Chiang Mai J. Sci. 2014; 41(5.2) : 1419-1428
http://epg.science.cmu.ac.th/ejournal/
Contributed Paper

Safe Level Graph for Majority Under-sampling Techniques

Chumphol Bunkhumpornpat [a] and Krung Sinapiromsaran [b]

[a] Theoretical and Empirical Research Group, Department of Computer Science, Faculty of Science, Chiang Mai University, Chiang Mai 50200, Thailand.

[b] Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University, Bangkok 10330, Thailand.

*Author for correspndence; e-mail: chumphol@chiangmai.ac.th

Received: 30 August 2012
Accepted: 12 August 2013

ABSTRACT

In classification tasks, imbalance data causes the inadequate predictive performance of a tiny minority class because the decision boundary determined by trivial classifiers tends to be biased toward a huge majority class. For handling the class imbalance problem, over- and under-sampling are applied at the data level. Over-sampling duplicates or synthesizes instances into a minority class. Although redundant instances do not harm correct classifications, they increase classification costs. Additionally, while synthetic instances expand the learning region, they are not actual instances. Under-sampling removes instances from a majority class to remedy the overlapping problem. Consequently, a downsized dataset can speed up a classification algorithm. This research investigates the behavior of several under-sampling techniques, while cleansing distinct majority class regions. We also propose a safe level graph to justify an appropriate parameter of our prior work, MUTE. The experiment shows that our decision from a safe level graph can improve the F-measure of RIPPER when evaluating minority classes.

Keywords: classification, class imbalance, under-sampling, MUTE, safe level graph, RIPPER

1. INTRODUCTION

A variety of research fields – including machine learning, data mining, and pattern recognition – employ classification [1], a technique related to prediction and forecasting. A model, or classifier, is built using characteristics of existing instances to foretell incoming instances. The method can be applied to many real-world domains; for example, diagnosing cancer, weather forecasting and predicting the likelihood of customers paying bills on time. In pre-processing for classification, a collected dataset is split into a training set for building a model and a test set for testing the model. RIPPER [2], decision tree C4.5 [3] and k nearest neighbors classifier [4] are three well-known and widely-used classifiers.

The class imbalance problem [5, 6, 7], a specific classification problem, is one of the 10 most challenging problems in data mining research [8]. A dataset is imbalance if it has a tiny number of target instances relative to other instances. The smaller class is referred to as a minority class (containing minority

instances) and the larger class is referred to as a majority class (containing majority instances).

In this paper, we aim to study the two-class case [9, 10, 11] for minority and majority classes. Although our study is designed to handle the case of a single minority class, our strategy can operate on an imbalance dataset with multiple minority classes by applying One Against All (OAA). This treats the dataset as a binary classification problem between each distinct class and the other classes; as a result, for instances in each class, a classifier is trained to predict whether the label of an unknown instance is the class. In classification tasks, an imbalance dataset leads to unsatisfactory predictive performance of trivial classifiers for a minority class because the decision boundary tends to be biased toward a huge majority class, rather than a tiny minority class. Moreover, in many applications, a minority class is labeled as noise and discarded. The objective of the problem is to improve the predictive performance of a minority class.

For handling the class imbalance problem, re-samplings [6] – or over- and under-sampling – are applied to the data level of a dataset. Over-sampling amplifies positive instances into a dataset for upsizing a minority class. Additive instances are generated based on either duplicate instances [6] or artificial instances [12]. Despite the fact that redundant instances do not harm the correct classifications of a model, they increase the classification costs for building a model. Although synthetic instances expand the learning region of a classifier, they are not actual instances in a dataset. Under-sampling wipes out some negative instances from a dataset for downsizing a majority class. As a side effect, under-sampling not only remedies the overlapping problem, but also speeds up the running time of a classification algorithm due to a number of removal instances. One study [13] applied cost curves and decision trees C4.5 with the default settings to investigate the interaction between under-sampling and over-sampling. The study found that under-sampling is superior to over-sampling when misclassification costs and class distribution are changed.

