An Improved SMOTE Algorithm Based on Genetic Algorithm for Imbalanced Data Classification

GU Qiong[1, 2], WANG Xian-Ming[3*], WU Zhao[1], NING Bing[1], XIN Chun-Sheng[1, 4]

[1]School of Mathematics and Computer Science
Hubei University of Arts and Science
Xiangyang, 441053, China

[2]Institute of Logic and Intelligence, Southwest University
Chongqing, 400715, China

[3]Oujiang College, Wenzhou University, Wenzhou, 325035, China

[4]Department of Electrical and Computer Engineering
Old Dominion University, Norfolk, USA
xmwung@163.com

JDIM
Journal of Digital
Information Management

***ABSTRACT**: Classification of imbalanced data has been recognized as a crucial problem in machine learning and data mining.In an imbalanced dataset, minority class instances are likely to be misclassified. When the synthetic minority over-sampling technique (SMOTE) is applied in imbalanced dataset classification, the same sampling rate is set for all samples of the minority class in the process of synthesizing new samples, this scenario involves blindness. To overcome this problem, an improved SMOTE algorithm based on genetic algorithm (GA), namely, GASMOTE was proposed. First, GASMOTE set different sampling rates for different minority class samples. A combination of the sampling rates corresponded to an individual in the population. Second, the selection, crossover, and mutation operators of GA were iteratively applied to the population to obtain the best combination of sampling rates when the stopping criteria were met. Lastly, the best combination of sampling rates was used in SMOTE to synthetize new samples. Experimental results on 10 typical imbalanced datasets show that GASMOTE increases the F-measure value by 5.9% and the G-mean value by 1.6% compared with the SMOTE algorithm. Meanwhile, GASMOTE increases the F-measure value by 3.7% and the G-mean value by 2.3% compared with the borderline-SMOTE algorithm. GASMOTE can be utilized as a new over-sampling technique to address the problem of imbalanced dataset classification. The GASMOTE algorithm can be then adopted in a practical engineering application, namely, prediction of rockburst in VCR rockburst datasets. The experimental results indicate that the GASMOTE algorithm can accu rately predict the rockburst occurrence and thus provides guidance to the design and construction of safe deep-mining engineering structures.*

Subject Categories and Descriptors
K.2.8 [Database Applications]: Data Mining

General Terms: Imbalanced Dataset, Data Mining, Classification

Keywords: Imbalanced Dataset, Classification, SMOTE, Sampling Rate, Genetic Algorithm, Rockburst

Received: 1 November 2015, Revised: 27 December 2015, Accepted: 30 December 2015

1. Introduction

Learning from imbalanced data has become a significant problem in many applications, such as biomedical data analysis [1–3], detection of oil spills in satellite radar images [4], text classification [5], and detection of fraudulent telephone calls [6]. Classification of imbalanced data is an important problem in machine learning and data mining [7]. In an imbalanced dataset, significantly fewer training instances exist in one class compared with another class. Correspondingly, the former is known as the minority class, and the latter is called the majority class. For an imbalanced dataset, most of the standard learning or classification algorithms tend to classify the majority class with a high accuracy rate and the minority class with a low accuracy rate [8]. This difference in accuracy rate results in poor performance of the classifier in diagnosing minority class samples. Thus, classification of an imbalanced dataset is a great challenge in classification research. In many applications, minority class samples are more important to identify than majority ones [9]. For example, most credit card transactions are normal in a credit card fraud test, and real credit card fraud transactions are few. However, identifying the few real credit card fraud transactions is important. In diagnosing diseases, common diseases are easier to diagnose than rare diseases. However, rare diseases, such as cancer, often need a timely diagnosis for effective treatment. Therefore, accurate classification of minority class samples in imbalanced datasets has become a popular research issue and poses a great challenge in data mining and machine learning [10].

The synthetic minority over-sampling technique (SMOTE) is utilized to classify imbalanced datasets. This technique synthesizes new samples of the minority class to balance a dataset by re-sampling the instances of the minority class. However, existing algorithms based on SMOTE use the same sampling rate for all instances of the minority class. This approach results in sub-optimal performance.

In this study, we propose a novel genetic algorithm (GA) [11] based on SMOTE (referred to as GASMOTE) to improve the performance of imbalanced data classification. The GASMOTE algorithm utilizes different sampling rates for different minority class instances and identifies the combination of optimal sampling rates. The combination of sampling rates is formulated as an individual in a population in the context of GA. The combination of optimal sampling rates is intelligently searched for. After obtaining the optimal sampling rates, over-sampling is performed on the instance of the minority class by using the corresponding optimal sampling rate. The dataset obtained after over-sampling is utilized as the training dataset for the construction of the classifier. In addition to general imbalanced dataset classification, the proposed GASMOTE algorithm is also employed in a practical engineering application, namely, prediction of rockburst in VCR stope rockburst instance data established by the South Africa Academy of Science.

The remainder of this paper is organized as follows. Section 2 presents a review of current state-of-the-art techniques for the classification of imbalanced datasets and the performance measures for data classification. Section 3 introduces the proposed GASMOTE algorithm. Section 4 presents the experimental results and engineering application of the GASMOTE algorithm, and Section 5 provides the conclusion.

2. State-of-the-Art-Techniques

The major approaches proposed in literature for imbalanced data classification are re-sampling, cost-sensitive learning, ensemble learning, and active learning.

The re-sampling approach, which is also called dataset reconstruction, involves changing the distribution of training set samples by data processing to improve the classification performance by reducing the imbalance of a dataset. This approach includes over-sampling, under-sampling, and other mixed sampling approaches [12]. Data re-sampling may balance the data class distribution by removing the majority class samples with under-sampling or increasing the minority class samples with over-sampling. Given that minority class samples of original data are copied, over-sampling may result in border or noise data, increase the processing time, and lead to over-fitting with low efficiency. Chawla[13] proposed the SMOTE algorithm, which demonstrates good performance in over-sampling processing of sample sets. This algorithm can randomly create and generate new minority class sample points based on a certain rule and merge these newly generated sample points with the original dataset to generate new training sets. This approach can be utilized to select, copy, and synthesize new minority class samples; thus, the over-learning problem can be avoided in the random over-sampling approach to some extent. However, this approach does not consider the samples synthesized by new minority classes to some extent. In minority classes, different samples have different roles in the over-sampling process, and the samples in the minority class border have a greater role than the samples in the minority class center. Taking samples in the minority class border may improve the recognition rate of classification decision surface to the minority class samples, and taking samples in the minority class center reduces the imbalance rate of datasets. Many algorithms have been proposed to improve SMOTE. B.X. Wang et al. [14] indicated that the SMOTE algorithm is prone to class overlapping or over-generation. H. Han [15] proposed the borderline-SMOTE algorithm. By comparing the number of majority classes and class samples neighboring the border sample in minority classes, this algorithm includes a sample if it is located at the border of minority class samples. Over-sampling is then conducted for the border samples of minority classes, i.e., interpolation is performed in the appropriate area. On this basis, H. He [16] made an improvement to ensure that the newly added samples are valuable. Chawla [17] believed that the lifting algorithm tends to provide minority class samples a large weight, which is equal to copying a part of the minority class samples, thereby achieving over-sampling. He combined lifting and sampling techniques and proposed the SMOTEBoost algorithm to improve performance in predicting minority classes after adding new minority class samples. Guo [18] designed the DataBoost-IM approach, which identifies the samples of majority and minority classes that are difficult to distinguish. Then, these samples are utilized to generate new synthesizing samples. Finally, the weights of different categories in the new dataset are balanced. According to Chen Si et al. [19], clustering and fusion of data prior to data pro cessing are implemented to identify the samples that are always located in the same class cluster in the multi-clustering process. Such samples are the center samples. The samples in the changing class cluster are the border samples. SMOTE sampling is then conducted for the border samples of minority classes, and under-sampling is conducted for the center samples of majority classes. Chen et al. [20] proposed the RAMOBoost approach, which determines each iterative learning process of minority class samples through the adaptive lifting technique based on the sampling probability distribution and transfers them by self-adaption to the classification border of the minority and majority class samples. Among various re-sampling approaches, the under-sampling approach is used more often because it reduces the training set and time of model training and increases efficiency. Several under-sampling approaches, such as the condensed nearest-neighbor rule, neighborhood-cleaning rule, one-sided selection, and Tomek link, have been proposed. These approaches determine border, noise, and redundant samples by certain rules and strategies; selectively remove the majority class samples that have a few roles in the classification, are far away from the classification border, or induce data overlapping; and retain the safe and small class samples as the training set of the classifier. However, given the principle of the majority class subset selected by under-sampling for training, several majority class samples are randomly reduced to lower the scale of majority classes. Consequently, the effective information of majority class is easily lost, and the potential useful and important information may be omitted in the samples.

