An Embedded Feature Selection Method for Imbalanced Data Classification

Haoyue Liu, *Student Member, IEEE*, MengChu Zhou, *Fellow, IEEE*, and Qing Liu

***Abstract*—Imbalanced data is one type of datasets that are frequently found in real-world applications, e.g., fraud detection and cancer diagnosis. For this type of datasets, improving the accuracy to identify their minority class is a critically important issue. Feature selection is one method to address this issue. An effective feature selection method can choose a subset of features that favor in the accurate determination of the minority class. A decision tree is a classifier that can be built up by using different splitting criteria. Its advantage is the ease of detecting which feature is used as a splitting node. Thus, it is possible to use a decision tree splitting criterion as a feature selection method. In this paper, an embedded feature selection method using our proposed weighted Gini index (WGI) is proposed. Its comparison results with Chi2, F-statistic and Gini index feature selection methods show that F-statistic and Chi2 reach the best performance when only a few features are selected. As the number of selected features increases, our proposed method has the highest probability of achieving the best performance. The area under a receiver operating characteristic curve (ROC AUC) and F-measure are used as evaluation criteria. Experimental results with two datasets show that ROC AUC performance can be high, even if only a few features are selected and used, and only changes slightly as more and more features are selected. However, the performance of F-measure achieves excellent performance only if 20% or more of features are chosen. The results are helpful for practitioners to select a proper feature selection method when facing a practical problem.**

***Index Terms*—Classification and regression tree, feature selection, imbalanced data, weighted Gini index (WGI).**

I. Introduction

In recent years, class imbalance is one of the problems that receive more and more attention in the area of big data analysis when one deals with real-world datasets arising from, e.g., telecommunications, biology, ecology, and fraud detection. They often exhibit the character of imbalanced datasets [1]. Class imbalance occurs when the number of observations falling within one class is larger than that in another class but the former class is not as important as the latter. This leads to the bias-to-majority problem when an imbalanced dataset is handled with some traditional machine learning algorithms. It is stated that the class with rare events often gets misclassified by directly using such algorithms, because they assume that the sizes of both majority class (known as a "negative" class) and minority one (known as a "positive" class) are approximately equal. When this problem occurs on an imbalanced dataset, even though a high overall classification accuracy can be observed, the accuracy to classify the more important minority class can be disappointedly low. Another important property of an imbalanced dataset, which needs our attention, is the imbalanced ratio, which is calculated by dividing the number of observations in the majority class by that in the minority class. In the wake of increasing imbalanced ratio, the bias-to-majority problem gets worse.

Two standard methods have been used to deal with a class imbalance problem [2]:

A) Data Level: Modify majority or minority class distribution to allow the classification algorithm to equally treat an imbalanced dataset. This kind of modification is called resampling that can be obtained by increasing the number of minority data samples or decreasing the number of majority ones. They are known as oversampling and undersampling, respectively. The easiest way for oversampling and undersampling is to randomly eliminate some data from the majority class or randomly replicate the minority class patterns, respectively. Besides randomly resampling an imbalance dataset, there are some other useful techniques to balance it. For instance, Ramentol *et al*. [3] propose a method that combines oversampling and undersampling methods to get a balanced dataset, which achieves the limit of loss of information and addition of synthetic patterns. Kang *et al*. [4] propose a noise-filtered undersampling scheme to eliminate the noisy minority data before executing resampling method for imbalanced classification.

B) Algorithm Level: This type of solution adjusts the existing classification algorithms to optimize a classifier's accuracy toward the minority class. The most popular method at the algorithm level is a cost-sensitive method that associates different misclassification costs for each class during a learning process. An effective ensemble of the cost-sensitive decision tree method is proposed in [5]. Lopez *et al*. [6] introduce a Chi-FRBCS-BigDataCS that renovates the simple Chi2 approach. It leads to a better performance than the latter without increasing the computational time, when handling an imbalanced dataset.

Manuscript received September 20, 2018; revised December 31, 2018; accepted February 21, 2019. This work was supported in part by the National Science Foundation of USA (CMMI-1162482). Recommended by Associate Editor Kao-shing Hwang. *(Corresponding author: Haoyue Liu.)*

Citation: H. Y. Liu, M. C. Zhou, and Q. Liu, "An embedded feature selection method for imbalanced data classification," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 3, pp. 703–715, May 2019.

H. Y. Liu and Q. Liu are with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: hl394@njit.edu; qing.liu@njit.edu).

M. C. Zhou is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA, and also with the Institute of Systems Engineering, Macau University of Science and Technology, Macau 999078, China (e-mail: zhou@njit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2019.1911447

In this work, we propose a cost-sensitive embedded feature selection method to solve the class imbalance problem. Feature selection is an essential technique for pre-processing a dataset. The quality of data can be enhanced by eliminating noisy, irrelevant and redundant features through selecting an optimal subset of features. Also, computational time can be reduced, when utilizing a subset of features instead of all in the original one. For an imbalanced dataset, feature selection means to distinguish which subset of features are more related to the minority class information. The work [7] presents an effective feature selection method called the neighborhood relationship preserving score for multi-label classification. A backward elimination approach for feature selection and embedded support vector machines (SVM) is proposed in [8], which is appropriate to select relevant features that are utilized to discriminate among classes for an imbalanced dataset. Feature selection results affect some classification algorithms to achieve a desired performance. Also, more understandable and applicable application results can be obtained after irrelevant features are abandoned [9].

The rest of this paper is organized as follows. A brief review of the related work is presented in Section II. The proposed method is given in Section III. Section IV introduces imbalanced datasets, describes experimental setting and analyzes experimental results. Section V concludes this work.

II. Related Work

Feature selection is crucial in data pre-processing of imbalanced datasets by identifying discriminative features and reducing computational time [10], [11]. When machine learning algorithms are used to handle an imbalanced dataset, they tend to encounter a bias-to-majority problem. However, if a subset of features that are more related to minority class information are selected as training data, classification results may be biased toward the minority class, which is not desired in practice [12]. Methods for feature selection are normally divided into three categories: filter, wrapper, and embedded methods. As shown in Fig. 1, they select a subset of features differently.

Filter methods directly rank features according to different performance evaluation metrics and select top-N features that contain the highest score. They are based on the intrinsic characteristics of a dataset. The feedback from classification results cannot influence the already selected features. Filter methods are considered as the lightest computational one among three feature selection methods. Chi2 is a frequently-used filter method. It shows the degree of correlation between each of features and class distribution [13]. If they are independent, the result is zero. The result's absolute value is larger if a given feature is more dependent on class distribution. The work [14] presents a new correlation-based feature selection method to identify relevant and redundancy features at the same time. This study indicates the efficiency and effectiveness of such methods when dealing with high-dimension datasets. However, the work [15] notes that filter methods have the disadvantage of not interacting with a classifier algorithm eventually used. Another disadvantage is that most of filter methods are universal in nature, meaning that they do not take into consideration the values of other features.

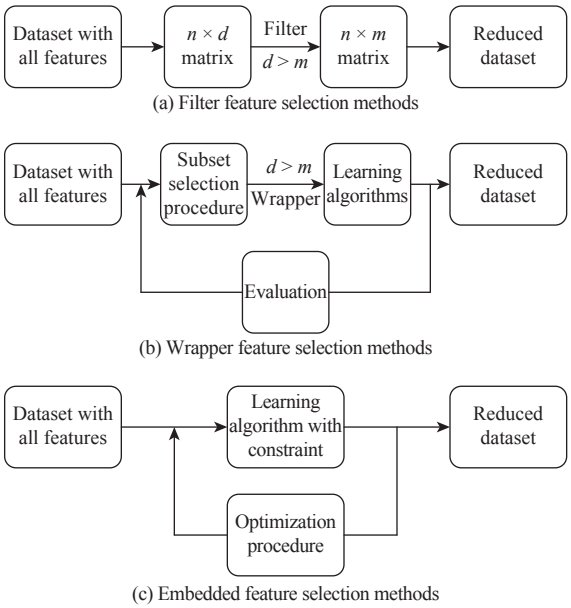


Fig. 1. Flowchart of filter, wrapper, and embedded feature selection methods.

In wrapper methods, a subset of features are selected when a model is trained through classification algorithms. These methods are generally computationally expensive [16]. Some common wrapper methods are forward selection, backward elimination, and recursive feature elimination. Forward selection [17] starts with having zero feature in a dataset and adds one feature which contains the highest classification performance in each iteration. It stops adding the feature, when no improvement of classification performance is observed, or all features are added to the model. Backward elimination [18] is the opposite of forward selection. It starts with all features and eliminates the least significant feature in each iteration till classification performance is not changed. Instead of using a single wrapper method such as forward selection, the study [19] presents a novel method, called a simulated annealing generic algorithm (SAGA), which incorporates existing wrapper methods into a single solution. The results demonstrate that the combined method reduces the weaknesses of wrapper methods that inherently have when they are individually used. Maldonado *et al*. [20] introduce a wrapper method based on SVM. An over-fitting problem can be avoided with their method, because of its inherent capability of splitting a dataset. They also find that their method using a Kernel function provides better results than those without using such function. One drawback of their proposed algorithm is that they use backward elimination, which requires high computational cost when working with high-dimension datasets. Zhu *et al.* [21] propose an improved NSGA-III algorithm that uses two strategies and a predefined multiple targeted search for feature selection in intrusion detection systems. The higher classification accuracy and lower computational complexity achieved for classes that have only a small number of instances. Moayedikia *et al*. [22] propose a novel feature selection method that uses the symmetrical uncertainty and harmony search. It can well solve the case that has a set of features with the same

assigned weights. Embedded methods are similar to wrapper methods. Feature selection is linked to classification algorithms. This link in embedded methods is stronger than that in wrapper methods. Embedded methods are a type of combination of filter and wrapper methods. It uses classification algorithms that contain their own built-in ability to select features. One type of embedded feature selection methods is to use inherent characteristics of decision tree algorithms: iterative dichotomiser 3 (ID3), random forest, and classification and regression tree (CART) [23]. Another commonly used one is to use least absolute shrinkage and selection operator (LASSO) regression l_1 regularization, which adds a penalty equivalent to an absolute value of the magnitude of coefficients.