Network intrusion detection [14] is an application that encounters the class imbalance problem. Rare attacks by unauthorized users compromise computer networks, but typically represent only a very small fraction of the total network traffic. The application aims to successfully detect intrusions, which is more important than regular packages. A computer network would be seriously harmed if a model classifies an intrusion as a non-intrusion; on the other hand, a computer network remains secure if a non-intrusion is classified as an intrusion. The class imbalance problem is also encountered in telecommunications risk management [15], the detection of fraudulent telephone calls [16], in-flight helicopter gearbox fault monitoring [17], the detection of oil spills in satellite radar images [18] and the identification of likely buyers of certain products in direct marketing problems [19].

The KDD Cup 1999 dataset [20] was a competition aimed at constructing a network intrusion detector to discover intrusions of less than 2% of total connections, behaving as minority instances. Accuracy is traditionally used in the classification of balance datasets to count correctly classified instances. Accuracy, however, does not take into account the class distribution of a dataset, especially the distribution of a minority class. Accuracy would reach 98% if naive models blindly classified all connections in the KDD dataset as normal connections, despite detecting no intrusions. Accordingly, accuracy is ineffective with imbalance datasets.

This paper is divided into 5 parts. The next section briefly describes the under-sampling techniques related to our research. The third section proposes our strategy for handling the

class imbalance problem. The fourth section illustrates our experimental results. The last section discusses and summarizes our study.

2. RELATED WORK

Random under-sampling [6] is a simple and non-heuristic method that randomly eliminates instances throughout the regions of a majority class to achieve a more balance dataset.

Tomek links [11, 22] in Definition 1 connect instances that are the nearest to each other but do not belong to the same class label. The instances in pairs of Tomek links are classified as either borderline or noise and are located in an overlapping region. For under-sampling, only majority instances in Tomek links are erased, while minority instances in a dataset are retained.

Definition 1: Let instances x, y possess different class labels and $\delta(x, y)$ be the distance between them. A pair (x, y) is identified as a Tomek link if there exists no instance z such that $\delta(x, z) < \delta(x, y)$ or $\delta(y, z) < \delta(y, x)$.

Our prior work MUTE: Majority Under-sampling Technique [23] is a noise removal algorithm that utilizes the concept of k nearest neighbors to dominate minority instances by getting rid of noise majority instances, which are the obstacle to classifiers separating minority and majority classes, especially in an over-lapping region. Safe level [24] in (1) is the individual characteristic of an instance and is assigned to a majority instance to detect noise in a majority class. In addition, a majority instance is labeled as noise if its safe level is equal to k because all its k nearest neighbors are from its opposite class. But a majority instance is safer if its safe level is lower due to more of its nearest neighbors coming from the same class. In the algorithm MUTE, majority instances are deleted if they have at least τ minority instances among the k nearest neighbors, where τ is a parameter. Additionally, we set τ as k in that paper so that we could investigate the noise effect of a majority class. However, the variant setting on τ might induce classifiers to perform better.

Safe level = the number of minority instances among the k nearest neighbors.	(1)

Algorithm: MUTE(D, τ)
Input: Original dataset D and Threshold τ
Output: Under-sampling dataset D – N
Variable: Majority instance n and its Safe level sln

1. N = Ø
2. For n in D
3. if $sl_n \geq \tau$
4. N = N U {n}
5. return D

3. UNDER-SAMPLING

In this paper, we systematically separate majority instances into four sets as shown in Definition 2 by filtering the safe levels of the instances. The descriptions of the sets are as follows:

1. *NOISE* is worthless; it should not be fed into any classification algorithms.

2. *BORDERLINE* is located in an over-lapping region; it can be taken away if the region is extremely blended.

3. *SAFE* is secure; nevertheless, it can be with drawn in order to emphasize a minority class.

