Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 35 (2008) 1041–1052
Expert Systems with Applications
www.elsevier.com/locate/eswa

Classification of weld flaws with imbalanced class data

T. Warren Liao *

Industrial Engineering Department, Louisiana State University, Baton Rouge, LA 70808, United States

Abstract

This paper presents research results of our investigation of the imbalanced data problem in the classification of different types of weld flaws, a multi-class classification problem. The one-against-all scheme is adopted to carry out multi-class classification and three algorithms including minimum distance, nearest neighbors, and fuzzy nearest neighbors are employed as the classifiers. The effectiveness of 22 data preprocessing methods for dealing with imbalanced data is evaluated in terms of eight evaluation criteria to determine whether any method would emerge to dominate the others. The test results indicate that: (1) nearest neighbor classifiers outperform the minimum distance classifier; (2) some data preprocessing methods do not improve any criterion and they vary from one classifier to another; (3) the combination of using the AHC_KM data preprocessing method with the 1-NN classifier is the best because they together produce the best performance in six of eight evaluation criteria; and (4) the most difficult weld flaw type to recognize is crack.
© 2007 Elsevier Ltd. All rights reserved.

Keywords: Multi-class classification; One-against-all; Weld flaws; Imbalanced data; Minimum distance classifier; *K* nearest neighbors; Fuzzy *k*-nearest neighbors

1. Introduction

Welding is a major joining process used to fabricate many engineered artifacts and structures such as cars, ships, space shuttles, off-shore drilling plate-forms, and pipe lines. Flaws introduced into the material as a result of welding are detrimental to the integrity of the fabricated artifacts/structures. Commonly seen weld flaws include lack of fusion, lack of penetration, gas holes, porosities, cracks, inclusions, etc. Of course, some flaw types might appear more often than others for a particular welding process. To maintain the desirable level of structural integrity, welds must be inspected according to the established standard. The results of weld inspection also provide useful information for identifying the potential problem in the fabrication process and for improving the welding operations. In the current industrial practice, weld inspection is often carried out by certified inspectors.

For a welding process to be acceptable it should produce far more good welds than flawed welds and it must be qualified to meet some standard. In fact, most industries today are striving for six-sigma quality, which is known to mean 3.4 ppm (parts per million) defective. There are thus relatively fewer instances of flawed welds than instances of good welds. Moreover, different types of weld flaws might not be equally distributed. The distribution of weld flaws might actually change from a material problem to a workmanship problem, and also from a welding process to another. This characteristic that the number of examples of one flaw type is much higher than the others is known as the class imbalance problem or the problem of imbalanced data. The class imbalance problem is thus intrinsic in the domain of weld inspection as in many other domains such as fraud detection, oil spill detection, and text classification that have been studied. It has been reported in machine learning research that when learning from imbalanced data sets, machine learning algorithms tend to produce high predictive accuracy over the majority class, but poor predictive accuracy over the minority class. Learning from imbalanced data thus becomes an important subfield in machine learning research.

Efforts have been made in the past to develop computer-aided weld inspection systems to improve the objectivity

* Tel.: +1 225 578 5365; fax: +1 225 578 5109.
E-mail address: ieliao@lsu.edu

0957-4174/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2007.08.044

and productivity of weld inspection operations. Liao (2003) decomposed the development of such a system into three stages and grouped past work into three categories accordingly as follows:

- Segmentation of welds from background: Felisberto, Lopes, Centeno, and Arruda (2006), Liao and Ni (1996), Liao and Tang (1997), Liao, Li, and Li (2000) and Liao (2004).
- Detection of weld flaws in each weld: Carrasco and Mery (2004), Daum, Rose, Heidt, and Builtjes (1987), Gayer, Saya, and Shiloh (1990), Hyatt, Kechter, and Nagashima (1996), Kaftandjian, Dupuis, Babot, and Zhu (2003), Liao and Li (1998), Liao, Li, and Li (1999), Murakami (1990) and Wang and Wong (2005).
- Classification of types of weld flaws: Aoki and Suga (1999), Kato et al. (1992), Liao (2003), Murakami (1990) and Wang and Liao (2002).

To the best of our knowledge none of the previous research in developing a computer-aided weld inspection system, including ours, has explicitly addressed the issue of imbalanced data. The reason is that this research on imbalanced data gets started only recently. It did not catch the attention of researchers working in developing a computer-aided weld inspection system until now.

The work reported in this paper continues our previous research effort towards developing a computer-assisted weld inspection system. Specifically, our objective is to investigate the imbalanced data problem in the classification of different types of weld flaws, which is a multi-class classification problem. The effectiveness of several methods for dealing with imbalanced data is evaluated to determine whether any method will emerge to dominate the others. The performance of three classifiers, which include minimum distance, nearest neighbors, and fuzzy nearest neighbors are also compared.

The remainder of the paper is organized as follows. Section 2 introduces the evaluation criteria. Section 3 presents our research methodology including data preprocessing methods to be investigated for their effectiveness in balancing imbalanced data, multi-class classification strategies, classification algorithms, and test method. Section 4 briefly describes the weld flaws data used in this study. Section 5 presents the test results, analyses and discussion. Related work is reviewed in Section 6, followed by the conclusions.

2. Evaluation criteria

Overall accuracy has been shown inadequate for measuring the performance of a classifier when the data is imbalanced. Therefore, much initial effort in machine learning research with imbalance data was devoted to the development of new evaluation criteria, primarily for two-class classification problems.

The confusion matrix is often used to represent the results of a two-class classification problem. Consider the confusion matrix shown in Table 1, in which TN, FN, TP, and FP denote the number of true negative examples, the number of false negative examples, the number of true positive examples, and the number of false positive examples, respectively. The sum of the two rows give the total number of examples in each class, which is $n^- = FP + TN$ and $n^+ = TP + FN$, respectively. Accordingly, the traditional overall accuracy is computed as $\frac{TN+TP}{TN+FP+FN+TP}$. The accuracy on positive examples, also called sensitivity, is $\frac{TP}{FN+TP}$ whereas the accuracy on negative examples, also called specificity, is $\frac{TN}{TN+FP}$. To maximize the accuracy on each of the two classes while keeping these accuracies balanced, Kubat and Matwin (1997) proposed to use the geometric mean of the two accuracies: $g = \sqrt{\frac{TN}{TN+FP} \cdot \frac{TP}{FN+TP}}$. Receiver operating characteristics (ROC) curves are often used to visualize the trade-off between true positive rates, TP/n^+, plotted on the y-axis and false positive rates, FP/n^-, plotted on the x-axis for a classification task. The point $(0,0)$ corresponds to the strategy of always predicting the negative (majority) class (in other words, never making a positive prediction) and the point $(1,1)$ to always predicting the positive class. The line $x = y$ represents the strategy of randomly guessing the class. The area under the curve $(0,0)$-(FPR, TPR)-$(1,1)$ can be computed as $(\text{TPR} - \text{FPR} + 1)/2$, where (FPR, TPR) is a particular classification result. Cohen, Hilario, Sax, Hogonnet, and Geissbuhler (2006) proposed the mean class weighted accuracy (CWA) for C-class classification that was defined as

Table 1
Confusion matrix

		Model predicted	
		Negative	Positive
Actual	Negative	TN	FP
	Positive	FN	TP

$$\text{CWA} = \sum_{i=1}^{C} w_i \text{accu}_i, \quad w_i \in [0,1] \quad \text{and} \quad \sum_{i=1}^{C} w_i = 1.$$

In binary classification, the above equation becomes

$$\text{CWA} = w \times \text{sensitivity} + (1 - w) \times \text{specificity}.$$

It is interesting to point out that when setting $w = 0.5$ the CWA in binary classification is identical to the area under the ROC curve for one-point classification result (FPR, TPR).