The work [24] presents an embedded feature selection method that uses two SVM formulations, i.e., cost sensitive SVM and support vector data description, and deals with the high-dimensional class-imbalanced datasets. It can obtain high average predictive performance.

III. Proposed Approach

In this section, we introduce the proposed embedded feature selection method. The main component of the proposed method is a decision tree algorithm called CART. According to the CART structure, we add an index-weighting method to deal with an imbalanced dataset, to be explained later.

A. CART

A decision tree algorithm is a commonly-used learning algorithm in classification and regression. It has been extensively applied in text classification [25], [26], spam detection [27], and categorical loan data [28]. In a classification problem, a decision tree represents a classification process that is based on features. Also, it can be viewed as a combination of a set of if-then rules.

A decision tree consists of nodes and directed edges. A node can be categorized into an internal or leaf node. Each of internal nodes represents a feature. A leaf node represents a class. A tree starts at the root node. An example of the decision tree is shown in Fig. 2. The root is "Age > 40", "Education", "House" and "Income > 1000" are internal nodes.

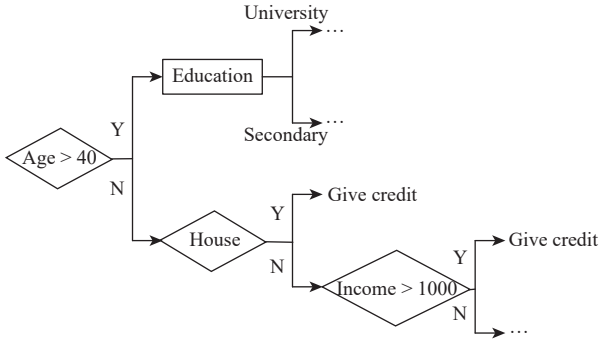


Fig. 2. An example of a decision tree.

A decision tree is built in a top-down fashion, by choosing a feature that best splits a dataset at each step. Different algorithms are built on different splitting criteria for measuring the so-called "best" feature. Quinlan [29] proposes a decision tree algorithm. The splitting criterion is based on an impurity measure called ID3. However, this algorithm fails to cope with continuous features. Only categorical values can be supported. Then, Quinlan [30] introduces C4.5 that can handle continuous attributes. The splitting criterion in C4.5 uses the concept of information entropy. Chen *et al.* [31] use two commonly splitting criteria: information gain and Gini index when constructing a tree for bioinformatics. Their method can deal with both discrete and continuous data types. By default, such a tree attempts to cover all possible outcomes in its structure. It leads to the over-fitting problem.

Different splitting criteria are expected to result in different decision trees. ID3 and C4.5 select nodes based on information gain that depends on the concept of entropy. CART uses a Gini index as a metric to test a node's impurity. This work focuses on CART.

Breiman [32] present CART in 1980s. It is a binary decision tree that is constructed by splitting data into two parts with maximum homogeneity. The advantages of using this algorithm in feature selection are as follows:

1) CART results are invariant to monotone transformations of its independent variables. Changing one or more variables to the logarithm or square root does not change the structure of the tree. Only the splitting values are different.

2) CART can easily handle outliers. Outliers can negatively affect the results of some learning algorithms. The splitting criterion for CART can easily handle them. It isolates them in a separate node.

3) Features can be repeatedly used in CART. When using a decision tree algorithm for feature selection, we need to calculate how often one feature is selected as a node in a tree. In a CART tree, it is easy to observe this frequency. Higher-frequency occurrence of a feature means that it has a higher chance to be selected.

4) CART can easily handle both continuous and categorical features in a binary tree. For a continuous feature, the Gini index can traverse all possible values, and then calculates each node's impurity corresponding to each value. Finally, CART selects a feature that contains the highest node impurity as the splitting node.

B. Weighted Gini Index

A decision tree is constructed by using a recursive method that requires to find the best splitting node once a time. Each of splitting nodes is selected by iterating through all possible features and all possible values. The Gini index is the splitting criterion that is used as an empirical impurity reduction measure in CART. The Gini index presents the impurity of the model. The smaller it is, the lower the impurity is, and the better the corresponding features are. It can be computed as follows:

Let K be a set of classes, p_k be the probability that class k has happened. The Gini index is defined as

$$G(p) = \sum_{k=1}^{K} p_k(1-p_k) = 1 - \sum_{k=1}^{K} p_k^2. \tag{1}$$

For a given dataset D, suppose that there are K classes and

the number of samples in class k is C_k. D's Gini index is expressed as

$$G(D) = 1 - \sum_{k=1}^{K} \left(\frac{|C_k|}{|D|} \right)^2. \tag{2}$$

If D is divided into D_1 and D_2 based on a certain value of feature x, Gini index of D can be computed as

$$G(D, x) = \frac{|D_1|}{|D|} G(D_1) + \frac{|D_2|}{|D|} G(D_2). \tag{3}$$

The splitting node is selected by ranking $G(D, x)$ in ascending order. Consider a binary classification case. After determining the splitting node, the splitting result can be given in Table I, which is similar to a confusion matrix.

TABLE I
SPLITTING RESULT FOR ONE FEATURE

		Root class	
		Positive (D_1)	Negative (D_2)
Original class (D)	Positive	True positive (TP)	False negative (FN)
	Negative	False positive (FP)	True negative (TN)

According to Table I, Gini indices are

$$G(D) = 1 - \left(\frac{TP + FN}{N} \right)^2 - \left(\frac{FP + TN}{N} \right)^2 \tag{4}$$

$$G(D_1) = 1 - \left(\frac{TP}{TP + FP} \right)^2 - \left(\frac{FP}{TP + FP} \right)^2 \tag{5}$$

$$G(D_2) = 1 - \left(\frac{FN}{FN + TN} \right)^2 - \left(\frac{TN}{FN + TN} \right)^2 \tag{6}$$

$$G(D, x) = \frac{TP + FP}{N} G(D_1) + \frac{FN + TN}{N} G(D_2) \tag{7}$$

$$\Delta G(x) = G(D) - G(D, x) \tag{8}$$

where the original dataset D contains two classes which are labeled as positive and negative. When selecting feature x as a splitting node, the original dataset is separated into two sub-datasets D_1 and D_2. Assume that D_1 is positive and D_2 is negative. TP and FP are the numbers of samples in sub-dataset D_1 whose original label is positive and negative, respectively. FN and TN are the numbers of samples in sub-dataset D_2 whose original label is positive and negative, respectively. N is the total number of samples in D. $G(D, x)$ is the impurity value for D when using feature x as a splitting node. When it equals 0, all samples have the same label. $\Delta G(x)$ denotes the decrement in impurity. Its larger value indicates the higher probability to choose x as a splitting node.

It is clear that the above Gini index related computation assumes that D_1 and D_2 have equal distribution. The calculated result would be biased to the majority class if an imbalanced dataset is handled, thus resulting in a biased-to-majority problem. To prevent the problem, this work proposes a weighted Gini index to increase the chance to choose the features which tend to bias the minority class [33]. Its formation is shown as follows:

$$G(D) = 1 - \left(\frac{(TP + FN) \times w}{N^*} \right)^2 - \left(\frac{FP + TN}{N^*} \right)^2 \tag{9}$$

$$G(D_1) = 1 - \left(\frac{TP \times w}{TP \times w + FP} \right)^2 - \left(\frac{FP}{TP \times w + FP} \right)^2 \tag{10}$$

$$G(D_2) = 1 - \left(\frac{FN \times w}{FN \times w + TN} \right)^2 - \left(\frac{TN}{FN \times w + TN} \right)^2 \tag{11}$$

$$G(D, x) = \frac{(TP + FP) \times w}{N^*} G(D_1) + \frac{FN + TN}{N^*} G(D_2) \tag{12}$$

$$N^* = (TP + FN) \times w + FP + TN \tag{13}$$

where w is a newly introduced weight. In this paper, it is set to be the multiple of imbalanced ratio of a given dataset.

Table II illustrates the meaning of weighted Gini index method that is used as the splitting criterion for an imbalanced dataset. Based on (4)−(8), $\Delta G(A) = 0.138 \times 10^{-3}$ and $\Delta G(B) = 0.118 \times 10^{-3}$. Feature A is chosen as the splitting node by using a Gini index method, even though only a small amount of minority data is correctly classified. However, when (9)−(13), which correspond to the weighted Gini index, are applied as the splitting criterion, $\Delta G(A) = 0.088$ and $\Delta G(B) = 0.324$. In this case, Feature B is chosen as a splitting node, since it achieves a higher classification accuracy for the minority class.

C. Feature Selection Method

A root, branches, nodes and leaves constitute a standard decision tree. Each non-leaf node represents a test on a feature, and each branch represents the output of that feature in a range, while each leaf node contains one class. Because of its advantages, we select CART as a feature selection method. The pseudo-code incorporating the proposed weighted Gini index into it is given in Algorithm 1.