The different strategies for the cost-sensitive learning algorithm and sample approach lie in [21] the different misclassification costs used in the classification decision to minimize the total cost of misclassification rather than the error rate of misclassification. They are concerned with the misclassification instance cost and endow the minority class with a high misclassification cost. In this manner, the classifier can improve the classification accuracy rate of minority classes to address the imbalanced data processing. The cost-sensitive learning approach changes the existing classification algorithm and makes it difficult in cost sensitivity, with a poor effect sometimes. Zhou et al. [22] proposed the cost-sensitive neural network approach. The threshold-value-changing technique is applied to regulate the threshold value to the un-valued class, thus avoiding the incorrect classification of high-cost samples. Sun et al. [23] proposed three cost-sensitive lifting approaches, namely, AdaC1, AdaC2, and AdaC3, which adopt the weight update strategy in the lifting algorithm. Many cost-sensitive learning approaches are frequently used. The first approach is adjusting the sample distribution. In this approach, the frequency of a category in the training set in a proportion is changed based on the misclassification cost. Its shortcoming is that the distribution of samples is changed, which may affect the algorithm's performance sometimes. The second approach is meta cost, which involves modifying the class mark of a training sample based on the minimum expecting cost through the “meta-learning process” and re-learning a new model with the modified training set. The third approach is cost-sensitive decision, which involves obtaining samples many times in a training set to generate several models and determining the probability of a testing sample in each category based on the model. Then, all misclassification costs of the testing sample are calculated, and the class marking is obtained by minimizing the cost.

In ensemble learning approach [24], several classifiers are combined as one to improve classification performance. The lifting technique is widely used, i.e., lifting [25] several weak classifiers and combining them to form one strong classifier that can improve the classification performance of an imbalanced dataset. Prof. Zhou Zhihua [26] studied the ensemble learning boosting technique, in which several weak classifiers are combined into one strong classifier that can create an ensemble model by boosting iteration whether data are imbalanced or not; hence, the performance of the weak classifier is improved. Mikel G. et al. [27] believed that the advantages of applying the boosting technique to the imbalanced learning problem are as follows: the data space re-sampling can reduce extra costs to automatically detect optimal class distribution and representative samples, assembling several classifiers can effectively avoid model over-fitting, and the bias of a specific learning algorithm is reduced. AdaBoost [28], as a representative of lifting samples, can increase the sample weight of misclassification and reduce the sample weight of correct classification; therefore, the system focuses on samples with classification errors in the subsequent iteration, which effectively improves the distribution of training data and the classification performance of minority class samples. Other scholars [29, 30] adjusted the existing class distribution and improved the existing ensemble algorithms to obtain good classification performance by applying the support vector machine (SVM) with an asymmetric misclassification cost. The EasyEnsemble approach, a representative of the bootstrap-sampling policy proposed by Liu et al. [31], is an important achievement. In this algorithm, a large class sample is divided into several independent subsets, and each subset is trained by one sub-classifier. All sub-classifiers are integrated into the final classifier. With the algorithm, omission of effective information is avoided, and sample information is used sufficiently. Hence, stable and efficient results are obtained. The algorithm is widely recognized because it reduces the possibility of important sample loss, with high under-sampling efficiency.

The traditional active learning approach is mainly utilized to solve imbalanced training data. Many scholars have recently proposed numerous active learning approaches for imbalanced datasets, including the SVM-based active learning approach proposed by Ertekin et al. [32, 33]. SVM-based active learning means that a group of training instances are effectively selected from random training datasets to significantly reduce the calculation cost when

	Predicted aspositive	Predicted asnegative
Actual positive class	TP	FN
Actual negative class	FP	TN

Table 1. Confusion matrix for a two-class classification problem

numerous imbalanced datasets are processed.

3. Methodology

3.1 Performance Evaluation Measures

Performance evaluation metrics play a crucial role in assessing classification performance and guiding the classifier design. Most of the studies on imbalanced data concentrated on the two-class classification problem because the multi-class problem can be simplified into a two-class problem. By convention, the class label of the minority class is positive, whereas the class label of the majority class is negative. Table 1 presents a confusion matrix of a two-class problem. The first column in the table is the actual class label of the samples, and the first row is their predicted class label. True positive (TP) and true negative (TN) denote the number of positive and negative samples that are correctly classified, respectively. False negative (FN) and false positive (FP) denote the number of misclassified positive and negative samples, respectively.

Total classification accuracy is the most commonly used performance measure. However, in the classification of imbalanced data, total accuracy is no longer a proper measure because the uncommon or rare class has minimal impact on accuracy compared with the prevalent class [34]. In fact, this measurement is meaningless to several applications in which the learning concern is the identification of rare class instances. If only the performance of the rare or positive class is considered, two measures are important, namely, TP rate (TP_{rate}) and positive predictive value (PP_{value}). In information retrieval, TP_{Rate} is defined as the recall denoting the percentage of the retrieved objects that are relevant.

$$\mathrm{Re}call = TP_{rate} = \frac{TP}{TP + FN} \qquad (1)$$

PP_{value} is defined as the precision denoting the percentage of the relevant objects that are identified for retrieval.

$$\mathrm{Pr}ecision = PP_{value} = \frac{TP}{TP + FP} \qquad (2)$$

The F-measure is also a popular performance metric for the imbalanced data classification problem [35]. This metric is a combination of recall and precision, which are effective metrics for information retrieval in which the data imbalance problem exists. The *F-measure* depends on the β factor, which is a parameter that has a value from 0 to infinity and is used to control the effects of recall and precision. When $\beta = 0$, *F-measure* is reduced to precision; when $\beta=\infty$, *F-measure* approaches recall.

$$F - measure = \frac{(1+\beta^2)\times recall \times precision}{\beta^2 \times recall + precision} \qquad (3)$$

When $\beta = 1$, *F-measure* integrates the two measures as an average, i.e., *F-measure* represents a harmonic mean between recall and precision.

$$F - measure = \frac{2 \times recall \times precision}{recall + precision} \qquad (4)$$

The harmonic mean of two numbers tends to be closer to the smaller of the two. Hence, a high F-measure value ensures that both recall and precision are reasonably high. When the performance of both classes is concerned, both TP_{rate} and TN_{rate} are expected to be high simultaneously. Kubat et al. [36] recommended the *G-mean* as a good performance measure.

$$G - mean = \sqrt{TP_{rate} \times TN_{rate}} \qquad (5)$$

The G-mean measures the balanced performance of a learning algorithm. A comparison of harmonic, geometric, and arithmetic means is presented in [35]. TPrate is utilized to evaluate the classification performance of the minority class, and TNrate is utilized to evaluate the classification performance of the majority class. The G-mean value is large when both TPrate and TNrate are high. Thus, the G-mean index is utilized to measure the overall classification accuracy rate.

To assess the performance of a classifier for imbalanced datasets, the focus should be on the performance of the minority class classification. Therefore, in this study, we used two performance measures, F-measure and G-mean, instead of the total accuracy metric. The F-measure index was used to evaluate the classification of the minority class in an imbalanced dataset, and the G-mean index was used to evaluate the overall classification performance of the imbalanced dataset.