The information retrieval community prefers to work with precision and recall, which are computed as $\frac{TP}{FP+TP}$ and $\frac{TP}{FN+TP}$, respectively. Sometimes the geometric mean of precision and recall is used. A more sophisticated criterion that considers both precision and recall is the F-measure that is defined as

$$F\text{-measure} = \frac{(1+\beta^2) \cdot \text{recall} \cdot \text{precision}}{\beta^2 \cdot \text{recall} + \text{precision}}.$$

If precision and recall are equally important, then β is set to 1. The F-measure with $\beta = 1$ is also known as the F_1-measure in the literature.

The geometric mean of accuracies, F_1-measure, and area under the ROC curve (AUC) are three most commonly used criteria to measure the classifier performance on imbalanced data (see the Related Work reviewed in Section 6). In this study, eight evaluation criteria are used to evaluate the effectiveness of a data preprocessing method for dealing with the imbalanced data problem; they include area under the ROC curve, geometric mean of accuracies, F_1-measure, class weighted accuracy, sensitivity, specificity, precision, and overall accuracy with notations AUC, GM, F1, CWA, SENS, SPEC, PREC and ACCU, respectively.

3. Research methodology

The methodology used in this research has four major components: data preprocessing methods for dealing with imbalanced data, multi-class classification strategies, classification algorithms, and testing method. Each component is detailed in the following sections.

3.1. Data preprocessing methods

A data preprocessing method for dealing with imbalanced data falls into one of the following three categories: under-sampling the majority class, over-sampling the minority class, or hybrid of over-sampling and under-sampling. Many commonly used data preprocessing methods are implemented in this study to investigate their effectiveness in classifying welding flaws. We prefer over-sampling methods over under-sampling methods because only very limited amount of data is available. A brief description of each implemented method is given below.

3.1.1. Over-sampling methods

- *Random Over-sampling (Rand_Over)*: This method randomly select examples from the minority class with replacement until the number of selected examples plus the original examples of the minority class is identical to that of the majority class.
- *Synthetic Minority Over-sampling TEchnique (SMOTE)*: This heuristic, originally developed by Chawla, Bowyer, Hall, and Kegelmeyer (2002), generates synthetic minority examples to be added to the original set. For each minority example, its five nearest neighbors of the same class are found. Some of these nearest neighbors are randomly selected according to the over-sampling rate. A new synthetic example is generated along the line between the minority example and every one of its selected nearest neighbors.
- *Borderline-SMOTE1 (SMOTE1)*: This method modifies the original SMOTE by over-sampling only those minority class examples near the borderline. The detailed procedure is not given here but can be found in Han, Wang, and Mao (2005).
- *Agglomerative Hierarchical Clustering Based (AHC)*: This method was first used by Cohen et al. (2006). It involves three major steps: (1) using an agglomerative hierarchical clustering algorithm such as single linkage to form a dendogram, (2) gathering clusters from all levels of the dendogram and computing the cluster centroids as synthetic examples, and (3) concatenating centroids with the original minority class examples. Though not clear whether it is in the original procedure, we remove the redundancies of centroids that might be found in more than one layer in our implementation.

3.1.2. Under-sampling methods

- *Random Under-sampling (RU)*: This is a non-heuristic method that randomly select examples from the majority class for removal without replacement until the remaining number of examples is same as that of the minority class.
- *Bootstrap Under-sampling (BU)*: This method is similar to random under-sampling, but with replacement. An example thus can be selected more than once.
- *Condensed Nearest Neighbor (CNN)*: This method first randomly draw one example from the majority class to be combined with all examples from the minority class to form a training set S, then use a 1-NN over S to classify the examples in the training set and move every misclassified example from the training set to S (Hart, 1968).
- *Edited Nearest Neighbor (ENN)*: This method was originally proposed by Wilson (1972). It works by removing noisy examples from the original set. An example is deleted if it is incorrectly classified by its k-nearest neighbors ($k = 3$ in our implementation).
- *Tomek Links (Tomek)*: Given two examples $E_i = (x_i, y_i)$ and $E_j = (x_j, y_j)$ where $y_i \neq y_j$ and $d(E_i, E_j)$ being the distance between E_i and E_j. The (E_i, E_j) pair forms a Tomek link if there exists no example E_l such that $d(E_i, E_l) < d(E_i, E_j)$ or $d(E_j, E_l) < d(E_i, E_j)$ according to Tomek (1976). This method removes examples belonging to the majority class in each Tomek link found.
- *One-Sided Selection (OSS)*: This method first applies Tomek links then follows it with CNN (Kubat & Matwin, 1997).
- *Neighborhood Cleaning Rule (NCR)*: This method was originally proposed by Laurikkala (2001) and it employs the Wilsons Edited Nearest Neighbor Rule to remove selected majority class examples. For each example $E_i = (x_i, y_i)$ in the training set, its three nearest neighbors are found. If E_i belongs to the majority class and the classification given by its three nearest neighbors is the minority class, then remove E_i. If E_i belongs to the

minority class and its three nearest neighbors misclassify it, then remove the nearest neighbors that belong to the majority class.

- *One-Sided Selection Reversed (OSSR)*: This method first applies the *CNN* rule then follows it with Tomek links (Batista, Monard, & Bazzan, 2004; Batista, Prati, & Monard, 2004). Since it uses the two methods in reversed order as OSS, we thus name it OSS reversed.
- *K-Means Based (KM)*: This method, first used by Cohen et al. (2006), applies the *k*-means clustering algorithm to group the majority class into sub-clusters and the resulting prototypes of sub-clusters are used as synthetic cases to replace all the original majority class examples.
- *Fuzzy-C-Means Based (FCM)*: This method is similar to *KM*, except that the fuzzy *c*-means algorithm is used instead of the *k*-means clustering algorithm.

3.1.3. Hybrid methods

- *SMOTE-RU Hybrid*: After over-sampling the minority class with SMOTE, the majority class is under-sampled by randomly removing samples from the majority class until the minority class becomes some specified percentage of the majority class (Chawla et al., 2002).
- *SMOTE-Tomek Hybrid*: Following over-sampling with SMOTE, examples from both majority and minority classes that form Tomek links are removed as a data cleaning method (Batista, Monard et al., 2004; Batista, Prati et al., 2004).
- *SMOTE-ENN Hybrid*: This method first uses SMOTE to generate synthetic minority examples and then applies ENN to remove each majority/minority class example from the data set that does not have at least two of its three nearest neighbors from the same class (Batista, Monard et al., 2004; Batista, Prati et al., 2004).
- *AHC-KM Hybrid*: This method combines AHC-based over-sampling and KM-based under-sampling (Cohen et al., 2006).
- *SMOTE-Bootstrap Hybrid (SMOTE_BU)*: This method was first used by Liu, Chawla, Harper, Shrilberg, and Stolcke (2006) and it involves first using SMOTE for over-sampling the minority class and then Bootstrap sampling the majority class so that both classes have the same or similar number of examples.

In principle, other under-sampling methods such as CNN, OSS, and OSSR can also be used to hybridize with SMOTE even though such hybridization has not been done before. In addition, SMOTE can be replaced with SMOTE1. To the best of our knowledge, no one has investigated the effectiveness of such hybrids as well. We thus decide to include them in this study. A total of 23 methods are tested including one without applying any preprocessing method, named *IM*, and 22 data preprocessing methods. To give a better idea of how our study differ from the previous studies, a more detailed review of previous research using data preprocessing methods for dealing with imbalanced data is given in Section 6 – Related Work.