Algorithm 1 CART for feature selection

Input: a dataset D

```
Tree = {}
if D is "pure" OR other stopping criteria are met then
  terminate
end if
for all feature f do
  Compute Gini index or weighted Gini index if we split on f
 end for
f_best = Best feature according to the above value
Tree = Create a decision node that tests f_best in the root
D_V = Induced sub-datasets from D based on f_best
for all D_V do
  Tree_V = Decision tree (D_V)
  Attach Tree_V to the corresponding branch of Tree
end for
return Tree
```

According to the structure of a decision tree, we propose a feature selection method [34]. Based on a training dataset, a decision tree is constructed from top to bottom. Each node is

TABLE II
SPLITTING RESULTS FOR TWO FEATURES

Original class (D)	Root class (Feature A)		Root class (Feature B)	
	Positive (D_1)	Negative (D_2)	Positive (D_1)	Negative (D_2)
Positive	3	7	8	2
Negative	10	9990	100	9990

chosen by comparing the Gini index or weighted one for each feature in Step 6 of Algorithm 1. A tree is established through repeating Step 6 till the maximum depth of the tree is reached or other stopping criteria are met. After computing the weighted Gini index value in Step 6 for all features and all possible values, we rank the weighted Gini index values in descending order. If one or more features contain the highest weighted Gini index value, the frequency of these features is increased by one. However, a splitting node is only decided by randomly choosing one. Increasing the features' frequency does not stop until the same prior mentioned stopping criterion is met. After the CART tree stops its growth, each feature has its corresponding frequency's score. The highest feature's frequency in a decision tree means that the feature is more relevant to the class. It also represents that this feature is the best one to distinguish the data. We rank scores in descending order, and then select top ranked features. The time complexity of the proposed feature selection method is same as building a CART tree based on Algorithm 1. Assuming that the number of instances is n and the number of features is m, the time complexity of building one level of the decision tree is $O(n \times m)$, and the depth of a CART tree is $\log_2(n)$. Thus, the complexity of the proposed feature selection method is $O(n \times m \times \log_2 n)$.

Next, we evaluate the performance of the proposed method and several other well-known ones.

IV. EXPERIMENTAL RESULTS

In this section, the performance of the proposed feature selection method with weighted Gini index (WGI) is examined and compared with three methods: Chi2, F-statistic and Gini index feature selection methods. Because Chi2 and F-statistic are two frequently-used filter methods, we introduce two benchmark datasets and performance measures. The comparison results are discussed in the end.

A. Benchmark Datasets

In order to test the performance of our proposed method, two benchmark datasets are used. The statlog (ladsat satellite) and letter recognition datasets are from the UCI machine learning repository. Table III summarizes their numbers of instances, numbers of features, and imbalance ratios.

The statlog dataset consists of multi-spectral values of pixels in 3 × 3 neighborhoods in a satellite image, and the classification is associated with the central pixel in each neighborhood. This dataset contains 6435 instances with 36 features. The features are numerical, and their values range from 0 to 255. This is a 6-class classification problem. For this multi-class dataset, we select the least frequent class label as the minority and the rest as the majority. Consequently, class 4 is treated as the minority class and the rest of classes are combined into one class as the majority one.

TABLE III
SUMMARY OF BENCHMARK DATASETS

Dataset	Instances	Features	Imbalanced ratio
Statlog	6435	36	9.3
Letter	20 000	16	24.3–27.25

The letter recognition dataset contains 20 000 instances and 16 features. The instances are of integer values. The objective of this dataset is based on 16 features to classify 26 capital English alphabets.

All features show different characteristics of letter images. The original dataset is 26 English letters from A to Z that correspond to 26 different classes. In this experiment, we only focus on binary classification problems. Hence, we transfer this multiple classification problem to 26 different binary classification problems, i.e., each letter with respect to the rest as shown in Table III. This leads to 26 imbalanced datasets. Their details are described in Table IV.

TABLE IV
SUMMARY OF LETTERS

Target	$\lvert S_M\rvert/\lvert S_m\rvert$	Target	$\lvert S_M\rvert/\lvert S_m\rvert$
A vs. ¬A	25.35	N vs. ¬N	25.54
B vs. ¬B	26.11	O vs. ¬O	26.56
C vs. ¬C	27,17	P vs. ¬P	24.91
D vs. ¬D	24.84	Q vs. ¬Q	25.54
E vs. ¬E	26.04	R vs. ¬R	26.38
G vs. ¬G	25.87	T vs. ¬T	25.13
H vs. ¬H	27.25	U vs. ¬U	24.6
I vs. ¬I	26.49	V vs. ¬V	26.18
J vs. ¬J	26.77	W vs. ¬W	26.6
K vs. ¬K	27.06	X vs. ¬X	26.74
L vs. ¬L	26.28	Y vs. ¬Y	25.44
M vs. ¬M	25.25	Z vs. ¬Z	27.25

B. Experimental Design

In our experiments, two filter feature selection methods: Chi2 and F-statistic, and one embedded feature selection method: Gini-index feature selection (GI-FS) are compared with our proposed method: weighted Gini index feature selection (GI-FS^w). For GI-FS^w, imbalanced ratio (ρ) and 1.5 times of that are two different weights that are selected in this paper. The 6-fold cross validation is used to avoid overfitting in the experiment.

In order to test the performance of four feature selection methods, the extreme gradient boosting (Xgboost) classifier [35] is used to test the classification performance of a dataset. For our experiment, its parameters are defined as: the learning rate equals 1, the number of sequential trees equals 200 and the maximum depth of a tree equals 10.

C. Performance Measure

Two evaluation metrics, i.e., area under the curve (AUC) of receiver operating characteristic (ROC) and F-measure are used to test the effectiveness of classification algorithms. To get the result of these two performance measures, a confusion matrix is needed. It is a table that consists of two rows and two columns that present the numbers of true positive (*TP*), false negative (*FN*), false positive (*FP*) and true negative (*TN*). In a binary classification, a confusion matrix has four outcomes only:

True positive (*TP*): positive data correctly classified as positive.

False negative (*FN*): positive data classified as negative.

False positive (*FP*): negative data classified as positive.

True negative (*TN*): negative data correctly classified as negative.

According to previous four outcomes from the confusion matrix, several evaluation measures can be described as follows:

$$\text{True positive rate } (TPR) = \frac{TP}{TP+FN} \tag{14}$$

$$\text{False positive rate } (FPR) = \frac{FP}{TN+FP} \tag{15}$$

ROC is a curve that models the trade-off between TRP and FPR [36], [37], which is constructed in a two-dimensional space. AUC means the area under the curve of ROC. The diagonal of AUC presents a random predictor: AUC is 0.5. The random predictor is commonly used as a baseline to see whether a model is useful or not.

F-measure [38] is a performance measure that is computed based on precision and recall. It is one of the popular metrics that evaluates the classification results for an imbalanced dataset, because it avoids using true positive, which tends to be extremely large in an imbalanced dataset.

$$Precision = \frac{TP}{TP+FP} \tag{16}$$

$$Recall = \frac{TP}{TP+FN} \tag{17}$$

$$F = \left(1+\beta^2\right) \times \frac{Precision \times Recall}{\beta^2 \times Precision + Recall} \tag{18}$$

where β is a weight value, which is used to determine either precision or recall more important. $\beta \in (0, +\infty)$ is used to decide the relative importance of recall over precision. When $\beta = 1$, two metrics are of same importance. $\beta > 1$ means that precision is more important [39].

The meaning of the precision is to compare the correctly classified positive instances to all the instances that have been classified as positive. The recall is to compare the correctly classified positive instances to the instances that actual label is positive. Another alternative name for recall is true positive rate. Ideally, a classifier with both high recall and high precision means its superb performance of correctly classifying a minority class. High recall represents that positive instances are mostly classified as positive and high precision illustrates the instances classified as positive mostly belong to the positive class. F-Measure contains the trade-off between precision and recall.

D. Statistical Comparison for Classifiers

Friedman's test [40] is a non-parametric statistical test, which is similar to the repeated-measures ANOVA. In this paper, we use it to evaluate the significance of the proposed algorithm with others and rank the algorithms according to the mean ranked performances of each algorithm based on each dataset separately. The null-hypothesis assumes that all the algorithms are equivalent. However, only the overall statistically significant difference cannot identify which algorithm particularly differs from other algorithms.

Wilcoxon signed rank test [41] and Holm test [42] are used for pairwise comparisons, when a null-hypothesis is rejected. It analyzes the significant differences on the different combinations of algorithms. Bergmann-Hommel test [43] can be applied to adjusted *p*-values for Wilcoxon test. According to the statistical results from Holm and Wilcoxon tests with adjusted *p*-values by using Bergmann-Hommel test, we can clearly show which algorithm significantly achieves the best performance among all tested algorithms.

E. Results and Their Analysis

1) Case Study 1: Statlog Dataset

In this section, the results of comparing GI-FS$^{\rho}$, GI-FS$^{1.5\rho}$, GI-FS, Chi2 and F-statistic feature selection methods are given and discussed. This dataset contains 6435 instances with 36 features. 6-fold cross validation is applied in the experiment as well.

In Fig. 3, we compare the tendency of AUC among the five different feature selection methods. It shows that GI-FS$^{1.5\rho}$ achieves the best performance among all five methods. Also, GI-FS$^{\rho}$, GI-FS and Chi2 need almost the same number of features to reach the maximum performance of AUC. It means that when the number of selected features is larger than a certain number, there is no difference to use any feature selection method among GI-FS^{w}, GI-FS and Chi2 based on AUC. If a small subset of features, e.g., the number is fewer than 5, are selected, GI-FSX is a better method than others.