4. *CORE* holds the most significant informa tion in a majority class; it should be kept after the under-sampling process is done.

Definition 2: Let D be a dataset, and sl_n be a safe level of a majority instance n. Sets of majority class are defined as follows:

$$NOISE = \{n \in D \mid sl_n = k\}$$

$$BORDERLINE = \{n \in D \mid \tfrac{1}{2}k \le sl_n < k\}$$

$$SAFE = \{n \in D \mid 0 < sl_n < \tfrac{1}{2}k\}$$

$$CORE = \{n \in D \mid sl_n = 0\}$$

The rest of this paper, noise, borderline, safe, and core instances are referred to members of *NOISE, BORDERLINE, SAFE,* and *CORE*, respectively. A noise instance is orbited by all nearest neighbors from a minority class. A borderline instance has most nearest neighbors from a minority class. A safe instance has most nearest neighbors from a majority class. A core instance is orbited by all nearest neighbors from a majority class. The locations of members of the four sets nearby an over-lapping region are illustrated in Figure 1. The symbols ● represent minority instances and the letters *n, b, s* and *c* represent the noise, borderline, safe and core instances, respectively.

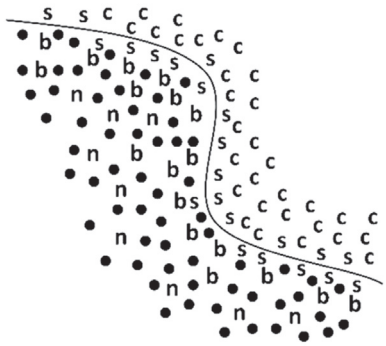

Figure 1 Noise, borderline, safe and core.

In the last section, we recall our prior specific MUTE, with τ set to k, to be referred to in this paper as MUTE(*D, k*). In this section, we propose two new specific MUTEs, where we set τ as $\frac{1}{2}k$ and 1. The functionalities of all MUTEs are described as follows:

1. MUTE(*D, k*) cleans only *NOISE* from an imbalance dataset.
2. MUTE(*D,* $\frac{1}{2}k$) deletes all instances in both *NOISE* and *BORDERLINE*.
3. MUTE(D, 1) erases *NOISE*, *BORDERLINE* and *SAFE* but holds *CORE* in a majority class.

Figure 2-4 exhibit the outputs of the three MUTEs by under-sampling the region in Figure 1. The more the value of τ decreases, the more the number of removal majority instances increases towards the core of a majority class.

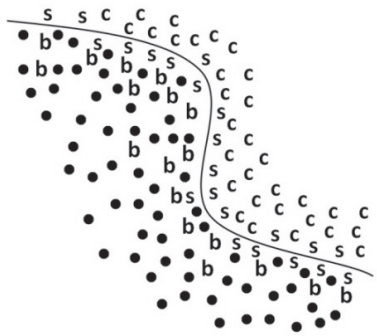

Figure 2. MUTE(*D, k*).

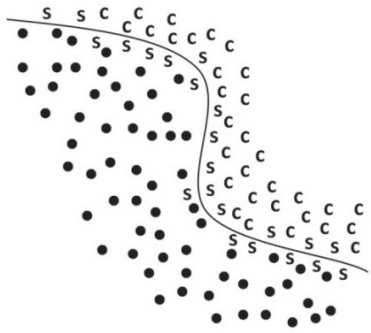

Figure 3. MUTE(D, $\frac{1}{2}$k).

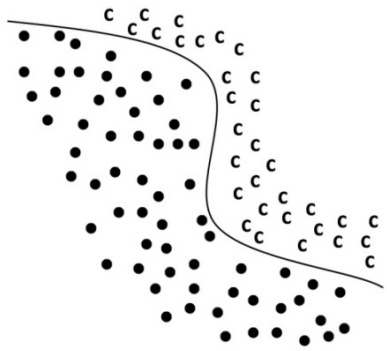

Figure 4. MUTE(D, 1).