3.2 SMOTE

SMOTE is an over-sampling method [13]. Its main idea is to create new minority class instances by interpolating several original minority class instances that lie together. SMOTE randomly selects one (or more depending on the over-sampling ratio) of the k nearest neighbors of a minority class instance and conducts a random interpolation of two instances to create a new synthetic instance. The synthetic instance is generated in the following manner: the difference between an original instance and its nearest neighbor is obtained, this difference is multiplied by a random number between 0 and 1, and the result is added where *ge* is the current generation, *Ge* is the maximum generation, and *rnd(2)* is the result obtained after positive to the original instance. Essentially, a random point is selected along the line segment between the original instance and its nearest neighbor. This approach effectively forces the decision region of the minority class to become general. Thus, the over-fitting problem is avoided, and the decision boundary for the minority class spreads further into the majority class space. Based on SMOTE, several algorithms have been proposed for the classification of imbalanced data [10, 15].

Existing SMOTE algorithms have one issue: they use the same sampling rate for all instances of the minority class. Different instances have different roles in sampling and classification. The corresponding sampling rate should be set based on the role of an instance. Hence, using the same sampling rate for all instances results in sub-optimal classification performance. The proposed GASMOTE algorithm that finds and uses the optimal sampling rates for different instances is described below.

3.3 GASMOTE Algorithm

Selecting samples from the minority class for over-sampling and setting of the sampling rate are related to the imbalanced degree of the dataset, overall distribution of samples, the internal distribution of minority class samples, number of samples, number of sample attributes, and types of attributes. These are complicated optimizing problems that can be solved by a determined mathematical model. GA, as a random searching optimization algorithm based on the genetic evolution law, is suitable [36] for solving multi-dimensional nonlinear complicated problems; it is widely used in function and engineering optimization [37, 38].

Different instances in the original training set are associated with sampling rates to obtain the highest accuracy rate of minority class classification and a good overall classification accuracy rate. The following GA is utilized to obtain the optimal sampling rates for different instances.

$$\begin{gathered}\max imize: y = f(X); s.t.: \min N \leqslant N_i \leqslant \max N; \\ i = 1, 2, \cdots, M; where: X = (N_1, N_2 \cdots, N_M),\end{gathered} \tag{6}$$

where *f(X)* is the objective function, i.e., the accuracy rate of the minority class classification and the overall classification of datasets; X is the decision vector, i.e., the sampling rates; *M* stands for the dimension of decision space, i.e., the number of minority class samples; N_i is the sampling rate of the minority class sample x_i; and *minN and maxN* are the lower and upper bounds of sampling rate N_i, respectively.

3.3.1 GASMOTE Algorithm

The proposed GASMOTE algorithm employs a GA to find the optimized sampling rates and generates a new dataset through over-sampling by using the optimized sampling rates. The algorithm consists of five steps.

Step 1. Encoding and initialization: In this step, a population of size P is generated for the GA. N_i denotes the sampling rate of the minority class instance x_i. In the context of a GA, we use an individual in a population to represent a combination of the sampling rates for all instances as follows:

$$X^j = (N_1^j, N_2^j, ..., N_M^j), j = 1, 2, ..., P, \tag{7}$$

where M stands for the length of a chromosome, i.e., the number of minority class instances, and P is the size of the population.

To initialize an individual, each node of the chromosome is set as a random integer value between the upper and lower bounds of the sampling rate, i.e.,

$$\begin{gathered}N_i^j = round(minN + (maxN \quad minN) \times rand(0,1)) \\ i = 1, 2, ..., M; j = 1, 2, ..., P,\end{gathered} \tag{8}$$

where *round*( ) stands for the rounding-off function. In other words, a matrix of random numbers bounded by the lower and upper bounds of the sampling rates is generated. Each column X^j is an individual.

Step 2. Selection operation: In this step, the fitness function value for each individual in the population is calculated, and the population is sorted in a descending order of the fitness function value. Pr denotes the selection probability. We duplicate (i.e., generate two copies) the first $P \times Pr$ individuals in the sorted population, eliminate the last $P \times Pr$ individuals in the population, and retain the individuals in the middle to generate a new population.

Step 3. Crossing operation: We perform part of disperse hybrid operators at $Pop/3$ times in the crossing operation, i.e., we randomly select two individuals from the population each time denoted as X^i and X^j as follows:

$$X^i = (N_1^i, N_2^i, \cdots, N_k^i, N_{k+1}^i, N_{k+2}^i \cdots, N_M^i) \tag{9}$$

$$X^j = (N_1^j, N_2^j, \cdots, N_k^j, N_{k+1}^j, N_{k+2}^j \cdots, N_M^j) \tag{10}$$

The node of the initial crossing is then randomly selected and denoted as k. All nodes after k in X^i and X^i are crossed, and the new individuals after crossing on X^i and X^j are as follows:

$$X^i = (N_1^i, N_2^i, \cdots, N_k^i, N_{k+1}^j, N_{k+2}^j \cdots, N_M^j), \tag{11}$$

$$X^j = (N_1^j, N_2^j, \cdots, N_k^j, N_{k+1}^i, N_{k+2}^i \cdots, N_M^i) \tag{12}$$

Step 4. Mutation operation: A random number between 0 and 1 is generated for each individual in the population. If the random number is smaller than mutation probability *Pm*, non-uniform mutation [39] occurs; otherwise, no mutation occurs. Supposing that mutation occurs for individual X^i, we randomly select a node in X^i, such as node k. The node value after mutation becomes

$$N_k^i = \begin{cases} N_k^i + (maxN - N_k^i) \times (1 - rand(0,1)^{(1-ge/Ge)^3}), rnd(2) = 0 \\ N_k^i - (N_k^i - minN) \times (1 - rand(0,1)^{(1-ge/Ge)^3}), rnd(2) = 1 \end{cases}, \tag{13}$$

where *ge* is the current generation, *Ge* is the maximum generation, and *rnd(2)* is the result obtained after positive integer module 2 is randomly generated equably.

Step 5. Termination check: If the termination condition is met, i.e., current generation ge is greater than maximum generation Ge, the algorithm outputs the optimal individual; otherwise, Step 2 is repeated. If the search for the optimal sampling rates terminates, the dataset is generated through SMOTE over-sampling by using the optimal sampling rates.

The pseudo-code for the GASMOTE algorithm is shown in Table 2.

```
GASMOTE Algorithm
1: Start
2: Inputting training set: Train
3: Initializing: Population Pop of P, Individual $X_i$ is a combination of the sampling rate of each minority class sample in Train.
4: The training set BestSmotedTrain is empty after initializing optimal sampling.
5: Calculate fitness function:
   Take the sample for Train based on $X_i$ to generate $SmotedTrain_i$;
   Perform classification by $SmotedTrain_i$ as a training set and Train as a testing set;
   Take the classification index G-mean value as the i-th individual of fitness function value $fitness_i$
6: While ($fitness_k \geq fitness_i$,(i = 1,2,… P))
7: $BestSmotedTrain = SmotedTrain_k$;
8: Selection operation;
9: Crossover operation;
10: Mutation operation;
11: If the termination condition is met
12: output BestSmotedTrain (best training set after sampling);
13: else go to 5
14: endwhile
15:  End
```

Table 2. Pseudo-code of the GASMOTE algorithm

3.3.2 Design of the Fitness Function

The fitness function is an index to evaluate good or poor individuals in a population and serves as a bridge to link GA with the specific optimization problem. Therefore, the selection of the fitness function is crucial in GA. In the GASMOTE algorithm, an individual in a population is a combination of the sampling rates of minority class samples. Based on this combination, SMOTE sampling is conducted for the original dataset. The obtained dataset is utilized as the training set for the classifier. The original dataset is then classified. The evaluation index G-mean [35] is calculated based on the classification result, and this index value is regarded as the fitness function value. A large fitness function value indicates an excellent representative individual. Given that the C4.5 decision tree algorithm [40] is the standard algorithm of the decision tree, the C4.5 classification algorithm is applied for sample classification in the GASMOTE fitness function calculation.