As shown in Fig. 4, we compare the tendency of F-measure among the five different feature selection approaches based on 170 sequential trees. The number of features is selected from 3 to 36. 36 is the total number of features in the Statlog dataset. We conclude that our proposed methods (GI-FS^{w}) have the better F-measure than Chi2 and F-statistic. Then we compare the performance of F-measure among our proposed methods. It shows that GI-FS$^{1.5\rho}$ achieves the better result than GI-FS$^{\rho}$ and GI-FS where ρ is imbalance ratio.

2) Case Study 2: Letter Recognition Dataset

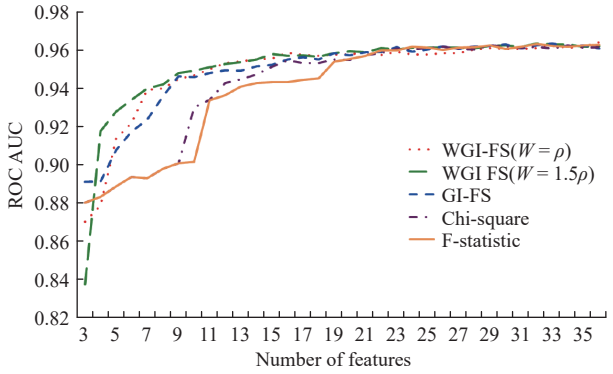


Fig. 3. Performance of five feature selection methods in terms of ROC AUC for Statlog dataset (color printing is desired for clarity).

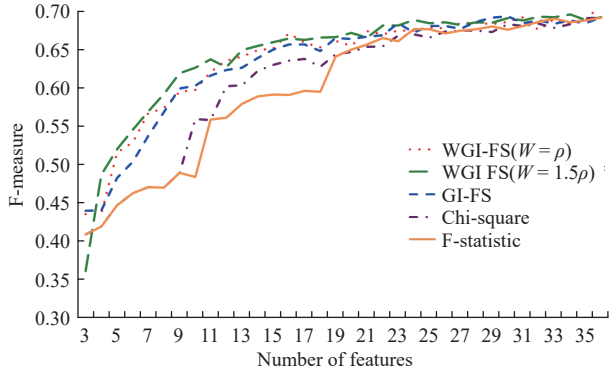


Fig. 4. Performance of five feature selection methods in terms of F-measure for Statlog dataset (color printing is desired for clarity).

In this section, we show the AUC and F-measure results of Chi2, F-statistic, GI-FS$^{1.5\rho}$, GI-FS$^{\rho}$ and GI-FS. The Friedman's test is a Chi2 test with 4 degrees of freedom. Table V presents Friedman's test results for AUC for 26 binary classification problems. 20%, 40%, 60% and 80% of all features are selected, respectively, in this experiment.

In Table VI when increasing the feature size to 20% of all features, Friedman's test results are $\chi^2 = 31.746$ and $p = 2.156 \times 10^{-6}$. The null-hypothesis is rejected because p-values are less than 0.05, which means that a significant difference exists among five feature selection methods when 20% of all features are selected. There is no significant difference, when increasing the feature's size from 40% to 80% of all features. The mean ranking results for all feature selection methods are shown in Table VII. The better performance a feature selection method achieves, the smaller its mean rank is. However, mean rank values make sense only when 20% of all features are selected, because no significant difference exists when the feature size is more than that. F-statistic contains the lowest MR, when 20% of all features are selected. However, it is easy to observe that even though only 20% of all features are used to classify, AUC performance is excellent. When increasing the feature's size to 60% or 80% of all features, we obtain the highest AUC, which is close to 1. This suggests that using AUC for this imbalance dataset is not an ideal metric. The detailed AUC results for the 26 binary classification problems are presented in Appendix.

TABLE V
FRIEDMAN'S TEST FOR AUC

	20%	40%	60%	80%
Chi2	31.746	0.853	7.961	0.085
p-value	2.156×10^{-6}	0.931	0.093	0.999

TABLE VI
FRIEDMAN'S TEST FOR F-MEASURE

	20%	40%	60%	80%
Chi2	33.362	0.915	17.423	14.177
p-value	1.007×10^{-6}	0.922	0.002	0.006

TABLE VII
MEAN RANK FOR AUC RESULTS THROUGH FRIEDMAN'S TEST

ζ (%)	Chi2	F-statistic	GI-FS$^{1.5\rho}$	GI-FS$^{\rho}$	GI-FS
20	2.17	**1.94**	3.42	3.76	3.69
40	3.13	3.11	2.85	2.85	3.06
60	3.46	3.46	2.60	2.60	2.88
80	3	3.03	2.92	3.02	3.02

Table VI shows Friedman's test results for F-measure for 26 binary classifications. According to the result, F-measure presents a very different picture in comparison with AUC. When 20%, 60% and 80% of features are selected, p-value equals 1.007×10^{-6}, 0.002 and 0.006, respectively. These p-values are less than the 5% critical difference level, which means the overall statistically significant difference exists. Table VIII shows the rankings in terms of F-measure results through Friedman's test based on selected features of different sizes.

According to mean rank values in Table VIII, F-statistic achieves the lowest mean rank value in Friedman's test when only 20% of features are selected. A comparison result for F-statistic with other methods through a Holm test is presented in Table IX. If p-value ls less than α/i that corresponds in the same row, the null hypothesis is rejected, which means a significant difference exists. Based on Table IX, only the Chi2's p-value is larger than α/i. Therefore, a Wilcoxon test is used to compare F-statistic and Chi2, and then a Bergmann-Hommel test is used for the adjusted p-value (Table X). It has shown no significant difference between Chi2 and F-statistic with an adjusted p-value of 0.280 (> 0.05).

When the selected feature's percentage is 40%, a p-value of 0.922 is obtained through Friedman's test among five feature selection methods. It shows that all the methods are not significantly different when $\zeta = 40\%$.

When 60% of all features are selected, p-value equals 0.002 (< 0.05) by using Friedman's test in Table VI. At the level, GI-FS$^{1.5\rho}$ obtains the best mean rank from Table VIII. So, a comparison result among GI-FS$^{1.5\rho}$ with other methods by using Holm test is presented in Table XI. GI-FS$^{1.5\rho}$ has significantly better performance than Chi2 and F-statistic methods. For GI-FS$^{\rho}$ and GI-FS methods, no significant

TABLE VIII
MEAN RANK FOR F-MEASURE RESULTS THROUGH FRIEDMAN'S TEST

ζ (%)	Chi2	F-statistic	GI-FS$^{1.5\rho}$	GI-FS$^{\rho}$	GI-FS
20%	2.15	**1.96**	3.25	3.73	3.90
40%	2.98	3.04	**2.84**	3.23	2.90
60%	3.73	3.5	**2.1**	3.09	2.46
80%	3.38	3.63	**2.25**	3.19	2.53

TABLE IX
HOLM TEST FOR F-STATISTIC WHEN USING 20% OF ALL FEATURES

Algorithm	p-value	α/i	Hyp.
GI-FS	9.461×10^{-6}	0.0125	Rejected
GI-FS$^{\rho}$	5.473×10^{-5}	0.01667	Rejected
GI-FS$^{1.5\rho}$	3.302×10^{-3}	0.025	Rejected
Chi2	0.6610	0.05	Accepted

TABLE X
WILCOXON TEST TO COMPARE F-MEASURE WITH CHI2

Comparison	p-value	Adj. p-value	Hyp. (α = 0.05)
Chi2 vs. F-statistic	0.227	0.280	Accepted

difference is detected according to Table XI. A comparison result through Wilcoxon test for GI-FS$^{1.5\rho}$ with GI-FS$^{\rho}$ and GI-FS is shown in Table XII. The adjusted p-values for GI-FS$^{1.5\rho}$ vs. GI-FS$^{\rho}$ and GI-FS$^{1.5\rho}$ vs. GI-FS are less than 0.05 with values of 0.038 and 0.047, respectively. The main conclusion for the ζ = 60% case is that GI-FS$^{1.5\rho}$ achieves significantly better performance than the rest of methods.

The p-value for all feature selection methods through Friedman's test is 0.006 < 0.05, thus signifying differences among the algorithms when 80% of all features are selected. According to the mean ranks in Table VIII, GI-FS$^{1.5\rho}$ has the best result. A Holm test is used to compare it with other methods. The result is shown in Table XIII, indicating that it has a significantly better performance than F-statistic and Chi2. It is hard to detect the difference between GI-FS$^{1.5\rho}$ and GI-FS$^{\rho}$ or GI-FS. Therefore, a pairwise comparison result is shown in Table XIV by using Wilcoxon test. It concludes GI-FS$^{1.5\rho}$ is better than GI-FS$^{\rho}$ with the adjusted p-value being 0.026.

To conclude, Chi2 and F-statistic achieve the best performance when only 20% features are selected. As ζ increases to 40%, five feature selection methods contain the similar performance. When ζ = 60%, GI-FS$^{1.5\rho}$ achieves better performance than other methods. When ζ = 80%, GI-FS$^{1.5\rho}$ has significantly better performance than F-statistic, Chi2 and GI-FS$^{\rho}$, but it remains statistically similar to GI-FS. According to the results, if only a small number of features are selected, it is better to use Chi2 and F-statistic as feature selection methods. when ζ > 20%, GI-FS$^{1.5\rho}$ is a best feature selection method that can achieve the superb performance. The detailed F-measure results are shown in Appendix.

