best alternative last.

**Step 5:** alternatives are ranked based on ascending and descending distillation processes.

### Preference Ranking Organisation METHod for Enrichment of Evaluations (PROMETHEE)

The PROMETHEE methods use pairwise comparisons and outranking relationships to choose the best alternatives. Since the purpose of this paper is to build a ranking of classification algorithms, PROMETHEE II is selected. The PROMETHEE II procedure described by Brans and Mareschal [17] was used:

**Step 1:** define aggregated preference indices.

Let  $a, b \in A$ , and let:

$$\begin{cases} \pi(a, b) = \sum_{j=1}^k p_j(a, b)w_j, \\ \pi(b, a) = \sum_{j=1}^k p_j(b, a)w_j. \end{cases}$$

where  $A$  is a finite set of possible alternatives  $\{a_1, a_2, \dots, a_n\}$ ,  $k$  represents the number of evaluation criteria and  $w_j$  is the weight of each criterion. Arbitrary numbers for the weights can be assigned by the DM. The weights are then normalized to ensure that  $\sum_{j=1}^k w_j = 1$ .

$\pi(a, b)$  indicates how  $a$  is preferred to  $b$  and  $\pi(b, a)$  indicates how  $b$  is preferred to  $a$ .  $P_j(a, b)$  and  $P_j(b, a)$  are the preference functions for alternatives  $a$  and  $b$ .

**Step 2:** calculate  $\pi(a, b)$  and  $\pi(b, a)$  for each pair of alternatives of  $A$ .

**Step 3:** define the positive and the negative outranking flow as follows:

The positive outranking flow:

$$\phi^+(a) = \frac{1}{n-1} \sum_{x \in A} \pi(a, x),$$

The negative outranking flow:

$$\phi^-(a) = \frac{1}{n-1} \sum_{x \in A} \pi(x, a).$$

**Step 4:** compute the net outranking flow for each alternative as follows:

$$\phi(a) = \phi^+(a) - \phi^-(a).$$

When  $\phi(a) > 0$ ,  $a$  is more outranking all the alternatives on all the evaluation criteria. When  $\phi(a) < 0$ ,  $a$  is more outranked.

### Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)

Hwang and Yoon [18] proposed the Technique for order preference by similarity to ideal solution (TOPSIS) method to rank alternatives over multiple criteria. It finds the best alternatives by minimizing the distance to the ideal solution and maximizing the distance to the nadir or negative-ideal solution. The following TOPSIS procedure adopted from Opricovic and Tzeng [19] is used:

**Step 1:** calculate the normalized decision matrix. The normalized value  $r_{ij}$  is calculated as:

$$r_{ij} = x_{ij} / \sqrt{\sum_{j=1}^J x_{ij}^2}, \quad j = 1, \dots, J; \quad i = 1, \dots, n.$$

where  $J$  and  $n$  denote the number of alternatives and the number of criteria, respectively. For alternative  $A_j$ , the performance measure of the  $i$ th criterion  $C_i$  is represented by  $x_{ij}$ .

**Step 2:** develop a set of weights  $w_i$  for each criterion and calculate the weighted normalized decision matrix. The weighted normalized value  $v_{ij}$  is calculated as:

$$v_{ij} = w_i r_{ij}, \quad j = 1, \dots, J; \quad i = 1, \dots, n.$$

where  $w_i$  is the weight of the  $i$ th criterion, and  $\sum_{i=1}^n w_i = 1$ .

**Step 3:** find the ideal alternative solution  $S^+$ , which is calculated as:

$$S^+ = \{v_1^+, \dots, v_n^+\} = \left\{ \left( \max_j v_{ij} \mid i \in I^+ \right), \left( \min_j v_{ij} \mid i \in I^- \right) \right\}$$

where  $I^+$  is associated with benefit criteria and  $I^-$  is associated with cost criteria.

**Step 4 :** find the negative-ideal alternative solution  $S^-$ , which is calculated as:

$$S^- = \{v_1^-, \dots, v_n^-\} = \left\{ \left( \min_j v_{ij} \mid i \in I^+ \right), \left( \max_j v_{ij} \mid i \in I^- \right) \right\}$$