4. Results, Analysis and Discussion

4.1 Experiment and Analysis

4.1.1 Experimental Datasets

The performance of the GASMOTE algorithm is evaluated with the 10 commonly used imbalanced datasets in literature [41], which are publicly available on the corresponding webpage [42]. Table 3 summarizes the properties of the selected imbalanced datasets. For each dataset, the dataset name (Datasets), number of examples (#Ex.), number of attributes (#Atts.), percentage of examples of each class (%min; %maj), and imbalanced rate of datasets (IR) are provided in the table.

Datasets	#Ex.	#Atts.	(%min; %maj)	IR
Yeast3	1484	8	(10.98,89.02)	8.11
Ecoli3	336	7	(10.88,89.12)	8.19
Yeast2vs4	514	8	(9.92,90.08)	9.08
Yeast05679vs4	528	8	(9.66,90.34)	9.35
Glass2	214	9	(8.78,91.22)	10.39
Ecoli4	336	7	(6.74,93.26)	13.84
Glass016vs5	184	9	(4.89,95.11)	19.44
Glass5	214	9	(4.20,95.80)	22.81
Yeast2vs8	482	8	(4.15,95.85)	23.10
Yeast4	1484	8	(3.43,96.57)	28.41

Table 3. Summary description of imbalanced datasets

4.1.2 Experimental Results and Analysis

A baseline classifier is defined first. The C4.5 learning algorithm constructs a decision tree top–down by using the normalized information gain (difference in entropy) that results from selecting an attribute for data splitting. The attribute with the highest normalized information gain is utilized to make a decision. We can consider the use of a classification tree algorithm that is specifically designed for the solution of imbalanced problems. Almost all the ensemble methodologies that we test in this study were proposed in combination with C4.5. C4.5 is widely used to deal with imbalanced datasets and is one of the top 10 data-mining algorithms. Given these facts, we select this algorithm as the most appropriate base learner. The C4.5 algorithm is utilized as the classification algorithm to compare the combination with SMOTE, borderline-SMOTE, and GASMOTE as well as the classification performance without over-sampling. The general

Datasets	C4.5	SMOTE+C4.5	Borderline-SMOTE+C4.5	GASMOTE+C4.5
Yeast3	0.751	0.767	0.767	0.797
Ecoli3	0.540	0.598	0.615	0.667
Yeast2vs4	0.644	0.734	0.701	0.778
Yeast05679vs4	0.405	0.468	0.453	0.511
Glass2	0	0.222	0.255	0.286
Ecoli4	0.732	0.549	0.727	0.773
Glass016vs5	0.533	0.818	0.818	0.842
Glass5	0.632	0.818	0.818	0.818
Yeast2vs8	0	0.524	0.556	0.564
Yeast4	0.129	0.305	0.314	0.360
Average	0.4366	0.5803	0.6024	0.6396

Table 4. Comparison of the F-measure of each algorithm (best if in bold)

Datasets	C4.5	SMOTE+C4.5	Borderline-SMOTE+C4.5	GASMOTE+C4.5
Yeast3	0.8778	0.8995	0.8995	0.9123
Ecoli3	0.6843	0.8592	0.8518	0.8499
Yeast2vs4	0.7345	0.9286	0.8721	0.8929
Yeast05679vs4	0.5533	0.7548	0.7154	0.7745
Glass2	0	0.5096	0.5568	0.5648
Ecoli4	0.8578	0.8138	0.8830	0.9117
Glass016vs5	0.6625	0.9885	0.9885	0.9375
Glass5	0.8087	0.9902	0.9902	0.9902
Yeast2vs8	0	0.7327	0.7025	0.7352
Yeast4	0.2786	0.7017	0.6410	0.7651
Average	0.5458	10.8179	0.8101	0.8334

Table 5. Comparison of the G-mean of each algorithm (best if in bold)

consensus in the data-mining community is that using 10-fold cross validation is a good compromise. The experimental process is available on the Waikato Environment for Knowledge Analysis [43] platform, and the experimental results are obtained through 10-fold cross validation. In the 10-fold cross validation, the original sample is randomly partitioned into 10 sub-samples. Out of the 10 sub-samples, a single sub-sample is retained as the validation data for testing the model. The remaining nine subsamples are utilized as training data. The cross-validation process is then repeated 10 times. This value of 10 is particularly attractive because it makes predictions using 90% of the data, thereby making it likely to be generalizable to the full data.

Tables 4 and 5 show the F-measure and G-mean values of classifications for the 10 datasets by the four algorithms (C4.5, SMOTE+C4.5, borderline-SMOTE+C4.5, and GASMOTE+C4.5). The average F-measure and G-mean values of the 10 datasets are listed in the last row for each approach. The maximum value for each dataset, i.e., the optimal result of the classification, is emphasized with bold text. Based on the data in Tables 4 and 5, the classification performance comparison figures for the four algorithms are obtained (Figures 1 and 2).

The experimental results in Tables 4 and 5 indicate that the classification performance of the GASMOTE+C4.5 algorithm is obviously superior to that of the three other algorithms. Figures 1 and 2 show that the broken lines representing the GASMOTE+C4.5 algorithm are basically in the most upward side of the figures. In the classification performance for minority classes, the GASMOTE+C4.5 algorithm has the largest *F-measure* value for all 10 datasets. It also has the largest *G-mean* value for seven datasets because different sampling rates are set for different samples of the minority class in the GASMOTE+C4.5 algorithm. Furthermore, the GASMOTE+C4.5 algorithm searches for the optimal sampling rate combination. For the three other algorithms, a fixed sampling rate is required to be set for each dataset. If the value is not appropriately set, the algorithm classification performance will be poor. Another advantage of the GASMOTE+C4.5 algorithm is that regardless of the change in sample number, attribute, imbalanced rate, and/or sample distribution of the dataset, the GASMOTE+C4.5 algorithm can find the optimal sampling rate combination self-adaptively and smartly. Therefore, this algorithm obtains the maximum classification performance in most imbalanced datasets. For the *G-mean* index, however, the GASMOTE+C4.5 algorithm does not have the maximum value in three datasets probably because the algorithm converges to a local optimal solution for the rate combination.

4.2 Engineering applications of the GASMOTE algorithm

Research on the rockburst mechanism has indicated that two factors, i.e., internal and external, exist during the occurrence of rockburst. The rockburst mechanism can be influenced by complex geological, engineering–environmental, and human excavation factors. For the strength-theory-based stress criteria, energy criterion, rigidity-theory-based outburst proneness criteria, fracture and damage theories, and dynamic disturbance theory, the test results of specific rock mass in the laboratory are generally utilized to predict the field engineering outburst proneness. The practicability and application scope of these indexes are substantially limited because accurately measuring or calculating them is difficult. In existing rockburst risk prediction, all classification training data are assumed as classification and prediction under the premise of imbalance. Nevertheless, rockburst occurs often in an imbalanced condition. If the previous method is utilized for small samples of rockburst and unbalanced categories of the potential hazards of rockburst, the prediction category that we are really concerned with will not achieve the optimal prediction result. Failure to consider the imbalanced prediction training data leads to a preference for the majority class data and neglect of the minority class data.