TABLE XI
HOLM TEST FOR GI-FS$^{1.5\rho}$ WHEN USING 60% OF ALL FEATURES

Algorithm	p-value	α/i	Hyp.
Chi2	0.001	0.0125	Rejected
F-statistic	0.007	0.01667	Rejected
GI-FS$^{\rho}$	0.072	0.025	Accepted
GI-FS	0.095	0.05	Accepted

TABLE XII
WILCOXON TEST TO COMPARE GI-FS$^{1.5\rho}$ WITH GI-FS$^{1\rho}$ AND GI-FS

Comparison	p-value	Adj. p-value	Hyp. (α = 0.05)
GI-FS$^{1.5\rho}$ vs. GI-FS$^{\rho}$	0.019	0.038	Rejected
GI-FS$^{1.5\rho}$ vs. GI-FS	0.033	0.047	Rejected

TABLE XIII
HOLM TEST FOR GI-FS$^{1.5\rho}$ WHEN USING 80% OF ALL FEATURES

Algorithm	p-value	α/i	Hyp.
F-statistic	0.002	0.0125	Rejected
Chi2	0.010	0.01667	Rejected
GI-FS$^{\rho}$	0.032	0.025	Accepted
GI-FS	0.511	0.05	Accepted

TABLE XIV
WILCOXON TEST TO COMPARE GI-FS$^{1.5\rho}$ WITH GI-FS$^{1\rho}$ AND GI-FS

Comparison	p-value	Adj. p-value	Hyp. (α=0.05)
GI-FS$^{1.5\rho}$ vs. GI-FS$^{\rho}$	0.0064	0.026	Rejected
GI-FS$^{1.5\rho}$ vs. GI-FS	0.1402	0.2804	Accepted

V. CONCLUSION AND FUTURE WORK

This work proposes a feature selection method based on our newly proposed weighted Gini index, called GI-FS^{w}. It can be used to deal with the imbalanced classification problems, which are commonly encountered in real-world datasets. Its novelty lies in the use of imbalance ratio dependent weight to revise the original Gini index.

There are some limitations in our methods. In a decision tree algorithm, a splitting node is easily changed. Also, it is hard to distinguish which feature should be first selected when multiple features achieve the same frequency. This work has just explored the cases where the weight of GI-FS^{w} equals imbalanced ratio or 1.5 times of that. Will the other weights perform better? Determining the optimal weight in GI-FS^{w} remains unsolved. In the future work, besides the issue mentioned above, we should detect and remove the noise before the next stage of classifier training. The impact of varying imbalance ratios on the proposed method's performance should be further examined together with its application to other datasets [44]–[47].

APPENDIX

The detailed AUC results for the 26 binary classification problems are shown in Table XV.

TABLE XV
AUC RESULTS FOR LETTER DATASET

ROC AUC		20%	40%	60%	80%	100%			20%	40%	60%	80%	100%
	Chi2	0.993	**0.999**	**1.000**	**1.000**	1.000		Chi2	0.985	**0.997**	0.998	0.999	1.000
	F-statistic	0.993	**0.999**	**1.000**	**1.000**	1.000		F-statistic	0.978	**0.997**	0.998	0.999	1.000
A	GI-FS$^{1.5\rho}$	0.993	**0.999**	**1.000**	**1.000**	1.000	**J**	GI-FS$^{1.5\rho}$	0.984	0.996	**0.999**	0.999	1.000
	GI-FS$^{\rho}$	**0.987**	**0.999**	**1.000**	**1.000**	1.000		GI-FS$^{\rho}$	0.956	**0.997**	**0.999**	**1.000**	1.000
	GI-FS	0.984	**0.999**	**1.000**	**1.000**	1.000		GI-FS	**0.987**	0.995	**0.999**	0.999	1.000
	Chi2	**0.970**	0.989	**0.997**	**0.999**	0.999		Chi2	0.967	0.989	0.995	0.998	0.999
	F-statistic	**0.970**	**0.990**	**0.997**	**0.999**	0.999		F-statistic	0.967	0.991	0.995	0.998	0.999
B	GI-FS$^{1.5\rho}$	0.955	**0.990**	**0.997**	0.998	0.999	**K**	GI-FS$^{1.5\rho}$	0.968	**0.996**	**0.998**	**0.999**	0.999
	GI-FS$^{\rho}$	0.946	0.989	0.996	0.997	0.999		GI-FS$^{\rho}$	0.971	0.994	**0.998**	0.998	0.999
	GI-FS	0.950	0.985	**0.997**	0.998	0.999		GI-FS	**0.978**	0.995	**0.998**	**0.999**	0.999
	Chi2	0.967	**0.996**	**0.999**	**1.000**	1.000		Chi2	**0.987**	0.997	0.998	0.999	1.000
	F-statistic	**0.988**	**0.996**	**0.999**	**1.000**	1.000		F-statistic	**0.987**	**0.998**	0.998	0.999	1.000
C	GI-FS$^{1.5\rho}$	0.980	**0.996**	**0.999**	0.999	1.000	**L**	GI-FS$^{1.5\rho}$	0.979	0.996	**0.999**	**1.000**	1.000
	GI-FS$^{\rho}$	0.965	**0.996**	**0.999**	**1.000**	1.000		GI-FS$^{\rho}$	0.975	**0.998**	**0.999**	0.999	1.000
	GI-FS	0.980	**0.996**	**0.999**	0.999	1.000		GI-FS	0.974	0.996	**0.999**	**1.000**	1.000
	Chi2	**0.969**	0.991	**0.998**	**0.999**	0.999		Chi2	**0.997**	0.999	**1.000**	**1.000**	1.000
	F-statistic	**0.969**	0.991	**0.998**	**0.999**	0.999		F-statistic	**0.997**	0.999	0.999	**1.000**	1.000
D	GI-FS$^{1.5\rho}$	0.959	**0.993**	**0.998**	**0.999**	0.999	**M**	GI-FS$^{1.5\rho}$	0.994	0.998	**1.000**	**1.000**	1.000
	GI-FS$^{\rho}$	0.957	0.990	0.997	**0.999**	0.999		GI-FS$^{\rho}$	0.995	**1.000**	**1.000**	**1.000**	1.000
	GI-FS	0.961	0.992	**0.998**	0.998	0.999		GI-FS	**0.997**	**1.000**	**1.000**	**1.000**	1.000
	Chi2	**0.980**	0.993	0.997	**0.999**	0.999		Chi2	**0.990**	0.998	**0.999**	**0.999**	1.000
	F-statistic	0.978	0.993	0.997	**0.999**	0.999		F-statistic	**0.990**	0.998	**0.999**	**0.999**	1.000
E	GI-FS$^{1.5\rho}$	0.969	**0.994**	**0.998**	**0.999**	0.999	**N**	GI-FS$^{1.5\rho}$	0.987	0.998	**0.999**	**0.999**	1.000
	GI-FS$^{\rho}$	0.968	0.993	0.997	0.998	0.999		GI-FS$^{\rho}$	0.984	**0.999**	**0.999**	**0.999**	1.000
	GI-FS	0.965	0.992	0.997	0.998	0.999		GI-FS	0.979	0.998	**0.999**	**0.999**	1.000
	Chi2	**0.977**	0.991	**0.998**	**0.998**	0.998		Chi2	0.961	0.989	0.996	**0.999**	0.999
	F-statistic	**0.977**	0.991	**0.998**	**0.998**	0.998		F-statistic	0.962	0.989	0.996	**0.999**	0.999
F	GI-FS$^{1.5\rho}$	0.969	0.993	**0.998**	**0.998**	0.998	**O**	GI-FS$^{1.5\rho}$	**0.963**	**0.992**	**0.998**	**0.999**	0.999
	GI-FS$^{\rho}$	0.965	0.988	0.996	**0.998**	0.998		GI-FS$^{\rho}$	0.954	0.988	0.995	0.998	0.999
	GI-FS	0.953	**0.995**	**0.998**	**0.998**	0.998		GI-FS	0.955	0.991	**0.998**	**0.999**	0.999
	Chi2	**0.967**	0.991	**0.998**	0.998	0.999		Chi2	**0.987**	**0.997**	**0.998**	**0.999**	0.999
	F-statistic	**0.967**	0.990	**0.998**	0.998	0.999		F-statistic	**0.987**	**0.997**	**0.998**	**0.999**	0.999
G	GI-FS$^{1.5\rho}$	0.943	0.991	**0.998**	**0.999**	0.999	**P**	GI-FS$^{1.5\rho}$	0.963	0.994	**0.998**	**0.999**	0.999
	GI-FS$^{\rho}$	0.954	**0.992**	0.997	**0.999**	0.999		GI-FS$^{\rho}$	0.954	0.993	0.997	**0.999**	0.999
	GI-FS	0.951	0.991	0.997	**0.999**	0.999		GI-FS	0.955	0.993	**0.998**	**0.999**	0.999
	Chi2	0.822	0.962	0.987	0.989	0.996		Chi2	0.970	0.993	0.997	**0.999**	0.999
	F-statistic	0.833	0.962	0.986	0.988	0.996		F-statistic	**0.974**	0.993	0.997	**0.999**	0.999
H	GI-FS$^{1.5\rho}$	0.948	0.982	**0.995**	0.995	0.996	**Q**	GI-FS$^{1.5\rho}$	0.957	0.994	**0.999**	**0.999**	0.999
	GI-FS$^{\rho}$	0.955	**0.990**	0.993	0.996	0.996		GI-FS$^{\rho}$	0.964	**0.995**	0.998	**0.999**	0.999
	GI-FS	**0.967**	0.988	**0.995**	0.995	0.996		GI-FS	0.961	**0.995**	**0.999**	**0.999**	0.999
	Chi2	0.981	0.992	0.998	**0.999**	0.999		Chi2	**0.974**	**0.993**	**0.998**	**0.998**	0.999
	F-statistic	**0.986**	0.992	0.998	**0.999**	0.999		F-statistic	**0.974**	**0.993**	**0.998**	**0.998**	0.999
I	GI-FS$^{1.5\rho}$	0.975	0.996	**0.999**	0.998	0.999	**R**	GI-FS$^{1.5\rho}$	0.960	0.992	**0.998**	**0.998**	0.999
	GI-FS$^{\rho}$	0.978	0.994	**0.999**	**0.999**	0.999		GI-FS$^{\rho}$	0.964	0.992	**0.998**	**0.998**	0.999
	GI-FS	0.976	**0.997**	**0.999**	**0.999**	0.999		GI-FS	0.952	**0.993**	**0.998**	**0.998**	0.999