**Step 5 :** Calculate the separation measures, using the  $n$ -dimensional Euclidean distance. The separation of each alternative from the ideal solution is calculated as:

$$D_j^+ = \sqrt{\sum_{i=1}^n (v_{ij} - v_i^+)^2}, j = 1, \dots, J.$$

The separation of each alternative from the negative-ideal solution is calculated as:

$$D_j^- = \sqrt{\sum_{i=1}^n (v_{ij} - v_i^-)^2}, j = 1, \dots, J.$$

**Step 6:** Calculate a ratio  $R_j^+$  that measures the relative closeness to the ideal solution and is calculated as:

$$R_j^+ = D_j^- / (D_j^+ + D_j^-), j = 1, \dots, J.$$

**Step 7 :** Rank alternatives by maximizing the ratio  $R_j^+$ .

### Vlsekriterijumska Optimizacija I Kompromisno Resenje (VIKOR)

VIKOR was proposed by Opricovic [20] for multicriteria optimization of complex systems. This paper uses the following VIKOR algorithm provided by Opricovic and Tzeng [19] in the experiment:

**Step 1 :** Determine the best  $f_i^*$  and the worst  $f_i^-$  values of all criterion functions,  $i = 1, 2, \dots, n$ .

$$f_i^* = \left\{ \begin{array}{l} \max_j f_{ij}, \text{ for benefit criteria} \\ \min_j f_{ij}, \text{ for cost criteria} \end{array} \right\}, j = 1, 2, \dots, J$$

$$f_i^- = \left\{ \begin{array}{l} \min_j f_{ij}, \text{ for benefit criteria} \\ \max_j f_{ij}, \text{ for cost criteria} \end{array} \right\}, j = 1, 2, \dots, J$$

Where  $J$  is the number of alternatives,  $n$  is the

number of criteria, and  $f_{ij}$  is the rating of  $i$ th criterion function for alternative  $a_j$ .

**Step 2:** Compute the values  $S_j$  and  $R_j$ ,  $j = 1, 2, \dots, J$ , by the relations

$$S_j = \sum_{i=1}^n w_i (f_i^* - f_{ij}) / (f_i^* - f_i^-),$$

$$R_j = \max_i [w_i (f_i^* - f_{ij}) / (f_i^* - f_i^-)],$$

where  $w_i$  is the weight of  $i$ th criteria,  $S_j$  and  $R_j$  are used to formulate ranking measure.

**Step 3:** Compute the values  $Q_j$ ,  $j = 1, 2, \dots, J$ , by the relations

$$Q_j = v(S_j - S^*) / (S^- - S^*) + (1-v)(R_j - R^*) / (R^- - R^*)$$

$$S^* = \min_j S_j, S^- = \max_j S_j,$$

$$R^* = \min_j R_j, R^- = \max_j R_j,$$

where the solution obtained by  $S^*$  is with a maximum group utility, the solution obtained by  $R^*$  is with a minimum individual regret of the opponent, and  $v$  is the weight of the strategy of the majority of criteria. The value of  $v$  is set to 0.5 in the experiment.

**Step 4 :** Rank the alternatives in decreasing order. There are three ranking lists:  $S$ ,  $R$ , and  $Q$ .

**Step 5 :** Propose the alternative  $a'$ , which is ranked the best by  $Q$ , as a compromise solution if the following two conditions are satisfied:

$$(a) Q(a'') - Q(a') \geq 1 / (J - 1); \quad (b)$$

Alternative  $a'$  is ranked the best by  $S$  or/and  $R$ .

If only the condition (b) is not satisfied, alternatives  $a'$  and  $a''$  are proposed as compromise solutions, where  $a''$  is ranked the second by  $Q$ . If the condition (a) is not satisfied, alternatives  $a', a'', \dots, a^M$  are proposed as compromise solutions, where  $a^M$  is ranked the  $M^{\text{th}}$  by  $Q$  and is determined by the relation  $Q(a^M) - Q(a') < 1 / (J - 1)$  for maximum  $M$ .

## Classification algorithms

The experimental study selects nine classifiers. The same set of classifiers is also used as base learners by feature subset evaluator WrapperSubsetEval. These classifiers belong to six categories of classification methods: trees, functions, Bayesian classifiers, lazy classifiers, rules, and miscellaneous classifiers. All of them are implemented in WEKA [5, 21, 22].

C4.5 decision tree is selected to represent the trees category. It constructs decision trees in a top-down recursive divide-and-conquer manner. The functions category includes linear logistic regression, radial basis function (RBF) network, and sequential minimal optimization (SMO). Bayesian classifiers category includes naïve Bayes. IB1, a basic nearest-neighbor instance-based learner provided by WEKA, represents lazy classifiers. An unknown instance is assigned to the same class as the training instance that is the closest to it measured by Euclidean distance. For the rules category, decision table and Repeated Incremental Pruning to Produce Error Reduction (RIPPER) rule induction were selected. Decision table builds a decision table majority classifier by selecting the right feature subsets. Instances not covered by a decision table can be determined by the nearest-neighbor method. RIPPER is an optimized version of incremental reduced error pruning (IREP). In addition, fuzzy lattice reasoning (FLR), which induces rules using fuzzy lattices, is chosen to represent the miscellaneous category.