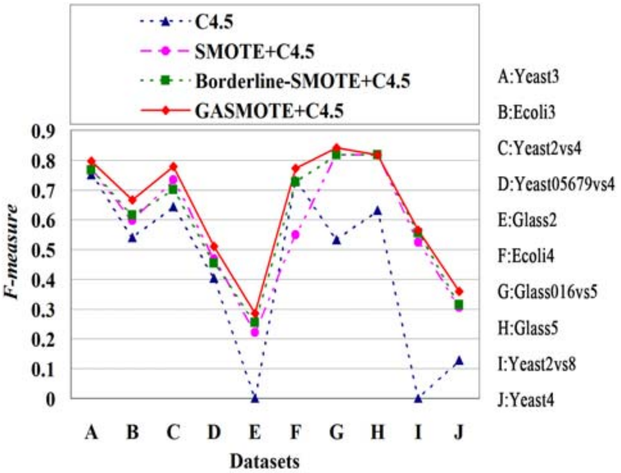


Figure 1. Comparison of the F-measure of the four algorithms

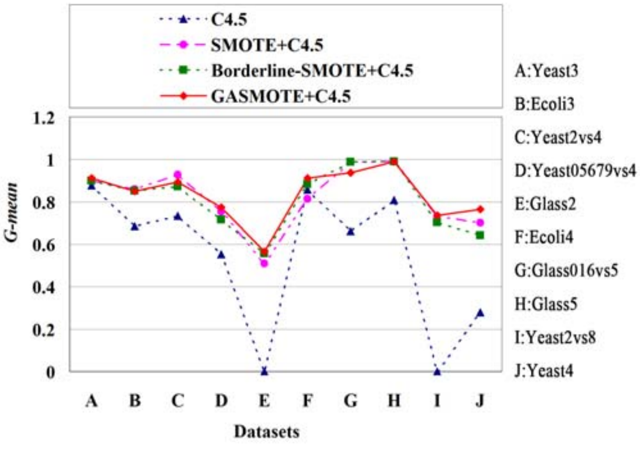


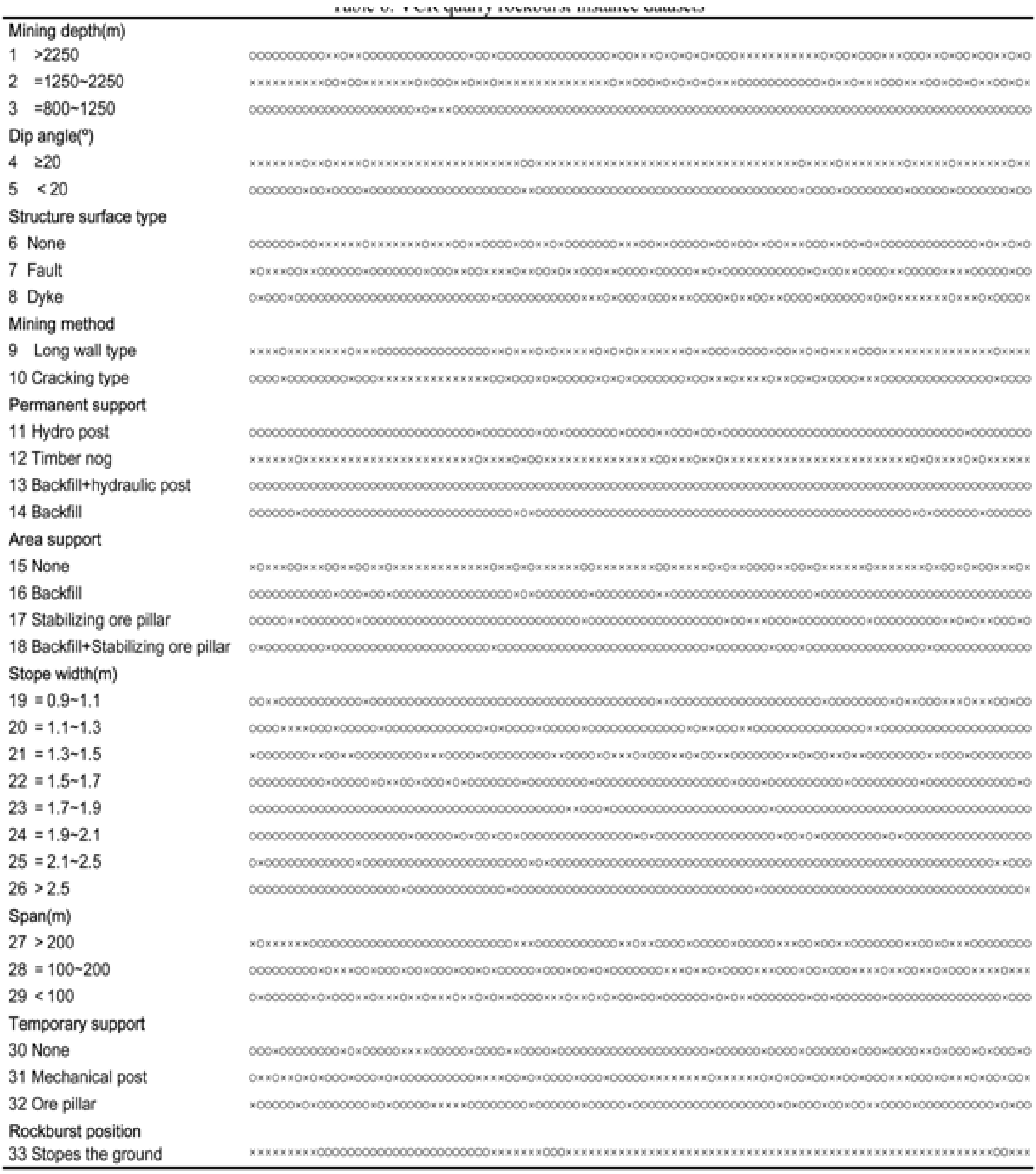
Figure 2. Comparison of the G-mean of the four algorithms

We apply the GASMOTE algorithm to rockburst risk prediction to obtain an improved classification performance. The VCR quarry rockburst experimental data established by the South Africa Academy of Science are selected for the experiment [44]. In the rockburst risk estimation model of the VCR carbonization deposit stope face, the following influencing factors are considered: (1) buried depth, (2) dip angle, type, and mining method (long wall and cracking) of the geologic structure face, (3) type and effect of temporary, permanent, and area supports, (4) width, direction, and span of the stope, (5) location and scale of rockburst when it occurs, and (6) collection measures implemented after it occurs. The details of VCR data are provided in Table 6. The rockburst database is established through an analysis of rockburst-influencing factors and collection of rockburst instances in deep mining with discretized data, in which the main influencing factors of rockburst are used as the input vectors. The occurrence of rockburst is utilized as the output scalar, where 0 stands for occurrence and 1 is for non-occurrence. In the columns of Table 6, "×" stands for a characteristic real existence of each instance record. If no characteristic value exists, it is marked with "o." A total of 32 characteristics (influencing factors) exist for each rockburst record, and each record is expressed as a 32-dimensional vector space made up of 1 or 0, where "×"corresponds to 1 and "o" denotes 0. This value is equivalent to the discretized input attribute value. GASMOTE algorithm re-sampling training is conducted on pre-classified data, which are then predicted. Given that we only consider the classification after the rockburst data are re-sampled, the classification results of rockburst data are verified by the C4.5 decision tree algorithm before and after the re-sampling of the GASMOTE algorithm.

Experiment 1: We select 98 samples from the dataset for training. Without loss of generality, we select the first 98 instances in the VCR quarry rockburst instance dataset, i.e., the 1st to the 98th samples. Then, we use the trained model to predict the result for the subsequent six samples from the dataset, i.e., the 99th to the 104th samples. If re-sampling pre-processing is not conducted for the training data, then only two samples are predicted correctly, and the other four are predicted incorrectly with the basic C4.5 algorithm for test. On the contrary, if we apply GASMOTE re-sampling to the 98 original data, we can predict all six samples correctly. The detailed classification results of the VCR quarry rockburst risk prediction values with the GASMOTE algorithm are illustrated in Table 7. The prediction results are consistent with the actual values, indicating that the implementation of the scheme is feasible in the imbalanced risk instance data of engineering instance rockburst. The scheme has a high accuracy rate and a good engineering application prospect.