Unfortunately, tuning parameters in a machine learning algorithm can be difficult and time consuming. In this section, we also provide a heuristic guideline called a safe level graph, where the x-axis represents the safe level and the y-axis the percentage of a minority instance. A safe level graph is used to determine an appropriate value of τ. In contrast to the interpretation of the safe levels of majority instances, minority instances are denser and sparser, if their safe levels are higher and lower, respectively. We categorize the curve of safe level graphs into three distribution shapes: *Skewed to the left, Symmetric bell shaped* and *Skewed to the right*. These are illustrated in Figure 5-7, respectively. Guidelines for interpreting τ using the safe level graph follow:

1. If *Skewed to the left*, most minority instances are dense, forming the core of a minority class. In this case, MUTE(*D, k*) should be run.
2. If *Symmetric bell shaped*, more of the minority instances are spread nearby the border of a minority class. In this case, MUTE(*D*, ½*k*) should be run.
3. *If Skewed to the right*, most minority instances skew into a majority class. In this case, MUTE(*D*, 1) should be run.

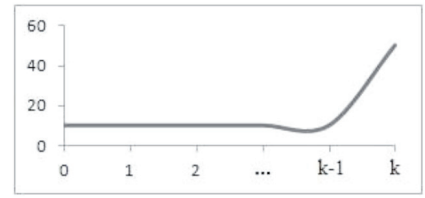

Figure 5. Skewed to the left.

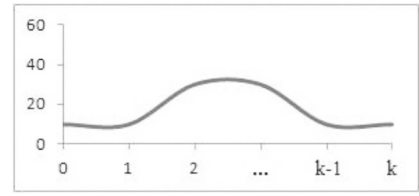

Figure 6. Symmetric bell shaped.

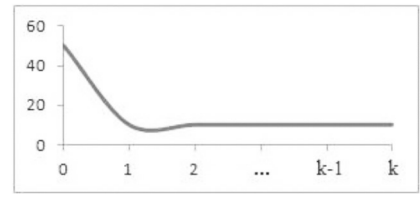

Figure 7. Skewed to the right.

4. MATERIALS AND METHODS

In this section, we investigate the behavior of pure under-sampling methods for cleansing imbalance datasets. The experiment was designed to support our conjecture that we need various MUTEs to operate on different imbalance datasets; moreover, we need a guideline for determining an appropriate MUTE. Our experiment design is explained as follows:

1. We applied three versions of MUTEs compared to a non-technique, the simplest technique and the state of the art under-sampling. We used 50% random under-sampling, which downsizes a majority class by half, as the representative of overall random under-samplings. For the computation of k nearest neighbors and

safe level in MUTE, parameters k are set to 5 according to the default values of several re-sampling techniques [12, 23, 24, 25, 26]. The list of all experimental methods is as follows:

- Random under-sampling
- Tomek links
- MUTE(D, k)
- MUTE(D, ½k)
- MUTE(D, 1)

2. We collected datasets with various degrees of imbalance from the UCI Repository of Machine Learning Databases [27] as detailed in Table 1.

3. We used RIPPER as an experimental classifier. RIPPER is a rule-based classifier, which extracts the conditions in the form of IF and THEN, and is similar to tree-based classifiers such as C4.5. The rule-based and tree-based classifiers can be converted to each other; however, RIPPER's more simplified rules offer advantages.

4. In evaluation process, we chose F-measure [21] as a performance measure. F-measure is appropriate for the class imbalance problem, because F-measure concentrates on a minority rather than majority class, and F-measure is large when classification rates of both classes are large.

5. For each dataset, a target class is chosen to be a minority class and the remaining classes are combined as a single majority class. In addition, the dataset is split into training and test sets, using 2/3 and 1/3 of the dataset, respectively, except Statlog (Landsat Satellite), which had already been separated by the dataset provider. We prepared three independent training sets to calculate the median of F-measure in order to prevent the randomness of the technique.

