2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

An Under-sampling Method Based on Fuzzy Logic for Large Imbalanced Dataset

Ginny Y. Wong *Student Member, IEEE*, Frank H.F. Leung, *Senior Member, IEEE*,
and Sai-Ho Ling, *Senior Member, IEEE*

***Abstract*—Large imbalanced datasets have introduced difficulties to classification problems. They cause a high error rate of the minority class samples and a long training time of the classification model. Therefore, re-sampling and data size reduction have become important steps to pre-process the data. In this paper, a sampling strategy over a large imbalanced dataset is proposed, in which the samples of the larger class are selected based on fuzzy logic. To further reduce the data size, the evolutionary computational method of CHC is employed. The evaluation is done by applying a Support Vector Machine (SVM) to train a classification model from the re-sampled training sets. From experimental results, it can be seen that our proposed method improves both the F-measure and AUC. The complexity of the classification model is also compared. It is found that our proposed method is superior to all other compared methods.**

I. Introduction

The classification of imbalanced datasets is a popular research topic. Most of the machine learning tools, such as neural network and support vector machines, are originally designed for well-balanced datasets to minimize the global error rate [1]. If the dataset is imbalanced, the samples are biased to the majority class. However, the minority class dataset is usually more important and more meaningful. For example, there are much less samples of people with a particular disease than those of healthy people in a medical problem. If a classifier is needed to label whether the people are infected or not, it is obvious that the minority class (people with a particular disease) is the more interested class.

Imbalanced datasets are commonly found in many applications, such as detection of oil spills from satellite images [2], spotting customers for telecommunications management [3], and identification of power distribution fault causes [4]. There are two main approaches to solve the problems caused by imbalanced datasets. One is the data level approach and the other is the algorithm level approach. Data level approaches [5]–[7] include balancing the class distribution by over-sampling the minority class or under-sampling the majority class. The solutions of algorithm level approaches improve the existing machine learning methods by adjusting the probabilistic estimate [8], modifying the cost per class [9], adding some penalty constants [10], or learning from one class instead of two classes [11].

Many experiments [12] show that preprocessing is a good data level approach to handle the imbalanced data. Moreover, preprocessing approaches are more flexible since they are independent of the chosen classifier. Therefore, we focus on re-sampling the class distribution in this paper. There are three main types of strategies for re-sampling data. The first one is over-sampling, which can be done randomly or by the method of Synthetic Minority Over-sampling Technique (SMOTE) [6]. The second one is under-sampling, which include the Tomek links [13] and Neighborhood Cleaning Rule(NCL) [14]. The last one is the hybrid method, which combines the two previous methods (over-sampling and under-sampling methods). Although over-sampling and hybrid methods outperform the under-sampling methods [12], under-sampling can produce less samples to deal with a large dataset.

An under-sampling method is proposed in this paper. The samples of the majority class are chosen based on fuzzy logic, which can be a useful tool to treat imbalanced datasets [12]. To further reduce the data size, an evolutionary algorithm (EA) is applied. The chosen EA is the CHC algorithm [15] (Cross-generational elitist selection, Heterogeneous recombination and Cataclysmic mutation) since it shows the ability of selecting the smallest and most representative instances among many algorithms studied in [16].

Some experiments are carried out to show the improved performance of our proposed method against other methods, which include random under-sampling (RUS), condensed nearest neighbor rule (CNN) [17], Tomek Links (TL) [13], one-sided selection (OSS) [18], and neighborhood cleaning rule (NCL) [14]. A large imbalanced dataset from UCI Repository [19] is used as the dataset. The Support Vector Machine (SVM) [20] is used as the tool for reaching a classification model from each re-sampled dataset, so as to evaluate the corresponding preprocessing method. The evaluation methods are based on the functions of precision and recall.

This paper is organized as follows: In Section II, some preprocessing methods and CHC are described. Section III introduces the details of the proposed sampling strategy and the evaluation method of this study. To show the effectiveness of our proposed method, the results and comparisons are discussed in Section IV. A conclusion is drawn in Section V.

II. Previous Work

This section describes some previous works about under-sampling methods, which will be compared with our proposed method in the experiments. The ideas about CHC will also be discussed.

Ginny Y. Wong is with the Centre for Signal Processing, Dept. of Electronic and Information Engg., The Hong Kong Polytechnic University, Hung Hom, Hong Kong. (e-mail: ginnyyk.wong@connect.polyu.hk).

Frank H.F. Leung is with the Centre for Signal Processing, Dept. of Electronic and Information Engg., The Hong Kong Polytechnic University, Hung Hom, Hong Kong. (e-mail: frank-h-f.leung@polyu.edu.hk).