## Performance measures

Five commonly used performance measures in classification are precision, true positive rate, false positive rate, F-measure, and the area under receiver operating characteristic (AUC) [23-24]. The following paragraphs briefly describe these measures.

- True Positive (TP): TP is the number of correctly classified fault-prone modules. TP rate measures how well a classifier can recognize fault-prone modules. It is also called sensitivity measure.

$$\text{True Positive rate/Sensitivity} = \frac{TP}{TP + FN}.$$

- False Positive (FP): FP is the number of non-fault-prone modules that is

misclassified as fault-prone class. FP rate measures the percentage of non-fault-prone modules that were incorrectly classified.

$$\text{False Positive rate} = \frac{FP}{FP + TN}.$$

- True Negative (TN): TN is the number of correctly classified non-fault-prone modules. TN rate measures how well a classifier can recognize non-fault-prone modules. It is also called specificity measure.

$$\text{True Negative rate/Specificity} = \frac{TN}{TN + FP}.$$

- False Negative (FN): FN is the number of fault-prone modules that is misclassified as non-fault-prone class. FN rate measures the percentage of fault-prone modules that were incorrectly classified.

$$\text{False Negative rate} = \frac{FN}{FN + TP}.$$

- Precision: This is the number of classified fault-prone modules that actually are fault-prone modules.

$$\text{Precision} = \frac{TP}{TP + FP}.$$

- Recall: This is the percentage of fault-prone modules that are correctly classified.

$$\text{Recall} = \frac{TP}{TP + FN}.$$

- F-measure: It is the harmonic mean of precision and recall. F-measure has been widely used in information retrieval.

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- AUC: ROC stands for Receiver Operating Characteristic, which shows the tradeoff between TP rate and FP rate. AUC represents the accuracy of a classifier. The larger the area, the better the classifier.

## Experimental design

The experiment was carried out according to the following process:

**Input:** datasets

**Output:** Ranking of classifiers

**Step 1:** Feature selection: apply eleven feature selection techniques to each dataset using WEKA 3.7 and calculate feature weights.

**Step 2:** Evaluate feature selection techniques using DEA, ELECTRE, PROMETHEE, TOPSIS, and VIKOR.

**Step 3:** Select the highly ranked feature selection techniques and use these techniques to re-calculate feature weights.

**Step 4:** Train and test the classification models on a randomly sampled partitions (i.e. 10-fold cross-validation) of each dataset with features selected by traditional feature selected technique. All methods are implemented using WEKA 3.7.

**Step 5:** Train and test the classification models on a randomly sampled partitions (i.e. 10-fold cross-validation) of each dataset with the features selected by Step 3. Compare these results with the results of Step 4.

**Step 6:** Evaluate classification algorithms using DEA, ELECTRE, PROMETHEE, TOPSIS, and VIKOR. All the MCDM methods are implemented using MATLAB.

**Step 7:** Generate four separate tables of the final ranking of classifiers provided by each MCDM method.

**END**

## 4. Conclusion remarks

The evaluation of feature selection techniques and classification algorithms can both be formulated as multi-criteria decision making (MCDM) problems. This paper developed an integrated research scheme integrates traditional feature selection methods and MCDM methods to improve the accuracy and reliability of prediction models. The proposed scheme first calculated a weight for each feature based on the results of traditional feature selection methods. It then applied MCDM methods to select the most relevant features, which were used in the following classification step. The performances of classifiers are also evaluated using MCDM methods and a recommendation of classifiers is made based on their accuracy and reliability.

## Acknowledgements

This research has been supported by grants from the National Natural Science Foundation of China (#70901011 and #71173028 for Yi Peng, #70901015 for Gang Kou, #70921061 for Yong Shi).