The decision tree generated by the VCR quarry rockburst instance data is shown in Figure 3.

Experiment 2: The main controlling factors for rockburst occurrence are studied to effectively explore how to control



Mining depth(m)	
1 >2250	○○○○○○○○○○××○××○○○○○○○○○○○○○○×○○×○○○○○○○○○○○○○○○×○○×××○×○×○×○×○○○×××××××××××○×○○×○○○×××○○○××○×○○×○○××○×○
2 =1250~2250	××××××××××○○×○○×××××××○×○○○××○××○×××××××××××××××○××○○○×○×○×○×○×××○○○○○○○○○○○×○××○×××○○○×××○○×○○×○××○○×○×
3 =800~1250	○○○○○○○○○○○○○○○○○○○○○○×○×××○○
Dip angle(º)	
4 ≥20	×××××××○××○××××○××××××××××××××××××××○○×××××××××××××××××××××××××××××××××××○××××○××××××××○×××××○×××××××○××
5 < 20	○○○○○○○×○○×○○○○×○○○○○○○○○○○○○○○○○○○○××○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○×○○○○×○○○○○○○○×○○○○○×○○○○○○○×○○
Structure surface type	
6 None	○○○○○○×○○××××××○×××××××○×××○○××○○○○×○○××○×○○○○○○○×××○○××○○○○○×○○×○○××○○×××○○○××○○×○×○○○○○○○○○○○○○×○××○×○
7 Fault	×○×××○○××○○○○○○×○○○○○○○×○○○××○○××××○××○○×○××○○○××○○○○×○○○○○××○×○○○○○○○○○○○×○×○○××○○○○××○○○○○××××○○○○○○×○○
8 Dyke	○×○○○×○○○○○○○○○○○○○○○○○○○○○○○○○○×○○○○○○○○○○○×××○×○○○×○○○×××○○○○×○××○○××○○○○×○○○○○○×○×○×××××××○×××○×○○○○×
Mining method	
9 Long wall type	××××○××××××××○×××○○○○○○○○○○○○○○○××○×××○×○×××××○×○×○×××××××○××○○○×○○○○×○○××○×○×××○○○××××××××××××××××○××××
10 Cracking type	○○○○×○○○○○○○○×○○○×××××××××××××××○○×○○○×○×○○○○○×○×○×○○○○○○○×○○×××○××××○××○○×○×○○○×××○○○○○○○○○○○○○○○○×○○○○
Permanent support	
11 Hydro post	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○×○○○○○○○×○○×○○○○○○○×○○○○××○○○×○○×○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○×○○○○○○○○
12 Timber nog	××××××○×××××××××××××××××××××××○××××○×○○×××××××××××××××○○×××○××○×××××××××××××××××××××××××○×○××××○×○××××××
13 Backfill+hydraulic post	○○
14 Backfill	○○○○○○×○○○○○○○○○○○○○○○○○○○○○○○○○○○○×○×○○×○×○○○○○○○×○○○○○○
Area support	
15 None	×○×××○○×××○○××○○××○×××××××××××××○××○×○××××××○○××××××××○○×××××○×○××○○○○××○○×○××××××○××××××××○×○○×○×○○×××○×
16 Backfill	○○○○○○○○○○○×○○○×○○×○○○○○○○○○○○○○○○○×○×○○○○○○○×○○○○○○○○××○○○○○○○○○○○○○○○○○○○×○○○○○○○○○○○○○○○○○○○○○○○○○○○○
17 Stabilizing ore pillar	○○○○○××○○○○○○○×○○○○○○○○○○○○○○○○○○○○○○○○○○○○○×○○○○○○○○○○○○○○○○○○×○○×××○○○×○○○○○○○○○×○○○○○○○○○××○×○××○○○×○
18 Backfill+Stabilizing ore pillar	○×○○○○○○○○×○○○○○○○○○○○○○○○○○○○○○×○○○○○○○○○○○○○○○○○○○○○○○○○○○×○○○○○○○×○○○×○○○○○○○○○○○○○○○○×○○○○○○○○○○○○○○
Stope width(m)	
19 = 0.9~1.1	○○××○○○○○○○○○○○×○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○××○○○○○○○○○○○○○○○○○○○○×○○○○○○○○×○××○○○×××○×××○○×○○
20 = 1.1~1.3	○○○○××××○○○×○○○○○×○○○○○○○○○○○○○×○×○○○○×○○○○○×○○○○○○○○○○○○○×○××○○○××○○○○○○○○○○○○○○○××○○○○○○○○○○○○○○○○○○○○
21 = 1.3~1.5	×○○○○○○○××○○××○○○○○○○○○×××○○○○×○○○○○○○○○××○○○○×○×××○×○○○××○×○○××○○○○○○○××○×○○××○××○○○○○○○○××○○○×○○○○○○○○
22 = 1.5~1.7	○○○○○○○○○○×○○○○○×○××○○×○○○×○×○○○○○○○×○○○○○○○○×○○○○○○○○○○○○○○○○○×○○○×○○○○○○○○○○×○○○○○○○○○×○○○○○○○○○○○○×○
23 = 1.7~1.9	○○××○○○×○○○○○○○○○○○○○○○○○○○○○×○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○
24 = 1.9~2.1	○○○○○○○○○○○○○○○○○○○○○×○○○○○×○×○○×○○×○○○○○○○○○○○○○○○×○×○○○○○○○○○○○○○○○○×○○×○×○○○○○○○○×○×○○○○○○○○○○○○○○○○○
25 = 2.1~2.5	○×○○○○○○○○○○○○×○○○○○○○○○○○○○○○○○○○○○○×○×○○○××○○○
26 > 2.5	○○○○○○○○○○○○○○○○○○○○○×○○○○○○○○○○○○○○×○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○×○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○×
Span(m)	
27 > 200	×○××××××○○○○○○○○○○○○○○○○○○○○○○○○○○○×××○○○○○○○○○○○××○××○○○○×○○○○○×○○○○○×××○○×○○××○○○○○○○××○○×○×××○○○○○○○○
28 = 100~200	○○○○○○○○○×○×××○○×○○○×○○×○○○×○○×○×○○○○○○×○○○×○○×○×○○○○○○○×××○××○×○○○○×××○○○×○○×○○○××××○××○○××○×○○○××××○×××
29 < 100	○×○○○○○○×○×○○○○××○×××○××○×××○××○×○××○○○○×××○××○×○×○○×○○×○○○○○○×○×○××○○○○○○○×○○×○○○○○○×○○○○○○○○○○○○○○○×○○○
Temporary support	
30 None	○○○×○○○○○○○○×○×○○○○○××××○○○○○○×○○○○××○○○○×○○○○○○○○○○○○○○○○○○○○×○○○○○○×○○○○×○○○○○○×○○○×○○○○××○×○○○×○×○○○×○
31 Mechanical post	○××○××○×○×○○○×○○○×○×○○○○○○○○○○××××○○×○×○○○○×○○○×○○○○○××××××××○××××××○×○×○○×○○××○○×○○○×××○○○×○×××○×○○×○○×
32 Ore pillar	×○○○○○×○×○○○○○○○×○×○○○○○×××××○○○○○○○○×○○○○○○×○○○○○×○○○○○○○○○○○○○○○○○○○×○×○○○×○○×○○××○○○○×○○○○○○○○○○×○×○○
Rockburst position	
33 Stopes the ground	×××××××××○○○○○○○○○○○○○○○○○○○○○○○×××××××○○○×××○○×××

Table 6. VCR quarry rockburst instance datasets

Predicted sample no.	Characteristic vector input	Prediction output	Actual situation
99	01010001100100001010000000010100	1 0	Rockburst occurring
100	10010100010100100000000010010001	0 1	No rockburst occurring
101	10010100100100100000000010001010	0 1	No rockburst occurring
102	01001010100100100010000000010001	1 0	Rockburst occurring
103	10010100100100001000010000010100	1 0	Rockburst occurring
104	01010001100100100000000001010010	1 0	Rockburst occurring