3.2. Multi-class classification strategies

Since more than two types of welding flaws are often involved in a welding process, classification of welding flaws is naturally a multi-class pattern recognition problem. Due to the lack of adequate evaluation criteria for multi-class classification with imbalanced data, most researchers focused only on two-class classification problems. For the subject application, we choose to use a multi-class classification strategy that is based on two-class classification. Two possible schemes are one-against-all (OAA) and one-against-one (OAO). Let the number of classes be C, the OAA scheme essentially implements C numbers of two-class classifiers whereas the OAO scheme requires a system of $C(C-1)/2$ two-class classifiers. Since half of the classes of our data set have 10 or fewer examples (see Section 4 for more detail), it would be difficult to learn a model for any paired two-class classification involving those classes. Therefore, the OAA scheme is adopted for this study. Rifkin and Klautau (2004) argued that a simple OAA scheme is as accurate as any other approach, assuming that the underlying binary classifiers are well-tuned regularized classifiers such as support vector machine. It, however, should be noted that the OAA scheme unavoidably exacerbate the data imbalance problem, especially when C is large.

When it is desired to classify a new example with the OAA scheme, each one of the C classifiers are run to determine whether the new example belongs to the class that the classifier is trained for. Let $f_i = 1$ if classifier i determines that the new example belongs to the class while $f_i = 0$ if not. There are three possible patterns of output from the C numbers of binary classifiers. The first output pattern is the ideal one with $f_i = 1$ and $f_j = 0$ for $\forall j \neq i$. In this case, the example clearly belongs to class i. The second output pattern is that $f_i = 0$ for $\forall i$. In this case, none of the C classifiers claims the example be its class. The third output pattern is that more than one classifier output "1". In this case there is a "tie" among those classes. To force a decision in the last two cases, one could pick the class that its corresponding binary classifier produces the highest output value of belongingness, among all the candidates.

The problem with this testing method as described above is that it does not provide the information needed to compute most of those criteria mentioned in Section 2, except overall accuracy. Therefore, we test each class data separately by the corresponding binary classifier and take the average of all class results as the performance of the overall OAA scheme for the multi-class classification problem at hand, classification of weld flaws.

3.3. Classification algorithms

Various classification algorithms have been used in the study of classifying imbalance data. They include decision

tree methods such as C4.5, C5.0, CART, neural networks such as multi-layer perceptron (MLP), radial basis function (RBF), and fuzzy ARTMAP, linear discriminant analysis, naïve Bayes, support vector machine, and neighbor neighbors methods. Due to the relative high number of features and fewer examples for some class of weld flaws, classifiers that require more examples to learn such as linear discriminant analysis, decision trees, and neural networks are not applicable for this study. Therefore, we choose to use minimum distance classifier, nearest neighbor, and fuzzy nearest neighbor algorithm.

3.4. Testing method

The k-fold cross-validation testing method is used. The maximal number of folds is determined by the smallest class. Since the smallest class of our data has only four examples, the maximal number of folds is 4 to ensure that there is at least one test example for the smallest class in each fold. The k-fold cross-validation data is generated by stratified sampling to ensure that the proportion of class data is kept. The k-fold cross-validation testing procedure is repeated 25 times by randomizing the entire data set before generating the k-fold cross-validation data for each combination of data preprocessing method and classifier. For each test, the average of each evaluation criteria is computed based on the k-fold test results.

4. Description of weld flaws data

The data set used in this study has 147 examples with each example having 12 numeric input attributes and one pattern attribute indicating the weld flaw type. Six types of weld flaws are contained in the data set; they include cracks, gas holes, hydrogen inclusions, lack of fusion, lack of penetration, and porosities with 37, 10, 28, 8, 4, and 60 examples, respectively.

Each weld flaw is first extracted from a radiographic image and 12 features describing its shape, size, location, and intensity information are then extracted. The 12 features are distance from the center of weld, mean radius, standard deviation of radius, circularity, compactness, major axis, width, length, elongation, Heywood diameter, mean intensity, and standard deviation of intensity. More detailed definition about each of the above 12 features can be found in Wang and Liao (2002). Extracting discriminant features is always important to the success of a pattern recognition task. The set of features extracted is definitely imperfect. Nevertheless, the feature set is good enough judging from the relative high accuracy achievable, as presented in the next section.

5. Results, analyses and discussion

The test results obtained in this study are organized by the classifier used in the following sections. For each data preprocessing method, the average of 25 runs and the associated confidence interval are computed for each one of the eight evaluation criteria. Two analyses are then carried out. The first analysis serves to determine whether each data preprocessing method improves the performance of each criterion, compared to using the imbalance data set. Specifically, we compute the 95% one-sided confidence interval of the difference between the mean of a data preprocessing method and the mean of no data preprocessing method, and conclude that both methods are statistically indifferent if zero is below the upper confidence bound. The second analysis intends to determine the statistically indifferent best data preprocessing methods. For the second analysis, the data preprocessing method that produces the highest average is first identified and then statistical testing is carried out to determine whether each other data preprocessing method statistically differ from the best identified. Specifically, we compute the 95% confidence interval of the difference between two means and conclude that both methods are statistically indifferent if zero is within the confidence interval.

5.1. Minimum distance classifier

Table 2 gives the statistical test results for determining whether a data preprocessing method is effective in improving the performance of each evaluation criterion, when the minimum distance classifier is used. The results indicate that out of 22 preprocessing methods, seven do not

Table 2
Statistical test results of whether a data preprocessing method improves the performance (1 for not and 0 for yes) when using the minimum distance classifier

Classifier	Preprocessing method	auc	gm	f1	cwa	sens	spec	prec	accu
md	im	1	1	1	1	1	1	1	1
md	ahc	0	0	0	0	0	1	1	1
md	ahc_fcm	0	0	0	0	0	0	1	1
md	ahc_km	0	0	0	0	0	1	1	1
md	rand_over	1	1	1	1	1	1	1	1
md	smote	1	1	1	1	1	1	1	1
md	smote_bu	0	0	0	0	0	0	0	0
md	smote_cnn	0	0	0	0	0	0	0	0
md	smote_enn	1	1	1	1	1	1	1	1
md	smote_ncr	1	1	1	1	1	1	1	1
md	smote_oss	0	0	0	0	0	1	0	1
md	smote_ru	0	0	0	0	0	0	0	0
md	smote_tomek	1	1	1	1	1	1	1	1
md	smote1	1	1	1	1	1	1	1	1
md	smote1_bu	1	1	1	1	0	0	0	0
md	smote1_cnn	0	0	0	0	0	0	0	0
md	smote1_enn	0	0	0	0	0	0	0	0
md	smote1_ncr	1	1	1	1	1	1	1	1
md	smote1_oss	0	0	1	0	0	0	0	0
md	smote1_ossr	0	0	1	0	1	0	1	0
md	smote1_ru	1	1	1	1	1	0	1	0
md	smote1_tomek	0	0	0	0	0	0	0	1

Table 3
Best data preprocessing methods identified when using the minimum distance classifier

Evaluation criterion	Best data preprocessing method	Statistically indifferent best data preprocessing methods
Area under the ROC curve (auc)	ahc	ahc_fcm, ahc_km
Geometric mean of accuracies (gm)	ahc	ahc_fcm, ahc_km
F_1 measure (f1)	ahc	ahc_fcm, ahc_km
Class weighted accuracy (cwa)	ahc	ahc_fcm, ahc_km
Sensitivity (sens)	ahc_fcm	ahc, ahc_km, smote_bu, smote_ru, smote1_bu
Specificity (spec)	smote_cnn	smote1_cnn, smote1_enn, smote1_oss, smote1_tomek
Precision (prec)		
	smote1_oss	smote_cnn, smote_oss, smote1_cnn, smote1_enn, smote1_tomek
Overall accuracy (accu)	smote1_oss	ahc, smote, smote_cnn, smote_oss, smote1_cnn, smote1_enn, smote1_tomek