Sai-Ho Ling is with the Centre for Health Technologies, Faculty of Engineering and Information Technology, University of Technology Sydney, NSW, Australia. (e-mail: steve.ling@uts.edu.au).

978-1-4799-2072-3/14/$31.00 ©2014 IEEE

A. Under-sampling Methods

Some instances of majority class are eliminated in order to balance the class distribution.

Random under-sampling(RUS) is a non-heuristic method that aims to balance the datasets by randomly removing some samples of the majority class. This method may easily remove some useful data.

Condensed nearest neighbor rule(CNN) [17] eliminates the majority class samples that are distant from the decision border since these samples can be considered as less relevant for learning. First, a majority class sample is randomly drawn and formed a subset with all the minority class samples. Then, 1-NN is used over this subset to classify the other majority class samples. Every misclassified majority samples are selected to form the re-sampled dataset.

Tomek links(TL) [13] is opposite to CNN. It edits out noisy and borderline majority class samples. Borderline samples can be treated as unsafe samples since only small changes can cause them to be assigned to a wrong class. The process can be described as follows. First, each sample is used to find another sample which has the minimum distance between them. If these two samples are in different classes, the sample of majority class will be removed. This method can increase the area of decision border. However, some useful data, which is important for the classification, may also be discarded.

One-sided selection(OSS) [18] applies TL followed by CNN. Taking advantages of those two methods, the remainder majority samples are safe and more relevant for learning.

Neighborhood Cleaning Rule(NCL) [14] uses the Wilson's Edited Nearest Neighbor Rule(ENN) to remove some majority class samples. First, three nearest neighbors of each sample in the training set are found. If the selected sample belongs to the majority class but the three nearest neighbors classify it wrongly, the selected sample will be removed. If the selected sample belongs to the minority class but the three nearest neighbors classify it wrongly, the nearest neighbors belonging to the majority class will be removed.

B. CHC [15]

CHC is a kind of EAs that combines a selection strategy with a highly disruptive recombination operator. To avoid premature convergence and maintain diversity, incest prevention and cataclysmic mutation are introduced. The process of CHC can be described as follows. Firstly, a population set of chromosomes P is created. Each chromosome $p_i = (p_{i1}, p_{i2}, \ldots, p_{in})$ is an n-dimensional vector, which is a set of genes, where p_{ij} is the jth gene value ($j = 1, 2, \ldots, n$) of the ith chromosome in the population ($i = 1, 2, \ldots, m$); m is the population size and n is the number of genes.

Secondly, the chromosomes are evaluated by a defined fitness function. The form of fitness function depends on the application. Thirdly, an intermediate population set of chromosomes C, which is of the same size as P is generated by copying all members of P in a random order.

Then, a uniform crossover (HUX) operator is applied on C to form C'. HUX exchanges half of the genes randomly between the parents. CHC also uses an additional method for incest prevention. Before applying HUX to the parents, the Hamming distance between them is calculated. If half of that distance is larger than a difference threshold d, HUX is applied; otherwise these two parents are deleted from C. The initial threshold d is set at $n/4$. After C' has formed, it is evaluated by the fitness function and an elitist selection is taken. Only the best chromosomes from both P and C' are selected to form the offspring population in the next generation. If the offspring population is the same as P, the difference threshold d is decreased by one.

CHC is different from the traditional genetic algorithm. Mutation is not performed at the recombination stage. CHC performs partial reinitialization (divergence) when the search becomes trapped (i.e., the difference threshold d becomes zero and no new offspring population is formed for several generations). The population is reinitialized, based on the best chromosome, by changing the elements' values randomly with a user-defined divergence rate D_{rate}. For example, if D_{rate} = 0.35, the values of 35% elements will be changed randomly. The search is then resumed with a new difference threshold $d = D_{rate} * (1 - D_{rate}) * n$. This process is called cataclysmic mutation.

CHC has shown its ability of selecting the smallest and most representative instances among the other algorithms studied in [16]. Therefore, it is chosen as the algorithm to reduce the size of dataset.

III. Methodology

In this section, the proposed under-sampling method and the evaluation method used in this paper are discussed. The proposed under-sampling method involves two stages. The majority class samples of the training sets are firstly under-sampled based on fuzzy logic. To further reduce the size of dataset, CHC is then implemented to both minority and majority class samples.

A. Fuzzy Set

In this paper, fuzzy logic is used to cluster the majority class samples and select the samples depending on their importance.