Table 7. VCR quarry rockburst prediction results

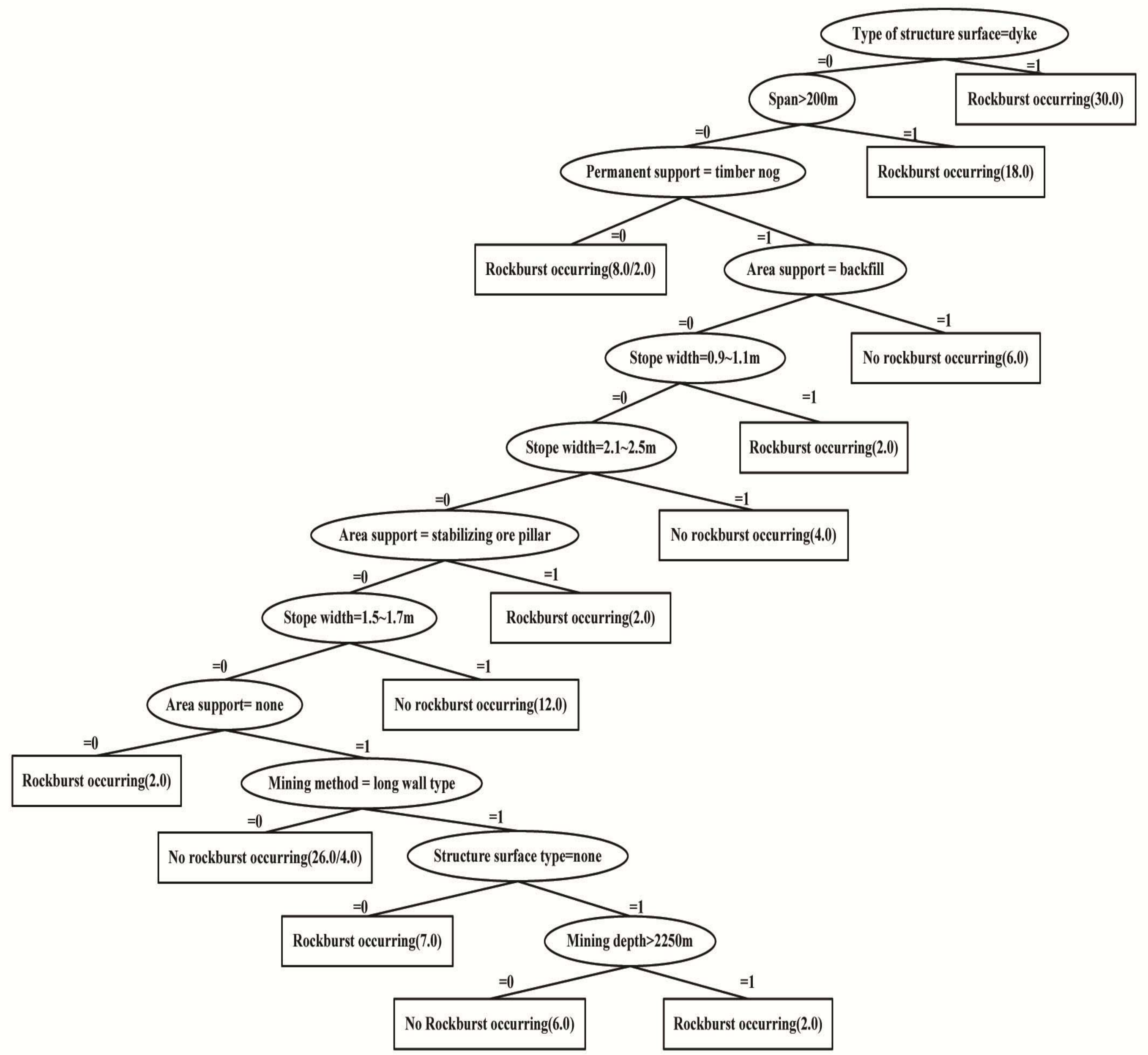

Type of structure surface=dyke
=0
=1
Span>200m
Rockburst occurring(30.0)
=0
=1
Permanent support = timber nog
Rockburst occurring(18.0)
=0
=1
Rockburst occurring(8.0/2.0)
Area support = backfill
=0
=1
Stope width=0.9~1.1m
No rockburst occurring(6.0)
=0
=1
Stope width=2.1~2.5m
Rockburst occurring(2.0)
=0
=1
Area support = stabilizing ore pillar
No rockburst occurring(4.0)
=0
=1
Stope width=1.5~1.7m
Rockburst occurring(2.0)
=0
=1
Area support= none
No rockburst occurring(12.0)
=0
=1
Rockburst occurring(2.0)
Mining method = long wall type
=0
=1
No rockburst occurring(26.0/4.0)
Structure surface type=none
=0
=1
Rockburst occurring(7.0)
Mining depth>2250m
=0
=1
No Rockburst occurring(6.0)
Rockburst occurring(2.0)

Figure 3. Untrimmed decision tree generated by the VCR quarry rockburst instance data

it. We take the characteristics of Instance 100 in Table 7 as a clue. The original influencing factor value of Instance 100 is marked as {1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1}, and no rockburst occurs as a result. We change the permanent support into other supports from the existing "permanent support = timber nog" or change the direction and span into "direction and span>200 m" from "span = 100–200 m." The rockburst risk state is then changed into rockburst occurrence. When the construction conditions are similar to that of Instance 100, the permanent support and span exert a significant influence on rockburst occurrence, and adverse support results in rockburst risk. Thus, the project construction party may reduce stress by strengthening the permanent support or increasing the project excavation width to lower the risk of rockburst occurrence. Similarly, the geological structure and mining depth have a significant influence on the possible occurrence of rockburst.

The results of the two engineering application examples indicate that the rockburst prediction result coincides with the actual situation. Synthesizing partial minority class data as training samples under imbalanced instance data of rockburst in the over-sampling method for the classification of imbalanced datasets is thus scientific and feasible. This method has high accuracy and an excellent prospect in engineering application. This method does not require the establishment of complex mathematical equations or mechanical calculation models. Given that the input data are objective or measurable, the method can be implemented simply.

5. Conclusions

All existing SMOTE algorithms use the same sampling rate for all minority class samples, which results in sub-optimal performance. A GA-based SMOTE over-sampling technique called GASMOTE was developed. With the GASMOTE algorithm, different sampling rates were used for over-sampling different minority class samples in imbalanced datasets, and the optimal sampling rate combination was determined. With the optimal sampling rates, SMOTE over-sampling was conducted for imbalanced datasets. The performance evaluation indicates that the proposed GASMOTE algorithm is superior to the original SMOTE and borderline-SMOTE algorithms in terms of overall classification accuracy rates for imbalanced datasets. GASMOTE was also used for risk prediction in rockburst instance data, which are also imbalanced. The results reveals that the prediction accuracy for rockburst occurrence is greatly improved compared with the original SMOTE algorithm. The proposed algorithm can be utilized to effectively identify the controlling factors corresponding to rockburst occurrence and provides a good scientific base for the design and construction of safe deep-mining structures.

Acknowledgments

This work was supported by the Humanities and Social Science Foundation of the Ministry of Education of China (15YJAZH015), National Natural Science Foundation of China (61172084), Zhejiang Provincial Natural Science Foundation (LY13F010005), Science and Technology Support Program of Hubei Province (2015BDH109,2015BHE029), and Science and Technology Development Foundation of Xiangyang.

References

[1] Anand, A., Pugalenthi, G., Fogel, G. B., Suganthan, P. N. (2010). An approach for classification of highly imbalanced data using weighting and undersampling. *Amino Acids*, 39 (5) 1385-91.

[2] Liu, L., Cai, Y. W., Feng, K., Peng, C., Niu, B. (2009). Prediction of protein-protein interactions based on pseaa composition and hybrid feature selection.*Biochemical & Biophysical Research Communications*, 380 (2) 318–322.