Table 4
Statistical test results of whether a data preprocessing method improves the performance (1 for not and 0 for yes) when using the 1-NN classifier

Classifier	Preprocessing methods	auc	gm	f1	cwa	sens	spec	prec	accu
1nn	im	1	1	1	1	1	1	1	1
1nn	ahc	1	1	1	1	0	1	1	1
1nn	ahc_fcm	0	0	0	0	0	0	0	0
1nn	ahc_km	0	0	1	0	0	0	0	1
1nn	rand_over	1	1	1	1	1	1	1	1
1nn	smote	1	1	1	1	0	1	1	1
1nn	smote_bu	0	0	0	0	0	0	0	0
1nn	smote_cnn	1	0	0	1	0	0	0	0
1nn	smote_enn	1	1	1	1	1	0	0	0
1nn	smote_ncr	1	1	1	1	0	0	0	0
1nn	smote_oss	1	0	0	1	0	0	0	0
1nn	smote_ossr	0	0	0	0	0	0	0	0
1nn	smote_ru	0	0	0	0	0	0	0	0
1nn	smote_tomek	0	0	1	0	0	0	1	0
1nn	smote1	1	0	1	1	1	1	1	1
1nn	smote1_bu	1	1	0	1	0	0	0	0
1nn	smote1_cnn	1	0	1	1	0	0	0	0
1nn	smote1_enn	0	0	0	0	0	1	1	0
1nn	smote1_ncr	1	1	1	1	1	1	1	1
1nn	smote1_oss	1	1	0	1	0	0	0	0
1nn	smote1_ossr	1	1	0	1	0	0	0	0
1nn	smote1_ru	1	1	0	1	0	0	0	0
1nn	smote1_tomek	0	0	0	0	0	1	1	1

improve any criterion, nine improve at least one criterion but not all criteria, and five methods improve all of the criteria. Note that the preprocessing method SMOTE_OSSR, short for the SMOTE and OSSR hybrid, did not work on our data; hence no result for it was given in the table. Table 3 summarizes the best and statistically indifferent best preprocessing methods by each evaluation criterion. The results indicate that the best preprocessing methods improve all criteria (because IM is not one of the statistically indifferent best in any criterion) but none emerges as the dominant method (because there is more than one statistically indifferent best). It will become clear in Section 5.4 that the best results obtained by using the minimum distance classifier are not comparable to those obtained by nearest neighbor classifiers.

5.2. K-nearest neighbors

Table 4 gives the statistical test results for determining whether a data preprocessing method is effective in improving the performance of each evaluation criterion, when 1-NN is used as the classifier. The results indicate that out of 22 preprocessing methods, two do not improve any criterion, eighteen improve some criteria, and four methods improve all of the criteria. Table 5 summarizes the best and statistically indifferent best preprocessing methods by each evaluation criterion. The results indicate that the best preprocessing methods improve on four of the eight criteria and none emerges as the dominant method. The four criteria that no improvement was made are the F_1-measure, specificity, precision, and overall accuracy; for them without using any preprocessing method (**im** in bold) is also one of the statistically indifferent best methods.

5.3. Fuzzy k-nearest neighbors

Table 6 gives the statistical test results for determining whether a data preprocessing method is effective in improving the performance of each evaluation criterion, when fuzzy 1-NN is used as the classifier. The results indicate that out of 22 preprocessing methods, four do not improve any criterion, sixteen improve some criteria, and no method improves all of the criteria. Note that two preprocessing methods, SMOTE_OSSR and SMOTE1_OSSR, did not work on our data; hence they did not produce any result. Table 7 summarizes the best and statistically indifferent best preprocessing methods by each evaluation criterion. The results indicate that the best preprocessing methods improve on five of the eight criteria and none emerges as the dominant method. The three criteria without improvement made are the F_1-measure, specificity and overall accuracy; for them without using any preprocessing method is either the best or one of the statistically indifferent best methods (**im** in bold).

5.4. Putting all three classifiers together

Table 8 summarizes the best and statistically indifferent best preprocessing methods among all results obtained by all three classifiers for each evaluation criterion. The

Table 5
Best data preprocessing methods identified when 1-NN is used as the classifier

Evaluation criterion	Best data pre-processing method	Statistically indifferent best data preprocessing methods
Area under the ROC curve	ahc_km	ahc, smote_tomek, smote1_cnn
Geometric mean of accuracies	ahc_km	ahc, smote_tomek, smote1_cnn, smote1_oss
F_1 measure	smote1	**im**, ahc, ahc_km, rand_over, smote_enn, smote_ncr, smote_tomek, smote1_ncr
Class weighted accuracy	ahc_km	ahc, smote_tomek, smote1_cnn
Sensitivity	smote_oss	ahc_fcm, ahc_km, smote_bu, smote_cnn, smote_ossr, smote_ru, smote1_bu, smote1_cnn, smote1_oss, smote1_ossr, smote1_ru
Specificity	smote1_enn	**im**, ahc, rand_over, smote, smote1
Precision	smote1_enn	**im**, ahc, smote1, smote1_ncr, smote1_tomek
Overall accuracy	smote1	**im**, ahc, ahc_km, rand_over, smote, smote1_ncr, smote1_tomek

Table 6
Statistical test results of whether a data preprocessing method improves the performance (1 for not and 0 for yes) when using the fuzzy 1-NN classifier

Classifier	Preprocessing methods	auc	gm	f1	cwa	sens	spec	prec	accu
f1nn	im	1	1	1	1	1	1	1	1
f1nn	ahc	0	0	1	0	0	1	1	1
f1nn	ahc_fcm	0	0	1	0	0	1	1	1
f1nn	ahc_km	0	0	1	0	0	1	1	1
f1nn	rand_over	1	1	1	1	1	1	1	1
f1nn	smote	0	0	1	0	0	1	1	1
f1nn	smote_bu	1	1	1	1	0	1	1	1
f1nn	smote_cnn	1	0	1	1	0	1	1	1
f1nn	smote_enn	0	0	1	0	0	1	1	1
f1nn	smote_ncr	1	1	1	1	0	1	1	1
f1nn	smote_oss	1	0	1	1	0	1	1	1
f1nn	smote_ru	1	1	1	1	0	1	1	1
f1nn	smote_tomek	0	0	1	0	0	1	1	1
f1nn	smote1	1	1	1	1	1	1	1	1
f1nn	smote1_bu	1	1	1	1	0	1	1	1
f1nn	smote1_cnn	1	1	1	1	0	1	1	1
f1nn	smote1_enn	1	1	1	1	1	1	0	1
f1nn	smote1_ncr	1	1	1	1	1	1	1	1
f1nn	smote1_oss	1	0	1	1	0	1	1	1
f1nn	smote1_ru	1	1	1	1	0	1	1	1
f1nn	smote1_tomek	1	1	1	1	1	1	1	1

results indicate that the best preprocessing methods improve on five of the eight criteria and none emerges as the dominant method. The three criteria without improvement made are the F_1-measure, specificity and overall accuracy, for either one of them both 1nn-im and f1nn-im (in bold in the table) are also the statistically indifferent best. A total of 29 classifier-preprocessing method combinations do not produce the best or statistically indifferent best for any criteria, which include all those using the minimum distance classifier (22 of them), one using 1-NN (1NN-SMOTE_NCR), and six using Fuzzy 1-NN. Therefore, the nearest neighbor algorithms clearly outperform the minimum distance classifier and 1-NN has a slight edge over fuzzy 1-NN.