6. We selected paired t-test as the statistical analysis. The test is provided at the end of this section

Table 2 shows the experimental result, in which ORG, RND and TMK represent an original dataset, 50% random under-sampling, and Tomek links, respectively. The MUTEs with underlined results were guided from their safe level graphs, which are illustrated in Figure 8-13.

From the experiment, most imbalance datasets are skewed to the left. It means that minority instances frequently are dense.

Guided MUTEs acquire superior F-measure on most experimental datasets, except Letter Recognition. The methods constantly improve F-measure of all datasets.

Random under-sampling achieves the best F-measure on Pima Indians Diabetes, which is the least imbalance ratio among all datasets. Unfortunately, the method occasionally shows no improvement or even drops the performance of the classifier.

Tomek links frequently drops F-measure, especially less imbalance Haberman's Survival with its inferior F-measure. However, the method obtains the best F-measure on highly imbalance Letter Recognition.

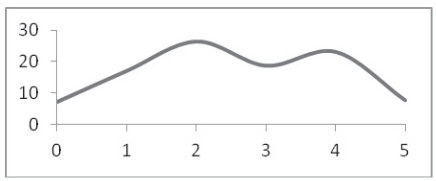

Figure 8. Pima Indians diabetes.

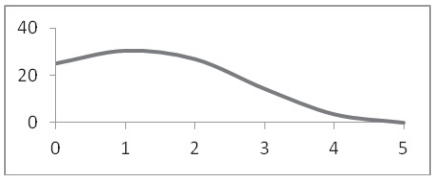

Figure 9. Haberman's survival.

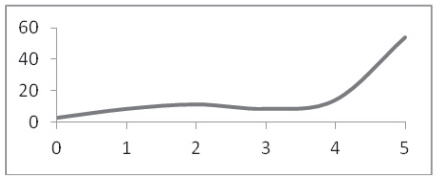

Figure 10. Glass identification.

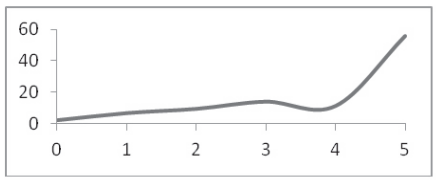

Figure 11. Image segmentation.

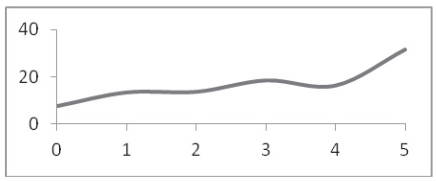

Figure 12. Statlog (Landsat satellite).

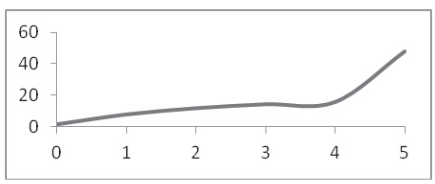

Figure 13. Letter recognition.

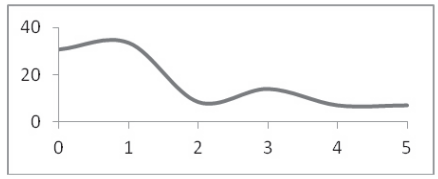

Figure 14. Page blocks classification.

We run paired t-test on the experimental result in Table 2. The null and alternative hypotheses are as follows:

H_0: $\mu_1 - \mu_2 = 0$

H_1: $\mu_1 - \mu_2 > 0$

From the hypothesis, μ_1 is the mean of guided MUTE as an indicator variable and μ_2 is the mean of an original dataset, random under-sampling or Tomek links. The paired t-test is shown in Table 3, in which symbols are described as follows:

- *df* is a degree of freedom.
- *t* is a t-statistic value.
- *p* is a two-tailed probability value.

We set significance level (α) as 0.10. The means are different across the paired observations if *p* is less than α, which is interpreted as H_0 is rejected.

Table 1. Dataset.