Table XV (Continued)

ROC AUC		20%	40%	60%	80%	100%			20%	40%	60%	80%	100%
	Chi2	0.965	0.992	0.996	**0.998**	0.999		Chi2	**0.997**	**0.999**	**1.000**	**1.000**	1.000
	F-statistic	0.965	0.992	0.996	**0.998**	0.999		F-statistic	**0.997**	**0.999**	**1.000**	**1.000**	1.000
S	GI-FS$^{1.5\rho}$	**0.966**	**0.994**	0.997	**0.998**	0.999	**W**	GI-FS$^{1.5\rho}$	0.993	**0.999**	**1.000**	**1.000**	1.000
	GI-FS$^{\rho}$	0.937	0.993	**0.998**	**0.998**	0.999		GI-FS$^{\rho}$	0.993	**0.999**	**1.000**	**1.000**	1.000
	GI-FS	0.960	0.989	0.997	**0.998**	0.999		GI-FS	0.989	**0.999**	**1.000**	**1.000**	1.000
	Chi2	**0.988**	**0.998**	**0.999**	**0.999**	1.000		Chi2	0.985	0.993	0.999	**1.000**	1.000
	F-statistic	**0.988**	**0.998**	**0.999**	**0.999**	1.000		F-statistic	0.987	0.993	0.999	**1.000**	1.000
T	GI-FS$^{1.5\rho}$	0.979	0.997	**0.999**	**0.999**	1.000	**X**	GI-FS$^{1.5\rho}$	0.975	**0.999**	**1.000**	**1.000**	1.000
	GI-FS$^{\rho}$	0.975	0.997	**0.999**	**0.999**	1.000		GI-FS$^{\rho}$	0.982	0.997	0.999	**1.000**	1.000
	GI-FS	0.974	0.997	**0.999**	**0.999**	1.000		GI-FS	**0.990**	0.997	0.999	**1.000**	1.000
	Chi2	0.982	**0.998**	**0.999**	**1.000**	1.000		Chi2	**0.983**	**0.999**	**1.000**	**1.000**	1.000
	F-statistic	0.982	0.997	**0.999**	**1.000**	1.000		F-statistic	**0.983**	0.998	**1.000**	**1.000**	1.000
U	GI-FS$^{1.5\rho}$	**0.985**	**0.998**	**0.999**	**1.000**	1.000	**Y**	GI-FS$^{1.5\rho}$	0.976	0.996	0.999	**1.000**	1.000
	GI-FS$^{\rho}$	0.983	**0.998**	**0.999**	**1.000**	1.000		GI-FS$^{\rho}$	0.978	0.996	0.999	**1.000**	1.000
	GI-FS	0.975	**0.998**	**0.999**	**1.000**	1.000		GI-FS	0.978	0.997	0.999	**1.000**	1.000
	Chi2	**0.991**	**0.997**	**0.999**	**0.999**	0.999		Chi2	0.992	0.999	**1.000**	**1.000**	1.000
	F-statistic	**0.991**	**0.997**	**0.999**	**0.999**	0.999		F-statistic	**0.994**	**1.000**	**1.000**	**1.000**	1.000
V	GI-FS$^{1.5\rho}$	0.966	0.996	**0.999**	**0.999**	0.999	**Z**	GI-FS$^{1.5\rho}$	0.980	0.999	**1.000**	**1.000**	1.000
	GI-FS$^{\rho}$	0.981	0.996	**0.999**	**0.999**	0.999		GI-FS$^{\rho}$	0.990	0.999	**1.000**	**1.000**	1.000
	GI-FS	0.975	0.996	**0.999**	**0.999**	0.999		GI-FS	0.981	0.999	**1.000**	**1.000**	1.000

The detailed F-measure results are shown in Table XVI.

TABLE XVI
F-MEASURE RESULTS FOR LETTER DATASET

F-measure		20%	40%	60%	80%	100%			20%	40%	60%	80%	100%
	Chi2	0.689	0.929	0.969	**0.988**	0.993		Chi2	**0.481**	0.654	0.790	0.858	0.890
	F-statistic	0.689	0.916	0.969	**0.988**	0.993		F-statistic	0.473	0.646	0.790	0.849	0.890
A	GI-FS$^{1.5\rho}$	**0.691**	**0.940**	**0.979**	0.987	0.993	**E**	GI-FS$^{1.5\rho}$	0.429	**0.690**	**0.840**	**0.878**	0.890
	GI-FS$^{\rho}$	0.636	0.919	0.973	0.982	0.993		GI-FS$^{\rho}$	0.414	0.667	0.801	0.868	0.890
	GI-FS	0.607	0.894	0.970	**0.988**	0.993		GI-FS	0.376	0.652	0.783	0.845	0.890
	Chi2	**0.439**	0.612	0.779	**0.863**	0.878		Chi2	0.480	0.680	0.859	**0.889**	0.896
	F-statistic	**0.439**	**0.618**	0.779	0.861	0.878		F-statistic	0.480	0.680	0.863	**0.889**	0.896
B	GI-FS$^{1.5\rho}$	0.361	0.615	0.787	0.852	0.878	**F**	GI-FS$^{1.5\rho}$	0.423	0.728	0.859	0.878	0.896
	GI-FS$^{\rho}$	0.337	0.617	0.769	0.823	0.878		GI-FS$^{\rho}$	0.410	0.649	0.809	0.862	0.896
	GI-FS	0.343	0.559	**0.792**	0.838	0.878		GI-FS	0.379	**0.741**	**0.868**	0.878	0.896
	Chi2	0.409	0.789	0.937	0.958	0.963		Chi2	**0.403**	0.669	0.857	0.886	0.908
	F-statistic	**0.560**	0.789	0.937	0.958	0.963		F-statistic	**0.403**	0.653	0.857	0.890	0.908
C	GI-FS$^{1.5\rho}$	0.514	**0.834**	**0.939**	**0.959**	0.963	**G**	GI-FS$^{1.5\rho}$	0.343	0.660	**0.861**	**0.894**	0.908
	GI-FS$^{\rho}$	0.439	0.780	0.914	0.953	0.963		GI-FS$^{\rho}$	0.364	0.670	0.832	0.887	0.908
	GI-FS	0.506	0.802	0.938	0.953	0.963		GI-FS	0.366	**0.679**	0.835	0.881	0.908
	Chi2	**0.440**	0.649	0.840	0.894	0.914		Chi2	0.167	0.422	0.601	0.655	0.788
	F-statistic	**0.440**	0.649	0.848	0.893	0.914		F-statistic	0.177	0.422	0.597	0.638	0.788
D	GI-FS$^{1.5\rho}$	0.379	**0.693**	**0.866**	0.897	0.914	**H**	GI-FS$^{1.5\rho}$	0.337	0.553	**0.757**	**0.768**	0.788
	GI-FS$^{\rho}$	0.373	0.613	0.818	**0.902**	0.914		GI-FS$^{\rho}$	0.350	**0.622**	0.711	0.763	0.788
	GI-FS	0.372	0.664	0.862	0.886	0.914		GI-FS	**0.407**	0.605	0.746	0.758	0.788

Table XVI (Continued)