Table 7
Best data preprocessing methods identified when fuzzy 1-NN is used as the classifier

Evaluation criterion	Best data pre-processing method	Statistically indifferent best data preprocessing methods
Area under the ROC curve	Ahc_km	smote_tomek
Geometric mean of accuracies	Ahc_km	smote_tomek
F_1 measure	Smote_tomek	**im**, ahc, rand_over, smote, smote_enn, smote1
Class weighted accuracy	Ahc_km	smote_tomek
Sensitivity	smote_bu	ahc_fcm, ahc_km, smote_oss, smote_ru, smote1_bu, smote1_ru
Specificity	smote1_enn	**im**, rand_over, smote1, smote1_tomek
Precision	smote1_enn	smote1_tomek
Overall accuracy	**Im**	ahc, rand_over, smote, smote1, smote1_tomek

To give some idea about the best performance attained in this study, the 95% confidence interval of the best combination of classifier and preprocessing method is given in the second column of Table 8 as well (inside the parenthesis) for each evaluation criterion. Note that the accuracy related criteria all have pretty high values: 0.937 ± 0.011 for geometric mean, 0.940 ± 0.011 for class weighted accuracy, 0.961 ± 0.018 for sensitivity, 0.948 ± 0.005 for specificity and 0.945 ± 0.007 for overall accuracy. Compared to the above criteria, precision of 0.860 ± 0.017 is not as good, which brings down the F_1-measure to 0.833 ± 0.017. The AUC value is identical to the CWA value because we set $w = 0.5$ in computing the class weighted average; this verifies what we discussed earlier in Section 2.

5.5. Discussion

Among all those combinations of classifier-preprocessing method tested in this study, 1NN-AHC_KM seems

Table 8
Best combinations of classifier and data preprocessing methods identified

Evaluation criterion	Best classifier-preprocessing method	Statistically indifferent best classifier-preprocessing methods
Area under the ROC curve	1nn-ahc_km (0.940 ± 0.011)	1nn-ahc, 1nn-smote_tomek, 1nn-smote1_cnn, f1nn-ahc_km
Geometric mean of accuracies	1nn-ahc_km (0.937 ± 0.011)	1nn-ahc, 1nn-smote_tomek, 1nn-smote1_cnn, 1nn-smote1_oss, f1nn-ahc_km
F_1 measure	f1nn-smote_tomek (0.833 ± 0.017)	**1nn-im**, 1nn-ahc, 1nn-ahc_km, 1nn-rand_over, 1nn-smote, 1nn-smote_enn, 1nn-smote_tomek, 1nn-smote1, 1nn-smote1_ncr, **f1nn-im**, f1nn-ahc, f1nn-rand_over, f1nn-smote, f1nn-smote_enn, f1nn-smote1
Class weighted accuracy	1nn-ahc_km (0.940 ± 0.011)	1nn-ahc, 1nn-smote_tomek, 1nn-smote1_cnn, f1nn-ahc_km
Sensitivity	1nn-smote_oss (0.961 ± 0.018)	1nn-ahc_fcm, 1nn-ahc_km, 1nn-smote_bu, 1nn-smote_cnn, 1nn-smote_ossr, 1nn-smote_ru, 1nn-smote1_bu, 1nn-smote1_cnn, 1nn-smote1_oss, 1nn-smote1_ossr, 1nn-smote1_ru, f1nn-ahc_fcm, f1nn-ahc_km, f1nn-smote_bu, f1nn-smote_oss, f1nn-smote1_bu, f1nn-smote1_ru
Specificity	f1nn-smote1_enn (0.948 ± 0.005)	**1nn-im**, 1nn-rand_over, 1nn-smote, 1nn-smote1, 1nn-smote1_enn, 1nn-smote1_tomek, **f1nn-im**, f1nn-rand_over, f1nn-smote1, f1nn-smote1_enn, f1nn-smote1_tomek
Precision	f1nn-smote1_enn (0.860 ± 0.017)	1nn-smote1, 1nn-smote1_enn, 1nn-smote1_tomek, f1nn-smote1_tomek
Overall accuracy	1nn-smote (0.945 ± 0.007)	**1nn-im**, 1nn-ahc, 1nn-ahc_km, 1nn-rand_over, 1nn-smote1, 1nn-smote1_ncr, 1nn-smote1_tomek, **f1nn-im**, f1nn-ahc, f1nn-rand_over, f1nn-smote, f1nn-smote1, f1nn-smote1_tomek

to be the best because it produces the highest numbers of best performances, six out of eight evaluation criteria. Assuming that all criteria are equally important, two combinations are tie for the second place: 1NN-AHC and 1NN-SMOTE1. The tie can be broken if one considers one criterion is weighted more than others. Eleven combinations are tie for the third place and so on. The rankings of all combinations of classifier-preprocessing methods that have at least one statistically indifferent best are given in Table 9. To give some idea about the performance difference between different classes (weld flaws), Table 10 gives the breakdown of the performance of 1NN-AHC_KM by class in terms of 95% confidence interval of the averages of four-fold cross-validation tests. In the table, the class that has the worst performance for each criterion is highlighted in bold. The results indicate that crack is most difficult to recognize among all six types of weld flaws, followed by gas hole and porosity.

Now that we have identified 1NN-AHC_KM be the best combination of classifier and preprocessing method, the next question naturally is whether using more than one nearest neighbor would further improve its performance. To study its effect, the number of nearest neighbors, k, was varied from 1 to 4. The reason that we cannot go higher is due to the limitation of the number of examples for the smallest class. Table 11 summarizes the results in terms of 95% confidence interval of the averages of four-fold cross-validation tests. The results indicate that as k increases no improvement can be made on the six criteria that 1NN-AHC_KM have the best performance (in bold). There is also no improvement on the specificity criterion as k increases. The only improvement is seen on the sensitivity criterion, for which $k = 2$ is better than $k = 1$.

Table 9
Sorted classifier-preprocessing method combinations by the number of best or statistically indifferent best performances

		auc	gm	f1	cwa	sens	Spec	prec	accu	total
1nn	ahc_km	1	1	1	1	0	0	1	1	6
1nn	ahc	1	1	1	0	0	0	1	1	5
1nn	smote1	0	1	0	0	1	1	1	1	5
1nn	im	0	1	0	0	1	0	1	1	4
1nn	rand_over	0	1	0	0	1	0	1	1	4
1nn	smote	0	1	0	0	1	0	1	1	4
1nn	smote_tomek	1	1	1	0	0	0	0	1	4
1nn	smote1_cnn	1	0	1	1	0	0	0	1	4
1nn	smote1_tomek	0	0	0	0	1	1	1	1	4
f1nn	im	0	1	0	0	1	0	1	1	4
f1nn	ahc_km	1	0	1	1	0	0	0	1	4
f1nn	rand_over	0	1	0	0	1	0	1	1	4
f1nn	smote1	0	1	0	0	1	0	1	1	4
f1nn	smote1_tomek	0	0	0	0	1	1	1	1	4
1nn	smote1_enn	0	0	0	0	1	1	0	1	3
1nn	smote1_ncr	0	1	0	0	0	0	1	1	3
1nn	smote1_oss	1	0	0	1	0	0	0	1	3
f1nn	ahc	0	1	0	0	0	0	1	1	3
f1nn	smote	0	1	0	0	0	0	1	1	3
f1nn	smote1_enn	0	0	0	0	1	1	0	1	3
1nn	ahc_fcm	0	0	0	1	0	0	0	1	2
1nn	smote_bu	0	0	0	1	0	0	0	1	2
1nn	smote_cnn	0	0	0	1	0	0	0	1	2
1nn	smote_enn	0	1	0	0	0	0	0	1	2
1nn	smote_oss	0	0	0	1	0	0	0	1	2
1nn	smote_ossr	0	0	0	1	0	0	0	1	2
1nn	smote_ru	0	0	0	1	0	0	0	1	2
1nn	smote1_bu	0	0	0	1	0	0	0	1	2
1nn	smote1_ossr	0	0	0	1	0	0	0	1	2
1nn	smote1_ru	0	0	0	1	0	0	0	1	2
f1nn	ahc_fcm	0	0	0	1	0	0	0	1	2
f1nn	smote_bu	0	0	0	1	0	0	0	1	2
f1nn	smote_enn	0	1	0	0	0	0	0	1	2
f1nn	smote_oss	0	0	0	1	0	0	0	1	2
f1nn	smote_tomek	0	1	0	0	0	0	0	1	2
f1nn	smote1_bu	0	0	0	1	0	0	0	1	2
f1nn	smote1_ru	0	0	0	1	0	0	0	1	2