Dataset	Instance	Column	Minority Class	Minority Percentage
Pima Indians Diabetes	768	8	Positive	34.89
Haberman's Survival	306	3	Died	26.47
Glass Identification	214	10	Non-window	23.83
Image Segmentation	2,310	19	Window	14.29
Statlog (Landsat Satellite)	6,435	36	Damp Gray Soil	9.73
Letter Recognition	20,000	16	H	3.67
Page Blocks Classification	5,473	10	Picture	2.10

Table 2. F-measure.

Dataset	ORG	RND	TMK	MUTE (D, k)	MUTE (D, ½k)	MUTE (D, 1)
Pima Indians Diabetes	.581	**.657**	.597	.569	**.657**	.603
Haberman's Survival	.408	.408	.238	.408	.408	**.410**
Glass Identification	.824	.824	.759	**.909**	.824	.824
Image Segmentation	.828	.794	.824	**.835**	.809	.793
Statlog (Landsat Satellite)	.584	.609	.585	**.646**	.625	.600
Letter Recognition	.713	.722	**.762**	.736	.756	.711
Page Blocks Classification	.701	.578	.632	.705	**.736**	.688

Table 3. Paired t-test.

Comparison	$\mu_1 - \mu_2$	*df*	*t*	*p*
MUTE - ORG	0.041428	6	3.295454	**0.016501**
MUTE - RND	0.047714	6	1.943180	**0.067466**
MUTE - TMK	0.076000	6	2.816286	**0.030502**

5. Conclusion

In this paper, we propose the three new versions of MUTEs that can be used for various imbalance datasets; moreover, we also design a safe level graph that can successfully guide MUTE.

The reasons that a safe level graph can guide an appropriate MUTE are that:

- If most minority instances are dense, the classifier will probably detect a minority class, so it is unnecessary to clean a lot of majority instances. In this case, we clean only *NOISE* to avoid the noise effect on a minority class.
- If most minority instances are spread in an over-lapping region in which *NOISE* and *BORDERLINE* are also located, minority and majority instances would be extremely blended. In this case, we delete the two sets to provide better separation between the two classes.
- If most minority instances are skew into a majority class, it is difficult for the classifier to recognize a minority class. In this case, we need to erase as much as majority instances by keeping only *CORE* to more dominate a minority class.

Random under-sampling and Tomek links fail to boost the predictive performance of a minority class due to the following facts:

- For Random under-sampling, due to the removal instances, especially in the core region of the class, a dataset's important information might be lost and the predictive performance of classifiers might be dropped.
- For Tomek links, deleting only nearest neighbors from a majority class is not enough to dominate a minority class.

The experiment reveals that the guided MUTE by a safe level graph significantly improves F-measure of RIPPER according to the paired t-test in most cases. However, random under-sampling and Tomek links produce unsatisfactory results. So, we can summarize that our safe level graph is an efficiency tool for class imbalance problem.

For future work, a re-defined safe level could be substituted for the current one. We could also consider the dual safe level graph of both minority and majority classes before estimating the parameters of MUTE. With this strategy, we expect to further improve the predictive performance using imbalance datasets.

References

1. Han J., Kamber M. and Pei J., *Data Mining: Concepts and Techniques*, 3rd Edn., Morgan Kaufmann, 2011.

2. Cohen W.W., Fast Effective Rule Induction, *Proceedings of the 12th International Conference on Machine Learning*, Lake Tahoe, USA, 1995; 115-123.