F-measure		20%	40%	60%	80%	100%			20%	40%	60%	80%	100%
	Chi2	**0.558**	0.743	0.891	0.938	0.947		Chi2	**0.458**	**0.699**	**0.846**	**0.877**	0.879
	F-statistic	0.557	0.683	0.891	0.938	0.947		F-statistic	0.455	**0.699**	**0.846**	**0.873**	0.879
I	GI-FS$^{1.5\rho}$	0.492	0.821	0.925	0.944	0.947	**R**	GI-FS$^{1.5\rho}$	0.354	0.643	0.842	0.869	0.879
	GI-FS$^{\rho}$	0.484	0.811	**0.935**	**0.946**	0.947		GI-FS$^{\rho}$	0.371	0.647	0.825	0.857	0.879
	GI-FS	0.494	**0.846**	0.926	0.943	0.947		GI-FS	0.347	0.673	0.819	0.866	0.879
	Chi2	0.574	0.813	0.861	0.926	0.947		Chi2	0.408	0.683	0.798	0.839	0.910
	F-statistic	0.523	0.813	0.890	0.926	0.947		F-statistic	0.408	0.691	0.798	0.838	0.910
J	GI-FS$^{1.5\rho}$	0.550	0.819	0.905	**0.947**	0.947	**S**	GI-FS$^{1.5\rho}$	**0.418**	**0.743**	0.831	0.882	0.910
	GI-FS$^{\rho}$	0.388	**0.847**	0.922	0.943	0.947		GI-FS$^{\rho}$	0.345	0.732	**0.847**	0.883	0.910
	GI-FS	**0.597**	0.795	**0.925**	0.943	0.947		GI-FS	0.404	0.671	0.831	**0.885**	0.910
	Chi2	0.414	0.622	0.747	0.888	0.922		Chi2	**0.587**	0.829	0.915	0.936	0.954
	F-statistic	0.414	0.643	0.757	0.883	0.922		F-statistic	**0.587**	**0.836**	**0.926**	0.931	0.954
K	GI-FS$^{1.5\rho}$	0.393	**0.773**	**0.880**	0.902	0.922	**T**	GI-FS$^{1.5\rho}$	0.483	0.798	0.923	**0.944**	0.954
	GI-FS$^{\rho}$	0.416	0.730	0.868	0.908	0.922		GI-FS$^{\rho}$	0.489	0.799	0.906	0.935	0.954
	GI-FS	**0.457**	0.771	0.862	**0.910**	0.922		GI-FS	0.469	0.829	0.924	0.943	0.954
	Chi2	**0.647**	0.832	0.910	0.946	0.964		Chi2	0.537	0.862	0.929	0.951	0.963
	F-statistic	0.600	0.822	0.910	0.946	0.964		F-statistic	0.537	0.834	0.929	0.951	0.963
L	GI-FS$^{1.5\rho}$	0.552	0.803	0.931	**0.960**	0.964	**U**	GI-FS$^{1.5\rho}$	**0.550**	0.863	**0.948**	0.960	0.963
	GI-FS$^{\rho}$	0.525	**0.863**	**0.944**	0.957	0.964		GI-FS$^{\rho}$	0.540	0.828	0.928	0.953	0.963
	GI-FS	0.489	0.839	0.939	0.957	0.964		GI-FS	0.482	**0.882**	0.939	**0.962**	0.963
	Chi2	**0.781**	0.900	0.951	0.973	0.981		Chi2	**0.675**	**0.826**	0.896	0.930	0.943
	F-statistic	**0.781**	0.900	0.937	0.973	0.981		F-statistic	**0.675**	**0.826**	0.896	0.929	0.943
M	GI-FS$^{1.5\rho}$	0.707	0.844	0.935	**0.978**	0.981	**V**	GI-FS$^{1.5\rho}$	0.523	0.788	0.904	0.933	0.943
	GI-FS$^{\rho}$	0.732	**0.938**	**0.972**	**0.978**	0.981		GI-FS$^{\rho}$	0.527	0.772	**0.917**	**0.944**	0.943
	GI-FS	0.760	0.937	**0.972**	**0.978**	0.981		GI-FS	0.511	0.760	0.914	0.937	0.943
	Chi2	**0.603**	0.828	0.902	0.940	0.956		Chi2	0.793	**0.927**	0.958	0.972	0.970
	F-statistic	**0.603**	0.842	0.915	0.940	0.956		F-statistic	**0.801**	**0.927**	0.956	0.972	0.970
N	GI-FS$^{1.5\rho}$	0.567	0.826	0.928	0.949	0.956	**W**	GI-FS$^{1.5\rho}$	0.657	0.900	**0.964**	0.973	0.970
	GI-FS$^{\rho}$	0.517	**0.863**	**0.933**	**0.950**	0.956		GI-FS$^{\rho}$	0.710	0.888	**0.964**	0.970	0.970
	GI-FS	0.493	0.841	0.928	**0.950**	0.956		GI-FS	0.585	0.920	0.959	**0.974**	0.970
	Chi2	0.378	0.624	0.798	0.878	0.906		Chi2	0.530	0.658	0.888	0.942	0.959
	F-statistic	0.393	0.622	0.798	0.876	0.906		F-statistic	0.556	0.658	0.888	0.942	0.959
O	GI-FS$^{1.5\rho}$	**0.415**	**0.698**	**0.863**	**0.889**	0.906	**X**	GI-FS$^{1.5\rho}$	0.450	**0.848**	**0.933**	0.941	0.959
	GI-FS$^{\rho}$	0.361	0.582	0.755	0.817	0.906		GI-FS$^{\rho}$	0.507	0.804	0.923	0.945	0.959
	GI-FS	0.366	0.675	0.826	0.888	0.906		GI-FS	**0.595**	0.840	0.924	**0.946**	0.959
	Chi2	**0.572**	**0.797**	**0.913**	0.926	0.941		Chi2	**0.520**	**0.875**	0.936	0.958	0.959
	F-statistic	**0.572**	**0.797**	**0.913**	**0.932**	0.941		F-statistic	**0.520**	0.871	**0.940**	0.958	0.959
P	GI-FS$^{1.5\rho}$	0.433	0.763	0.897	0.927	0.941	**Y**	GI-FS$^{1.5\rho}$	0.518	0.801	0.919	0.956	0.959
	GI-FS$^{\rho}$	0.388	0.739	0.887	0.925	0.941		GI-FS$^{\rho}$	0.482	0.795	0.921	0.955	0.959
	GI-FS	0.383	0.741	0.893	0.926	0.941		GI-FS	0.500	0.811	0.931	**0.959**	0.959
	Chi2	0.386	**0.758**	0.852	0.942	0.946		Chi2	0.604	0.897	0.946	0.961	0.973
	F-statistic	**0.466**	**0.758**	0.876	0.942	0.946		F-statistic	**0.663**	**0.910**	0.943	0.961	0.973
Q	GI-FS$^{1.5\rho}$	0.374	0.724	0.906	**0.944**	0.946	**Z**	GI-FS$^{1.5\rho}$	0.523	0.870	0.941	0.963	0.973
	GI-FS$^{\rho}$	0.408	0.740	0.889	0.919	0.946		GI-FS$^{\rho}$	0.593	0.878	0.946	0.964	0.973
	GI-FS	0.374	0.741	**0.921**	0.943	0.946		GI-FS	0.509	0.861	**0.949**	**0.967**	0.973

REFERENCES

[1] F. Wang, T. Xu, T. Tang, M. C. Zhou, and H. Wang, "Bilevel feature extraction-based text mining for fault diagnosis of railway systems," *IEEE Trans. Intelligent Transportation Systems*, vol. 18, no. 1, pp. 49–58, Jan. 2017.

[2] D. Ramyachitra and P. Manikandan, "Imbalanced dataset classification and solutions: a review," *Inter. J. Computing and Business Research* (*IJCBR*), vol. 5, no. 4, Jul. 2014.

[3] E. Ramentol, Y. Caballero, R. Bello, and F. Herrera, "SMOTE-RSB*: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory," *Knowledge and Information Syst.*, vol. 33, no. 2, pp. 245–265, Nov. 2012.

[4] Q. Kang, X. Chen, S. Li, and M. C. Zhou, "A noise-filtered undersampling scheme for imbalanced classification," *IEEE Trans. Cybernetics*, vol. 47, no. 12, pp. 4263–4274, Dec. 2018.

[5] B. Krawczyk, M. Woźniak, and G. Schaefer, "Cost-sensitive decision tree ensembles for effective imbalanced classification," *Applied Soft Computing*, vol. 14, pp. 554–562, Jan. 2014.

[6] V. Lopez, S. del Rio, J. Manuel Benitez, and F. Herrera, "On the use of MapReduce to build linguistic fuzzy rule based classification systems for big data," in *Proc. IEEE Int. Conf. Fuzzy Syst.*. pp. 1905−1912, IEEE, Jul. 2014.

[7] Z. L. Cai and W. Zhu, "Feature selection for multi-label classification using neighborhood preservation," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 1, pp. 320–330, Jan. 2018.

[8] C. Jian, J. Gao, and Y. Ao, "A new sampling method for classifying imbalanced data based on support vector machine ensemble," *Neurocomputing*, vol. 193, pp. 115–122, 2016.

[9] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Machine Learning Research*, vol. 3, pp. 1157–1182, Mar. 2003.

[10] X. H. Yuan, L. B. Kong, D. C. Feng, and Z. C. Wei, "Automatic feature point detection and tracking of human actions in time-of-flight videos," *IEEE/CAA J. Autom Sinica*, vol. 4, no. 4, pp. 677–685, Oct. 2017.

[11] J. Wang, L. Qiao, Y. Ye, and Y. Chen, "Fractional envelope analysis for rolling element bearing weak fault feature extraction," *IEEE/CAA J. Autom. Sinica*, vol. 4, no. 2, pp. 353–360, 2017.

[12] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Editorial: special issue on learning from imbalanced data sets," *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 1–6, 2004.

[13] A. K. Uysal and S. Gunal, "A novel probabilistic feature selection method for text classification," *Knowledge-Based Systems*, vol. 36, pp. 226–235, 2012.

[14] L. Yu and H. Liu, "Feature selection for high-dimensional data: a fast correlation-based filter solution," in *Proc. Int. Conf. Machine Learning*, vol. 3, pp. 856−863, 2003.

[15] V. Bolón-Canedo, N. Sánchez-Marono, A. Alonso-Betanzos, J. Manuel Benítez, and F. Herrera, "A review of microarray datasets and applied feature selection methods," *Information Sciences*, vol. 282, pp. 111–135, 2014.

[16] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 41, no. 1, pp. 16–28, 2014.

[17] H. Liu and H. Motoda, "Feature selection for knowledge discovery and data mining," *Springer Science & Business Media*, vol. 454, 2012.