[3] He, H., Shen, X. (2007). A ranked subspace learning method for gene expression data classification.*In*: Proceedings of the 2007 International Conference on Artificial Intelligence(ICAI, p. 358-364.Las Vegas. Nevada. USA: CSREA Press, June 25-28.

[4] Kubat, M., Holte, R. C., Matwin, S. (1998). Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, 30 (2-3) 195-215.

[5] Dolores, M. (2004). Del castillo and josé ignacio serrano. a multistrategy approach for digital text categorization from imbalanced documents. *Sigkdd Explor News Letter*, 6 (1) 70-79.

[6] Phua, C., Alahakoon, D., Lee, V. (2004). Minority report in fraud detection: classification of skewed data. *ACM Sigkdd Explorations Newsletter*, 6 (1) 50-59.

[7] Soda, P. (2011). A multi-objective optimisation approach for class imbalance learning. *Pattern Recognition*, 44 (8) 1801–1810.

[8] He, H., Garcia, E. A. (2008). Learning from imbalanced data. *IEEE Transactions on Knowledge & Data Engineering*, 21 (9) 1263-1284.

[9] Qiong, G. U., Yuan, L., Xiong, Q. J., Ning, B., Wen-Xin, L. I. (2011). A comparative study of cost-sensitive learning algorithm based on imbalanced data sets. *Microelectronics & Computer*, 28 (8) 146-145.

[10] Wang, C., Pan, Z., Dong, L., Chunsen, M. A. (2013). Research on classification for imbalanced dataset based on improved smote.*Computer Engineering & Applications*, 49 (2) 184-170.

[11] Ji-Ke, G. E., Qiu, Y. H., Chun-Ming, W. U., Guo-Lin, P.U.(2008).Summary of genetic algorithms research. *Application Research of Computers*, 25 (10) 2911-2916.

[12] Estabrooks, A., Jo, T., Japkowicz, N. (2004). A multiple resampling method for learning from imbalanced data sets. *Computational Intelligence*, 20 (1) 18–36.

[13] Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P. (2011). Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research,* 16 (1) 321-357.

[14] Putthiporn,T, Chidchanok. L. (2013)handling imbalanced data sets with synthetic boundary data generation using bootstrap re-sampling and adaboost techniques, *Pattern Recognition Letters,* 34 (3) 1339-1347.

[15] Han, H., Wang, W. Y., Mao, B. H. (2005) Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. *In*:International Conference on Intelligent Computing(ICIC 2005), 878-887, Heidelberg, Berlin: Springer, August 23-26.

[16] He, H., Bai, Y., Garcia, E. A., Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *In*: IEEE Joint Conference on Neural Networks(IJCNN 2008), 1322-1328,Vancouver, BC, Canada: IEEE, June 1-8.

[17] Chawla, N. V., Lazarevic, A., Hall, L. O., Bowyer, K. (2003) SMOTEBoost: Improving prediction of the Minority Class in Boosting. *In*: 7th European Conference on Principles and Practice of Knowledge Discovery in Databases(PKDD2003), 107-119,Cavtat Dubrovnik,Croatia: Springer Berlin Heidelberg. September 22-26.

[18] Guo, H., Viktor, H. L. (2004). Learning from imbalanced data sets with boosting and data generation: the databoost-im approach. *Acm Sigkdd Explorations Newsletter*, 6 (1) 30-39.

[19] Chen, S., Guo, G. D., Chen, L. F. (2010). Clustering ensembles based classification method for imbalanced

data sets. *Pattern Recognition & Artificial Intelligence*, 23 (6) 772-780.

[20] Sheng, C., Haibo, H., Garcia, E. A. (2010). Ramoboost: ranked minority oversampling in boosting. *IEEE Transactions on Neural Networks*, 21 (10) 1624-1642.

[21]Ling, C., Shen, G., Victor, S(2007). A Comparative Study of Cost-Sensitive Classifiers. *Chinese Journal of Computers,* 30 (8) 1203-1212.

[22]Zhou, Z. H., Liu, X. Y. (2006). Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge & Data Engineering*, 18 (1) 63-77

[23]Sun, Y., Kamel, M. S., Wong, A. K. C., Wang,Y. (2007). Cost-sensitive boosting for classification of imbalanced data. Patter Chen, Q. G. (2007). Combined classifier algorithm for imbalanced datasets. *Computer Engineering and Design*, 28 (23) 5687-5690.

[25] Luo, B., Guang-Zhu, Y. U. (2007). Adaboost classification of multiple classes with imbalanced distribution. *Journal of Yangtze University*(Natural Science Edition), 4 (2) 50-54.

[26] Zhou, Z.H. (2012). Ensemble methods: foundations and algorithms. London, UK: Chapman & Hall.

[27] Galar, M., FernaìNdez, A., Barrenechea, E., Bustince, H., Herrera, F. (2012). A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems Man & Cybernetics* Part C, 42 (4) 463-484.

[28] Liao, H. W.,Zhou, D. L. (2012).Review of AdaBoost and Its Improvement.Computer Systems & Applications, 21 (5) 240-244.

[29] Liu,Y., An, A.J., Huang, X. J. (2006). Boosting Prediction Accuracy on Imbalanced Datasets with SVM Ensembles, *In:* 10th Pacific-Asia Conference(PAKDD 2006), 107-118,Singapore,: Springer Berlin Heidelberg, April 9-12.

[30] Wang, B. X., Japkowicz, N. (2010). Boosting support vector machines for imbalanced data sets. *Knowledge & Information Systems,* 25 (1) 1-20.

[31] Xu-Ying, L., Jianxin, W., Zhi-Hua, Z. (2009). Exploratory Undersampling for Class-Imbalance Learning. *IEEE Transactions on Systems, Man, and Cybernetics*,Part B, 39 (2) 539-550.

[32] Ertekin, S., Huang, J., Bottou, L Giles, L. (2007). Learning on the border: Active learning in imbalanced data classification, *In*: Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management(CIKM 2007), p. 127-136. Lisbon, Portugal, November 6-10.

[33] Ertekin, S., Huang, J., Giles, C. L. (2007). Active learning for class imbalance problem. *In*: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval(SIGIR), 823-824, Amsterdam, Netherlands: SIGIR, July 23-27.

[34]Weiss, G. M. (2004). Mining with rarity: a unifying framework. *ACM Sigkdd Explorations Newsletter*, 6 (1) 7-19.

[35] Van Rijsbergen, C. J. (1979). Information retrieval. MA,USA: Butterworth-Heinemann Newton.

[36] Kubat, M., Holte, R. C., Matwin, S. (1998). Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, 30 (2-3) 195-215.

[37] Wang,Y. N., Ge, H. W.(2010).Improved simulated annealing genetic algorithm To solve TSP problem, *Computer Engineering and Application,* 46 (5) 44-48

[38] Gong, W., Cai, Z. (2009). Research on an å-domination based orthogonal differential evolution algorithm for multi-objective optimization. *Journal of Computer Research & Development*, 21 (1) 23-27.

[39] Pan, Z.J.,Kang, L.S.(1998). Evolutionary computation. Beijing: Tsinghua University Press.

[40] Salzberg, S. L. (1994). C4.5: programs for machine learning, *Machine Learning,* 16 (3) 235-240.

[41] Galar, M., FernaìNdez, A., Barrenechea, E., Bustince, H., Herrera, F. (2012). A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems Man & Cybernetics* Part C, 42 (4) 463-484.

[42] KEEL-dataset repository.http://www.keel.es/dataset.php

[43] Lan, H., Witten,EibeFrank(2000).Data Mining:practical Machine Learning Tools and Techniques with Java Implementations. Seattle, Wa:Morgan Kaufmann.

[44] Xiating, Feng. (2000). Introduction of Intelligent Rock mechanics. Beijing,CN:Science press.