3. Quinlan J.R., *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1992.

4. Cover T. and Hart P.E., Nearest neighbor pattern classification, *IEEE Trans. Inf. Theory*, 1967; **13(1)**: 21-27.

5. Japkowicz N., Class Imbalance: Are We Focusing on the Right Issue?, *Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Data Sets II*, Washington, DC, USA, 2003; 17–23.
6. Japkowicz N., The Class Imbalance Problem: Significance and Strategies, *Proceedings of the 2000 International Conference on Artificial Intelligence*, Las Vegas, USA, 2000; 111-117.
7. Chawla N.V., Japkowicz N. and Kolcz A., Editorial: special issue on learning from imbalanced data sets, *SIGKDD Explor.*, 2004; **6(1)**: 1-6.
8. Yang Q. and Wu X., 10 Challenging problems in data mining research, *Int. J. Inf. Technol. & Dec. Mak.*, 2006; **5(4)**: 597-604.
9. Kubat M., Holte R. and Matwin S., Learning When Negative Examples Abound, *Proceedings of the 9th European Conference on Machine Learning*, Prague, Czech Republic, 1997; 146-153.
10. Kubat M. and Matwin S., Addressing the Curse of Imbalanced Training Sets: One-Sided Selection, *Proceedings of the 14th International Conference on Machine Learning*, Nashville, USA, 1997; 179-186.
11. Tomek I., Two modifications of CNN, *IEEE Trans. Sys., Man Cyb.*, 1976; **6(11)**: 769-772.
12. Chawla N.V., Bowyer K.W., Hall L.O. and Kegelmeyer W.P., SMOTE: Synthetic minority over-sampling technique, *J. Artif. Intell. Res.*, 2002; **16**: 341-378.
13. Drummond C. and Holte R.C., C4.5, Class Imbalance, and Cost Sensitivity Why Under-Sampling Beats Over-Sampling, *Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Data Sets II*, Washington, DC, USA, 2003; 1-8.
14. Khor K.C., Ting C.Y. and Phon-Amnuaisuk S., A cascaded classifier approach for improving detection rates on rare attack categories in network intrusion detection, *Appl. Intell.*, 2012; **36(2)**: 320-329.
15. Ezawa K., Singh M. and Norton S., Learning Goal Oriented Bayesian Networks for Telecommunications Risk Management, *Proceedings of the 13th International Conference on Machine Learning*, Bari, Italy, 1996; 139-147.
16. Fawcett T. and Provost F., Combining Data Mining and Machine Learning for Effective User Profile, *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, Portland, USA, 1996; 8-13.
17. Japkowicz N., Myers C. and Gluck M., A Novelty Detection Approach to Classification, *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Montreal, Canada, 1995; 518-523.
18. Kubat M., Holte R. and Matwin S., Machine learning for the detection of oil spills in satellite radar images, *Mach. Learn.*, 1998; **30**: 195-215.
19. Ling C. and Li C., Data Mining for Direct Marketing Problems and Solutions, *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, New York, USA, 1998; 73-79.
20. KDD Cup – acm; sigkdd Available at: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
21. Buckland M. and Gey F., The relationship between recall and precision, *J. Am. Soc. Info. Sci.*, 1994; **45(1)**: 12-19.
22. Batista G.E.A.P.A., Prati R.C. and M.C. Monard, A study of the behavior of

several methods for balancing machine learning training data, *SIGKDD Explor.*, 2004; **6(1)**: 20-29.

23. Bunkhumpornpat C., Sinapiromsaran K. and Lursinsap C., MUTE: Majority Under -sampling Technique, *Proceedings of the 8th International Conference on Information, Communications, and Signal Processing*, Singapore, 2011; 1-4.

24. Bunkhumpornpat C., Sinapiromsaran K. and Lursinsap C., Safe-Level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem, *Lect. Notes Artif. Intell.*, 2009; **5476**: 475-482.

25. Bunkhumpornpat C., Sinapiromsaran K. and Lursinsap C., DBSMOTE: Density -based synthetic minority over-sampling technique, *Appl. Intell.*, 2012; **36(3)**: 664-684.

26. Han H., Wang W.Y. and Mao B.H., Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning, *Lect. Notes Comput. Sci.*, 2005; **3644**: 878-887.

27. Blake C.L. and Merz C.J., UCI Repository of Machine Learning Databases; Available at: http://archive.ics.uci.edu/ml/. Accessed 2012.