Table 10
Break down by class (weld flaw) for the best combination, 1nn-ahc_km

Criterion	Class 1 (crack)	Class 2 (gas hole)	Class 3 (hydrogen inclusion)	Class 4 (lack of fusion)	Class 5 (lack of penetration)	Class 6 (porosity)
Auc	**0.869 ± 0.011**	0.926 ± 0.024	0.981 ± 0.011	0.963 ± 0.002	0.993 ± 0.004	0.906 ± 0.011
Gm	**0.867 ± 0.011**	0.918 ± 0.028	0.980 ± 0.010	0.962 ± 0.002	0.993 ± 0.004	0.902 ± 0.011
F1	0.776 ± 0.016	**0.684 ± 0.067**	0.956 ± 0.023	0.709 ± 0.020	0.892 ± 0.049	0.887 ± 0.011
CWA	**0.869 ± 0.011**	0.926 ± 0.024	0.981 ± 0.011	0.963 ± 0.002	0.993 ± 0.004	0.906 ± 0.011
Sens	**0.867 ± 0.018**	0.900 ± 0.043	0.979 ± 0.015	1.000 ± 0.000	1.000 ± 0.000	0.977 ± 0.015
Spec	0.870 ± 0.012	0.951 ± 0.006	0.983 ± 0.011	0.926 ± 0.004	0.986 ± 0.008	**0.836 ± 0.018**
Prec	0.714 ± 0.023	**0.605 ± 0.064**	0.939 ± 0.037	0.616 ± 0.034	0.860 ± 0.061	0.817 ± 0.016
Accu	**0.869 ± 0.010**	0.949 ± 0.008	0.982 ± 0.010	0.931 ± 0.004	0.986 ± 0.007	0.894 ± 0.011

Table 11
Effect of k on the best combination, 1nn-ahc_km

Criterion	$K = 1$	2	3	4
Auc	**0.940 ± 0.011**	0.906 ± 0.012	0.923 ± 0.012	0.897 ± 0.009
Gm	**0.937 ± 0.011**	0.900 ± 0.014	0.920 ± 0.013	0.891 ± 0.010
F1	**0.817 ± 0.027**	0.704 ± 0.026	0.764 ± 0.025	0.678 ± 0.029
CWA	**0.940 ± 0.011**	0.906 ± 0.012	0.923 ± 0.012	0.897 ± 0.009
Sens	0.954 ± 0.015	**0.976 ± 0.014**	0.951 ± 0.013	0.960 ± 0.013
Spec	**0.925 ± 0.009**	0.836 ± 0.018	0.894 ± 0.015	0.834 ± 0.015
Prec	**0.759 ± 0.004**	0.602 ± 0.037	0.686 ± 0.032	0.573 ± 0.038
Accu	**0.935 ± 0.008**	0.869 ± 0.014	0.909 ± 0.012	0.866 ± 0.013

6. Related work

Previous work in classifying imbalanced data can be roughly grouped into four categories: data preprocessing (or sampling), modification of standard classifier, one-class learning, feature selection, and ensemble approaches. To limit the scope, only work related to the first category is reviewed here in this section.

Focusing on two-class imbalanced data problems, Kubat and Matwin (1997) proposed a under-sampling method called one-sided selection (OSS), which is adapted from the technique of Tomek links by allowing only the removal of examples from the majority class while leaving the examples from the minority class untouched. The merits of OSS were evaluated based on k-fold cross-validations in terms of overall accuracy, accuracy of positive examples, accuracy of negative examples, and the geometric mean of the last two. Stratified sampling was carried out to ensure that each of the k subsets had the same proportion of positive and negative examples. 1-NN and C4.5 were selected as the classifiers. The test results suggested that OSS should be used only if one of the classes has prohibitively few examples. For the oil spill detection application, Kubat, Holte, and Matwin (1998) developed the SHRINK algorithm that is insensitive to imbalanced data. SHRINK was found to outperform 1-NN, but not C4.5 with OSS. Laurikkala (2001) proposed a new method, called neighborhood cleaning rule, for balancing imbalanced class distribution with data reduction. The new method was found to outperform simple random selection within classes and one-sided selection method in their experiments with 10 real-world data sets from UCI machine learning repository using the 3-NN and C4.5 classifiers. The 10-fold CV test results were measured by overall accuracies, true positive rates, true negative rates, and true positive rates for class of interest; and the statistical significance of performance differences was determined based on the two-tailed Wilcoxon signed ranks test due to small sample sizes (10 pairs in each comparison).

Chawla et al. (2002) proposed an over-sampling method called SMOTE, which involves creating synthetic minority class examples. Using C4.5 as the learner, they experimented nine different datasets with the 10-fold cross-validation test scheme and showed that a combination of SMOTE and under-sampling the majority class could achieve better classifier performance (in ROC space) than varying the ratios in Ripper or class priors in Naïve Bayes. Han et al. (2005) presented two new minority over-sampling methods, borderline-SMOTE1 and borderline-SMOTE2, in which only the minority examples near the borderline are over-sampled. Four over-sampling methods including SMOTE, random over-sampling, and the two new methods were applied to four different data sets. True positive rates and F-measure values computed for these methods are the average results of three independent 10-fold cross-validation experiments using C4.5 as the learner. As a whole, border-SMOTE1 was the best performer among all four tested.

Japkowicz and Stephen (2002) carried out a systematic study of the class imbalance problem. First, they tested the C5.0 decision tree induction system with a series of artificial concept-learning domains with various

combinations of concept complexity, training set size, and degree of imbalance. The results suggested that a huge class imbalance will not hinder classification of a domain whose concept is very easy to learn or the training set is large. Conversely, a small class imbalance could greatly harm a very small data set or one representing a very complex concept. In addition, they reported that as long as all the sub-clusters of a domain have a reasonable size, the class imbalance problem is of no consequence for C5.0; whereas if some sub-clusters are very small, then C5.0's performance will deteriorate. Secondly, they compared the performance of three methods for dealing with class imbalances: over-sampling, under-sampling, and cost-modifying, with C5.0 as the classifier. The cost-modifying method was found to be most effective among all. Thirdly, they investigated whether two other learning algorithms: multi-layer perceptron (MLP) and support vector machine (SVM) are affected by class imbalance. It was found that compared to C5.0, MLP was less sensitive and SVM was not sensitive at all. For each experiment, four types of results were reported: (1) the corrected results in which no matter what degree of class imbalance is present in the training set, the contribution of the false positive error rate is the same as that of the false negative one in the overall report; (2) the uncorrected results in which the reported error rate reflects the same imbalance as the one present in the training set; (3) the false positive error rate; and (4) the false negative error rate.