[18] S. Shilaskar and A. Ghatol, "Feature selection for medical diagnosis: Evaluation for cardiovascular diseases," *Expert Syst. with Applications*, vol. 40, no. 10, pp. 4146–4153, 2013.

[19] I. A. Gheyas and L. S. Smith, "Feature subset selection in large dimensionality domains," *Pattern Recognition*, vol. 43, no. 1, pp. 5–13, 2010.

[20] S. Maldonado and R. Weber, "A wrapper method for feature selection using support vector machines," *Information Sciences*, vol. 179, no. 13, pp. 2208–2217, 2009.

[21] Y. Zhu, J. Liang, J. Chen, and M. Zhong, "An improved NSGA-III algorithm for feature selection used in intrusion detection," *J. Knowledge-Based Syst.*, vol. 116, pp. 74–85, Jan. 2017.

[22] A. Moayedikia, K. L. Ong, Y. L. Boo, W. G. Yeoh, and R. Jensen, "Feature selection for high dimensional imbalanced class data using harmony search," *J. Engineering Applications of Artificial Intelligence*, vol. 57, pp. 38–49, Jan. 2017.

[23] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Machine Learning Research*, vol. 3, pp. 1157–1182, Mar. 2003.

[24] S. Maldonado and J. Lopez, "Dealing with high-dimensional class-imbalanced datasets: embedded feature selection for SVM classification," *J. Applied Soft Computing*, vol. 67, pp. 94–105, Jun. 2018.

[25] C. Apté, F. Damerau, and M. S. Weiss, "Automated learning of decision rules for text categorization," *ACM Trans. Information Syst.*, vol. 12, no. 3, pp. 233–251, 1994.

[26] G. Forman, "An extensive empirical study of feature selection metrics for text classification," *J. Machine Learning Research*, vol. 3, pp. 1289–1305, Mar. 2003.

[27] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri, "Kno. your neighbors: Web spam detection using the web topology," in *Proc. the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 423−430, Jul. 2007.

[28] H. Koh, W. C. Tan, and G. C. Peng, "Credit scoring using data mining techniques," *Singapore Management Review*, vol. 26, no. 2, pp. 252004.

[29] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[30] J. R. Quinlan, "Constructing decision tree," C4, 5, pp. 17–26, 1993.

[31] X. Chen, M. Wang, and H. Zhang, "The use of classification trees for bioinformatics," *Wiley Interdisciplinary Reviews*: *Data Mining and Knowledge Discovery*, vol. 1, pp. 55–63, 2011.

[32] L. Breiman, "Classification and regression trees," *Routledge*, 2017.

[33] H. Y. Liu, M. C. Zhou, X.S. Lu, and C. Yao, "Weighted Gini index feature selection method for imbalanced data," in *Proc. 15th IEEE International Conference on Networking, Sensing and Control (ICNSC)*, pp. 1−6, Mar. 2018.

[34] H. Y. Liu and M. C. Zhou, "Decision tree rule-based feature selection for large-scale imbalanced data," in *Proc. 26th IEEE Wireless and Optical Communication Conf. (WOCC)*, pp. 1−6, IEEE, Apr. 2017.

[35] T. Q. Chen and T. He, "Xgboost: extreme gradient boosting," *R Package Version 0.4−2*, 2015.

[36] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

[37] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Editorial: special issue on learning from imbalanced data sets," *ACM SIGKDD Explorations Newsletter*, vol. 1, pp. 1–6, 2004.

[38] D. D. Lewis, and A. G. William. "A sequential algorithm for training text classifiers," in *Proc. 17th Annu. Int. ACM SIGIR Conf on Research and Development in Information Retrieval*, Springer-Verlag New York, Inc., pp. 3−12, 1994.

[39] C. J. Van Rijsbergen. *Information Retrieval* (2nd ed.). Butterworth-Heinemann, Newton, MA, USA, 1979.

[40] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *The Annu. of Mathematical Statistics*, no. 1, pp. 86–92, 1940.

[41] R. F. Woolson, "Wilcoxon signed-rank test," *Wiley Encyclopedia of Clinical Trials*, pp. 1–3, 2007.

[42] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Machine Learning Research*, vol. 7, pp. 1–30, Jan. 2006.

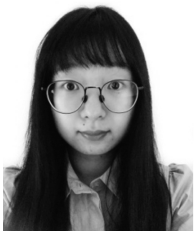
[43] S. Garcia and H. Francisco, "An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons," *J. Machine Learning Research*, vol. 9, pp. 2677–2694, Dec. 2008.

[44] P. Zhang, S. Shu, and M. C. Zhou, "An Online Fault Detection Method based on SVM-Grid for Cloud Computing Systems," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 2, pp. 445–456, Mar. 2018.

[45] J. Cheng, M. Chen, M. Zhou, S. Gao, C. Liu, and C. Liu, "Overlapping Community Change Point Detection in an Evolving Network," *IEEE Trans. Big Data*, DOI: 10.1109/TBDATA.2018.2880780, Nov. 2018.

[46] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and J. Wang, "Dendritic neuron model with effective learning algorithms for classification, approximation and prediction," *IEEE Trans-Neural Networks and Learning Syst.*, DOI: 10.1109/TNNLS.2018.2846646, 2018.

[47] Q. Kang, L. Shi, M. C. Zhou, X. Wang, Q. Wu, and Z. Wei, "A Distance-based Weighted Undersampling Scheme for Support Vector Machines and Its Application to Imbalanced Classification," *IEEE Trans. Neural Networks and Learning Syst.*, vol. 29, no. 9, pp. 4152–4165, Sep. 2018.



Haoyue Liu (S'17) received the B.S. degree from Kunming University of Science and Technology, Kunming, China, in 2014, and the M.S. degree from the New Jersey Institute of Technology, Newark, NJ, USA in 2016, where she is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering. Her current research interests include machine learning, natural language processing, sentiment analysis, and big data analytics.

MengChu Zhou (S'88−M'90−SM'93−F'03) received the B.S. degree in control engineering from Nanjing University of Science and Technology, Nanjing, China in 1983, M.S. degree in automatic control from Beijing Institute of Technology, Beijing, China in 1986, and Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, NY in 1990. He joined New Jersey Institute of Technology (NJIT), Newark, NJ in 1990, and is now a Distinguished Professor of electrical and computer engineering. His research interests include Petri nets, intelligent automation, internet of things, big data, web services, and intelligent transportation. He has over 800 publications including 12 books, 460+ journal papers (360+ in IEEE transactions), 12 patents and 29 book-chapters. He was invited to lecture in Australia, Canada, China, France, Germany, Italy, Japan, Korea, Mexico, Qatar, Saudi Arabia, Singapore, and US and served as a plenary/keynote speaker for many conferences. He is the founding Editor of *IEEE Press Book Series on Systems Science and Engineering* and Editor-in-Chief of *IEEE/CAA Journal of Automatica Sinica*. He served as Associate Editor of *IEEE Transactions on Robotics and Automation*, *IEEE Transactions on Automation Science and Engineering*, *IEEE Transactions on Systems*, *Man and Cybernetics*: *Systems*, and *IEEE Transactions on Industrial Informatics*, and Editor of *IEEE Transactions on Automation Science and Engineering*. He served as a Guest-Editor for many journals including *IEEE Internet of Things Journal*, *IEEE Transactions on Industrial Electronics*, and *IEEE Transactions on Semiconductor Manufacturing*. He is also Associate Editor of *IEEE Transactions on Intelligent Transportation Systems*, *IEEE Internet of Things Journal*, and *Frontiers of Information Technology & Electronic Engineering*. He was General Chair of IEEE Conf. on Automation Science and Engineering, Washington D.C., August 23–26, 2008, General Co-Chair of 2003 IEEE International Conference on System, Man and Cybernetics (SMC), Washington DC, October 5–8, 2003, Founding General Co-Chair of 2004 IEEE Int. Conf. on Networking, Sensing and Control, Taipei, March 21–23, 2004, and General Chair of 2006 IEEE Int. Conf. on Networking, Sensing and Control, Ft. Lauderdale, Florida, USA. April 23–25, 2006. He was Program Chair of 2010 IEEE International Conference on Mechatronics and Automation, August 4–7, 2010, Xi'an, China, 1998 and 2001 IEEE International Conference on SMC and 1997 IEEE International Conference on Emerging Technologies and Factory Automation. He organized and chaired over 100 technical sessions and served on program committees for many conferences. Dr. Zhou has led or participated in over 50 research and education projects with total budget over $12M, funded by National Science Foundation, Department of Defense, National Institute of Standards and Technology (NIST), New Jersey Science and Technology Commission, and Industry, USA. He is a recipient of Humboldt Research Award for US Senior Scientists from Alexander von Humboldt Foundation, Franklin V. Taylor Memorial Award and the Norbert Wiener Award from *IEEE Systems*, *Man and Cybernetics Society* for which he serves as VP for Conferences and Meetings. He is a life member of Chinese Association for Science and Technology-USA and served as its President in 1999. He is a Fellow of International Federation of Automatic Control (IFAC), American Association for the Advancement of Science (AAAS) and Chinese Association of Automation (CAA).

Qing Liu is an Assistant Professor in the Department of Electrical and Computer Engineering at New Jersey Institute of Technology. Prior to that, he was a staff scientist at Science Data Group, Oak Ridge National Laboratory for 7 years. He received the Ph.D. degree in computer engineering from the University of New Mexico in 2008, M.S. and B.S. degrees from Nanjing University of Posts and Telecom, China, in 2004 and 2001, respectively. He is a member of Association for Computing Machinery (ACM). His research interests include high-performance computing, data management, and computer systems.