## REFERENCES

1. PENG, Y., G. KOU, Y. SHI, Z. CHEN, A **Descriptive Framework for the Field of Data Mining and Knowledge Discovery**, International Journal of Information Technology and Decision Making, vol. 7, no. 4, 2008, pp. 639-682.
2. MYRTVEIT, I., E. STENSRUD, M. SHEPPERD, **Reliability and Validity in Comparative Studies of Software Prediction Models**, IEEE Transactions on Software Engineering, vol. 31, no. 5, 2005, pp. 380-391.
3. PENG, Y., G. KOU, G. WANG, H. WANG, F. KO, **Empirical Evaluation of Classifiers for Software Risk Management**, International Journal of Information Technology and Decision Making, vol. 8, no. 4, 2009, pp. 749-768.
4. PENG, Y., G. WANG., H. WANG, **User Preferences based Software Defect Detection Algorithms Selection using MCDM**, Information Sciences, doi:10.1016/j.ins.2010.04.019, 2010.
5. WITTEN, I. H., E. FRANK, **Data Mining: Practical Machine Learning Tools and Techniques**, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.
6. RICE, J., **The Algorithm Selection Problem**, Advances in Computers, vol. 15, 1976, pp. 65-118.
7. SMITH-MILES, K. A., **Cross-Disciplinary Perspectives on Meta-learning for Algorithm Selection**, ACM Computing Surveys, vol. 41, no. 1, December 2008.
8. NAKHAEIZADEH, G., A. SCHNABL, **Development of Multi-Criteria Metrics for Evaluation of Data Mining Algorithms**, Proceeding of the Third International Conference on Knowledge Discovery and Data Mining (KDD'97), Newport Beach, California, August 14-17, 1997, pp. 37-42.

9. ROKACH, L., **Ensemble-based Classifiers**, *Artificial Intelligence Review*, vol. 33, no. 1-2, 2010, pp. 1-39.
10. ERGU, D., G. KOU, Y. PENG, Y. SHI, **A Simple Method to Improve the Consistency Ratio of the Pair-wise Comparison Matrix in ANP**, *European Journal of Operational Research*, vol. 213, no. 1, 2011, pp. 246-259.
11. ERGU, D., G. KOU, Y. SHI, Y. SHI, **Analytic Network Process in Risk Assessment and Decision Analysis**, *Computers & Operations Research*, doi:10.1016/j.cor.2011.03.005. 2011b
12. KOU, G., Y. SHI, S. Y. WANG, **Multiple Criteria Decision Making and Decision Support Systems - Guest Editor's Introduction**, DOI:10.1016/j.dss.2010.11.027, *Decision Support Systems*, vol. 51, no. 2, 2011, pp. 247-249.
13. PENG, Y., G. KOU, G. WANG, Y. SHI, **FAMCDM: A Fusion Approach of MCDM Methods to Rank Multiclass Classification Algorithms**, *Omega*, vol. 39, no. 6, 2011, pp. 677-689.
14. CHARNES, A., W. W. COOPER, E. RHODES, **Measuring the Efficiency of Decision Making Units**, *European Journal of Operational Research*, vol. 2, no. 6, 1978, pp. 429-444.
15. COOPER, W. W., L. M. SEIFORD, J. ZHU, **Data Envelopment Analysis: History, Models and Interpretations**, Chapter 1, 1-39, in Cooper, W. W. Seiford, L. M., Zhu, J. (eds), *Handbook on Data Envelopment Analysis*, Kluwer Academic Publisher, Boston, 2004.
16. ROY, B., **Classement et choix en presence de points de vue multiples (la methode ELECTRE)** *R.I.R.O.*, vol. 8, 1968, pp. 57-75.
17. BRANS, J. P., B. MARESCHAL, **PROMETHEE Methods**, In *Multiple Criteria Decision Analysis: State of the Art Surveys*, J. Figueira, V. Mousseau and B. Roy (eds.), Springer, New York, 2005, pp. 163-195.
18. HWANG, C. L., K. YOON, **Multiple Attribute Decision Making Methods and Applications**, Springer, Berlin Heidelberg, 1981.
19. OPRICOVIC, S., G. H. TZENG, **Compromise Solution by MC DM Methods: A Comparative Analysis of VIKOR and TOPSIS**, *European Journal of Operational Research*, vol. 156, 2004, pp. 445-455.
20. OPRICOVIC, S., **Multicriteria Optimization of Civil Engineering Systems**. Faculty of Civil Engineering, Belgrade, 1998.
21. KOU, G., C. LOU, **Multiple Factor Hierarchical Clustering Algorithm for Large Scale Web Page and Search Engine Clickstream Data**, DOI: 10.1007/s10479-010-0704-3, *Annals of Operations Research*, 2010.
22. KOU, G., Y. LU, Y. PENG, Y. SHI, **Evaluation of Classification Algorithms using MCDM and Rank Correlation**, *International Journal of Information Technology & Decision Making*, vol. 11, no. 1, 2012, pp. 197-225.
23. PENG, Y., G. KOU, G. WANG, W. WU, Y. SHI, **Ensemble of Software Defect Predictors: An AHP-based Evaluation Method**, *International Journal of Information Technology & Decision Making*, vol. 10, no. 1, 2011, pp. 187-206.
24. PENG, Y., G. WANG, G. KOU, Y. SHI, **An Empirical study of Classification Algorithm Evaluation in Financial Risk Management**, *Applied Soft Computing*, vol. 11, no. 2, 2011, pp. 2906-2915.