Barandela, Sánchez, García, and Rangel (2003) studied several selection algorithms for downsizing the training data (majority class alone or both majority and minority classes) and proposed a weighted distance function for internally biasing the discrimination procedure. Two selection algorithms are in the group of editing: the classical Wilson's editing rule, and the k-NCN (nearest centroid neighborhood) scheme. The third selection algorithm is the modified selective condensing. Two combinations of editing-condensing were also employed. The preprocessed training examples were classified with the NN rule, using both Euclidean and weighted distance measures. Four datasets taken from the UCI Repository were tested with five-fold cross-validation and the averaged results of the geometric mean were computed. Applying k-NCN on the majority class together with the use of weighted distance measure produced the best results. Murphey, Guo, and Feldkamp (2004) conducted a study on three neural network learners, multi-layered back-propagation, radial basis function, and fuzzy ARTMAP, to learn from unbalanced and noisy data using three different training methods: duplicating minority class examples, Snowball technique (Wang & Jean, 1993), and multidimensional Gaussian modeling of data noise. The last training method was proposed by the authors to generate new minority data examples near the classification boundary. The authors showed through experimental results that their noise modeling algorithm was effective in the training of both BP and fuzzy ARTMAP neural networks when the noise level is high on unbalanced data.

In their Editorial for the Special Issue on Learning from Imbalanced Data Sets, Chawla, Japlowicz, and Kotcz (2006) gave a short overview of the papers that were presented at the 2000 AAAI Conference and the 20003 ICML Conference and briefly described the papers contained in the Special Issue. Weiss (2004) presented an overview of the field of learning from imbalanced data. He discussed the role that rare classes and rare cases play in data mining, described the problems caused by these two forms of rarity, and the methods for addressing these problems. Using 13 UCI datasets and C4.5 as the learner, Batista, Prati et al. (2004) performed a broad experimental evaluation involving ten methods, three of them proposed by the authors, to cope with the class imbalance problem. The experimental results indicated that: (1) in general, over-sampling methods provide more accurate results than under-sampling methods considering the area under the ROC curve; (2) two of their proposed methods, SMOTE + Tomek and SMOTE + ENN, produced very good results for data sets with a small number of positive examples; and (3) decision trees induced from over-sampled data are usually more complex than the ones induced from original imbalanced data. Using experimental results on data taken from the SWISS-PROT database, Batista, Monard et al. (2004) showed that the symbolic classifiers induced by C4.5 with the balanced data sets using SMOTE + ENN, random over-sampling, and random under-sampling outperformed the ones induced using the original skewed data sets.

In an attempt to build more effective classifiers for the prosody model that is implemented as a CART decision tree classifier, Liu et al. (2006) investigated four different sampling approaches and a bagging scheme to cope with the imbalanced data distribution. The four sampling approaches are random down-sampling, over-sampling using replication, SMOTE, and ensemble down-sampling. The ensemble down-sampling approach first split the majority class into N subsets with each having roughly the same number of examples as the minority class, then use each of these subsets together with the minority class to train a classifier, and the results of N classifiers are averaged to obtain the final decision. Empirical evaluations in a pilot study showed that down-sampling the dataset works reasonably well, while requiring less training time. Both SMOTE and ensemble down-sampling outperformed the down-sampling approach when the prosody model used alone, but not when the prosody model is combined with the language model. Evaluation was also performed on two corpora that differ in speaking style: conversational phone speech and broadcast news. It was found that: (1) when the prosody model used alone, applying bagging on the original training set achieves significantly better results than ensemble bagging on both corpora; (2) the performance difference mentioned in (1) disappears when the prosody model is combined with the language model; and (3) there is a computational advantage of using down-sam-

pled training sets compared to using the original training set. To remedy class imbalance in monitoring and detection of nosocomial or hospital acquired infections (NIs), Cohen et al. (2006) explored a new re-sampling approach in which class-specific sub-clustering was used to generate synthetic cases by over-sampling of rare positives or/and under-sampling of the non-infected majority. They conducted stratified five-fold cross-validation experiments on a dataset of 683 patient records with each having 49 variables (75 infected cases out of a total of 683). Five learning algorithms, which include IB1, Naive Bayes, C4.5, AdaBoost, and SVM, were run first on the original class distribution, then on training data balanced via random sub-sampling, random over-sampling and different variants of their approach. Among all five algorithms, Naïve Bayes was found to have the highest sensitivity (87%) and class weighted accuracy (84%), when agglomerative hierarchical clustering (AHC)-based over-sampling is combined with *K*-means sub-sampling.

Huang, Hung, and Jiau (2006) first analyzed different classification algorithms that were employed to predict the creditworthiness of a bank's customers based on checking account information and then proposed a data cleaning strategy that uses a classifier as a filter that filters out specific instances in the training dataset. Various combinations of methods, such as C4.5, ID3, Naïve Bayes, and PRISM, were tested with either two classes: declined and good (including risky) or three classes: declined, risky, and good. Garcia and Cano (2006) proposed an evolutionary under-sampling (EUS) method to tackle the problem of imbalanced data. EUS codes chromosomes in binary with length equaling to the number of total training examples. An example is selected if its corresponding gene is coded in '1.' The fitness function is defined as

$$\text{Fitness}(S) = g - \left|1 - \frac{n_+}{n_-}\right| \cdot P,$$

where g is geometric mean of balanced accuracies produced by a subset of examples selected S, n_+ (n_-) is the number of positive (negative) examples selected from the minority (majority) class, and P is a penalty factor. Two versions of evolutionary operator: heterogeneous recombinations and cataclysmic mutation (CHC) and PBIL, denoted as CHC-US and PBIL-US, were applied. Seven data sets taken from the UCI Repository were tested with 1-NN using 10-fold cross-validation. It was found that: (1) CHC-US and PBIL-US had higher values of average geometric means than four prototype selection methods and six other under-sampling methods, and (2) EUS obtained better reduction than non-evolutionary under-sampling methods based on the Wilcoxon Signed Ranks Test.

Alejo, García, Sotoca, Mollineda, and Sánchez (2006) utilized the classical Wilson's editing rule to filter out noise or atypical patterns and to study its effect on the performance of two neural networks: RBF and MLP. Using six synthetic data sets generated with different levels of overlapping and three data sets taken from the UCI Repository, the classification performance of the two neural networks were evaluated in terms of the average values of the geometric mean. The application of the editing technique was found to improve the performance of the RBF neural network but not that of the MLP. Xie and Qiu (2007) showed theoretically and experimentally that imbalanced data had a negative effect on the performance of linear discriminant analysis (LDA) for binary classification. Ten data sets taken from UCI were tested using four-fold cross-validation and AUC was calculated for each test. Four sampling methods including random over-sampling, random under-sampling, Tomek links, and SMOTE were used to rebalance the original data sets. The experimental results indicated that LDA performed better with balanced data sets, especially those balanced with over-sampling methods.

7. Conclusions

This paper has presented the details of a study carried out to investigate the effectiveness of 22 data preprocessing methods for dealing with the imbalanced data problem inherent to the classification of six different types of weld flaws. The consideration of imbalanced data is new in classifying weld flaws. In addition, many of these data preprocessing methods are new and have not been used by previous researchers in any other applications.

The one-against-all scheme was used to perform multiclass classification using minimum distance, *k*-nearest neighbors, and fuzzy *k*-nearest neighbors as the classifiers. The effectiveness was measured using eight evaluation criteria developed primarily for binary classification. *K*-fold cross-validation data were repeatedly generated by stratified sampling for testing. Based on the test results and subsequent analyses, the following observations can be made:

(1) Nearest neighbor classifiers outperform the minimum distance classifier. When putting all the results obtained by all three classifiers together, none of the results obtained by the minimum distance classifier is in the list of statistically indifferent best in any one of the eight criteria.
(2) The number of data preprocessing methods that do not improve any criterion is 7, 2, and 4 when the classifier is minimum distance, *k*-nearest neighbors, and fuzzy *k*-nearest neighbors, respectively.
(3) The number of data preprocessing methods that improve all eight criteria is 5, 4, and 0 when the classifier is minimum distance, *k*-nearest neighbors, and fuzzy *k*-nearest neighbors, respectively.
(4) The combination of using the AHC_KM data preprocessing method with the 1-NN classifier is the best among all because they together produce the best performance in six of eight evaluation criteria.
(5) The most difficult weld flaw type to recognize is crack.

(6) 1-NN seems to be better than higher k values for most criteria, except the sensitivity criterion.

Possible topics for future study include:

(1) Developing evaluation criteria appropriate for multi-class classification with imbalanced data.
(2) Applying feature selection/weighting method alone or together with data preprocessing method for dealing with imbalanced data.
(3) Extracting other features from radiographic images with the hope to find more discriminant features.
(4) Trying other classifiers or learning algorithms to obtain better than 1-NN results.

References

Alejo, R., García, V., Sotoca, J. M., Mollineda, R. A., & Sánchez, J. S. (2006). Improving the classification accuracy of RBF and MLP neural networks trained with imbalanced samples. In E. Corchado et al. (Eds.), *IDEAL 2006*, LNCS 4224 (pp. 464–471).

Aoki, L., & Suga, Y. (1999). Application of artificial neural network to discrimination of defect type in automatic radiographic testing of welds. *ISIJ International, 39*(10), 1081–1087.

Barandela, R., Sánchez, J. S., García, V., & Rangel, E. (2003). Strategies for learning in class imbalance problems. *Pattern Recognition, 36*, 849–851.

Batista, G. E. A. P. A., Monard, M. C., & Bazzan, L. C. (2004). Improving rule induction precision for automated annotation by balancing skewed data sets. In J. A. López et al. (Eds.), *KELSI 2004*, LNAI 3303 (pp. 20–32).

Batista, G. E. A. P. A., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *Sigkdd Explorations, 6*(1), 20–29.

Carrasco, M., & Mery, D. (2004). Segmentation of welding discontinuities using a robust algorithm. *Materials Evaluation*(November), 1142–1147.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357.

Chawla, N. V., Japlowicz, N., & Kotcz, A. (2006). Editorial: Special issue on learning from imbalanced data sets. *Aigkdd Explorations, 6*(1), 1–6.

Cohen, G., Hilario, M., Sax, H., Hogonnet, S., & Geissbuhler, A. (2006). Learning from imbalanced data in surveillance of nosocomial infection. *Artificial Intelligence in Medicine, 37*, 7–18.

Daum, W., Rose, P., Heidt, H., & Builtjes, J. H. (1987). Automatic recognition of weld defects in X-ray inspection. *British Journal of NDT, 29*(2), 79–82.

Felisberto, M. K., Lopes, H. S., Centeno, T. M., & Arruda, L. V. A. (2006). An object detection and recognition system for weld bead extraction from digital radiographs. *Computer Vision and Image Understanding, 102*, 238–249.

Garcia, S., & Cano, J. R. (2006). A proposal of evolutionary prototype selection for class imbalanced problems. In E. Corchado et al. (Eds.), *IDEAL 2006*, LNCS 4224 (pp. 1415–1423).

Gayer, A., Saya, A., & Shiloh, A. (1990). Automatic recognition of welding defects in real-time radiography. *NDT International, 23*(3), 131–136.

Han, H., Wang, W. -Y., & Mao, B. -H. (2005). Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In D. S. Huang et al. (Eds.), *ICIC 2005*, Part I, LNCS 3644 (pp. 878–887).

Hart, P. E. (1968). The condensed nearest neighbor rule. *IEEE Transformation Information Theory, 18*, 515–516.

Huang, Y.-M., Hung, C.-M., & Jiau, H. C. (2006). Evaluation of neural networks and data mining methods on a credit assessment task for class imbalance problem. *Nonlinear Analysis: Real World Applications, 7*, 720–747.

Hyatt, R., Kechter, G. E., & Nagashima, S. (1996). A method for defect segmentation in digital radiographs of pipeline girth welds. *Materials Evaluation*, 925–928.

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent Data Analysis, 6*, 429–449.

Kaftandjian, V., Dupuis, O., Babot, D., & Zhu, Y. M. (2003). Uncertainty modeling using Dempster-Shafer theory for improving detection of weld defects. *Pattern Recognition Letters, 24*, 547–564.

Kato, Y., Okumura, T., Matsui, S., Itoga, K., Harada, T., Sugimoto, K., et al. (1992). Development of an automatic weld defect identification system for radiographic testing. *Welding in the World, 30*(7/8), 182–188.

Kubat, M., Holte, R. C., & Matwin, S. (1998). Machine learning for the detection of oil spills in satellite radar images. *Machine Learning, 30*, 195–215.

Kubat, M., & Matwin, S. (1997). Addressing the curse of imbalanced training sets: One-sided selection. In *Proceedings of the 14th international conference on machine learning* (pp. 179–186).

Laurikkala, J. (2001) Improving identification of difficult small classes by balancing class distribution. Report A-2001-2. Department of Computer and Information Sciences, University of Tampere.

Liao, T. W. (2003). Classification of welding flaw types with fuzzy expert systems. *Expert Systems with Applications, 25*, 101–111.

Liao, T. W. (2004). Fuzzy reasoning based automatic inspection of radiographic welds: Weld recognition. *Journal of Intelligent Manufacturing, 15*, 69–85.

Liao, T. W., & Li, Y. M. (1998). An automated radiographic NDT system for weld inspection: Part II. Flaw detection. *NDT&E International, 31*(3), 183–192.

Liao, T. W., Li, D. M., & Li, Y. M. (1999). Detection of welding flaws from radiographic images with fuzzy clustering methods. *Fuzzy Sets and Systems, 108*(2), 145–158.

Liao, T. W., Li, D. M., & Li, Y. M. (2000). Extraction of welds from radiographic images using fuzzy classifiers. *Information Sciences, 126*, 21–42.

Liao, T. W., & Ni, J. (1996). An automated radiographic NDT system for weld inspection: Part I. Weld extraction. *NDT&E International, 29*(3), 157–162.

Liao, T. W., & Tang, K. (1997). Automated extraction of welds from digitized radiographic images based on MLP neural networks. *Applied Artificial Intelligence, 11*, 197–218.

Liu, Y., Chawla, N. V., Harper, M. P., Shrilberg, E., & Stolcke, A. (2006). A study in machine learning from imbalanced data for sentence boundary detection in speech. *Computer Speech and Language, 20*, 468–494.

Murakami, K. (1990). Image processing for non-destructive testing. *Welding International, 4*(2), 144–149.

Murphey, Y. L., Guo, H., & Feldkamp, L. A. (2004). Neural learning from unbalanced data. *Applied Intelligence, 21*, 117–128.

Rifkin, R., & Klautau, A. (2004). In defense of one-vs-all classification. *Journal of Machine Learning Research, 5*, 101–141.

Tomek, I. (1976). Two modifications of CNN. *IEEE Transactions Systems, Man, and Communications, 6*, 769–772.

Wang, G., & Liao, T. W. (2002). Automated identification of different types of welding defects in radiographic images. *NDT&E International, 35*, 519–528.

Wang, J., & Jean, J. (1993). Resolve multifont character confusion with neural network. *Pattern Recognition, 26*(1), 173–187.

Wang, X., & Wong, S. (2005). Radiographic image segmentation for weld inspection using a robust algorithm. *Research in Nondestructive Evaluation, 16*, 131–142.

Weiss, G. M. (2004). Mining with rarity: A unifying framework. *Sigkdd Explorations, 6*(1), 7–19.

Wilson, D. L. (1972). Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions of Systems, Man, and Cybernetics, 2*, 408–421.

Xie, J., & Qiu, Z. (2007). The effect of imbalanced data sets on LDA: A theoretical and empirical analysis. *Pattern Recognition, 40*, 557–562.