Let the class $negative$ be the majority class and only m training samples (X_p) of the class negative are considered, where $X_p = (x_{p1}, \ldots, x_{pn})$ is an n-dimensional vector, $p = 1, 2, \ldots, m$ and x_{pi} is the ith attribute value ($i = 1, 2, \ldots, n$) of the pth training sample. The jth fuzzy if-then rule is written as follows:

$$\text{Rule } j : \text{IF } z_1 \text{ is } A_1^j \text{ AND } \ldots \text{ AND } z_n \text{ is } A_n^j \\ \text{THEN class = negative with } w_j, \qquad (1)$$

where A_α^j is a fuzzy term of the jth rule corresponding to the attribute z_α, $\alpha = (1, 2, \ldots, n)$, $z = (z_1, z_2, \ldots, z_n)$ is an n-dimensional attribute vector, and w_j is the rule weight. The Gaussian membership functions are used as antecedent fuzzy sets, which are formed based on the distribution of the attributes. A rule base is formed by the training samples of negative class. The corresponding label of each attribute has the highest membership value among the other labels. This label is selected and the rule is formed. The maximum

TABLE I: Label Setting of Each Membership Function of the ith Attributes.

Label	m_{ik}	σ_{ik}
1	Area($\frac{1}{L+1}$)	$\frac{stdev_i*(L+1)/2}{L}$
2	Area($\frac{2}{L+1}$)	$\frac{stdev_i*(L-1)/2}{L}$
$\vdots$	$\vdots$	$\vdots$
$\frac{L+1}{2}$	$mean_i$	$\frac{stdev_i}{L}$
$\vdots$	$\vdots$	$\vdots$
L-1	Area($\frac{L-1}{L+1}$)	$\frac{stdev_i*(L-1)/2}{L}$
L	Area($\frac{L}{L+1}$)	$\frac{stdev_i*(L+1)/2}{L}$
Note: Area($\frac{1}{L+1}$) means the samples smaller than that m_{ik} have occupied $\frac{1}{L+1}$ number of samples.		

number of rules depends on the number of labels and attributes and equals to L^n, where L is the number of labels. The arrangement of each label of the membership functions and the way of finding the values of rule weights are introduced in the following subsections.

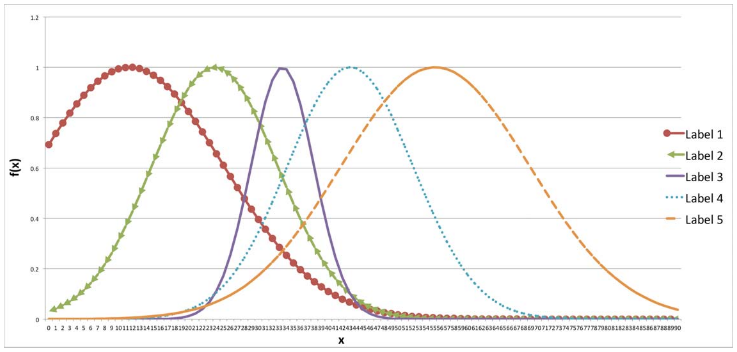

Fig. 1: Arrangement of the membership function of each label. 5 labels are employed as an example.

1) Membership Functions: When deciding the membership function of each label, the distribution of the attribute is considered. First, the mean value ($mean_i$) and standard deviation ($stdev_i$) of the ith attribute are calculated, where $i = 1, 2, \dots, n$. The samples closer to the mean value are treated as more informative. Therefore, the membership function near the mean value is assigned with a narrower "bell" to cluster the samples. An odd number should be used as the number of labels. Consider L labels per attribute are employed and the Gaussian membership function of label k ($k = 1, 2, \dots, L$) is defined as follows:

$$f_k(x_{pi}) = e^{-\frac{(x_{pi}-m_{ik})^2}{2\sigma_{ik}}}, \tag{2}$$

where m_{ik} and σ_{ik} is the mean and standard deviation of the kth label corresponding to the attribute i respectively. Both m_{ik} and σ_{ik} are assigned based on $mean_i$ and $stdev_i$. Table I shows the methods of setting the parameters of each membership function. An example of 5 labels is shown in Fig. 1. This setting of membership functions can cluster the samples near the mean value more significantly.

2) Rule Weight: The rule weight w_j is used to reflect the degree of matching of each fuzzy rule over all the negative samples, so that the importance of each rule can be evaluated. First, the fuzzy value of each sample is calculated. The fuzzy value of X_p for the jth fuzzy rule is defined as follows:

$$\mu_{A^j}(X_p) = T(\mu_{A_1^j}(x_{p1}), \dots, \mu_{A_n^j}(x_{pn})), \tag{3}$$

where the product T-norm is used. The rule weight (w_j) is calculated by adding all the fuzzy values of each sample.

$$w_j = \sum_{p=1}^{m}(\mu_{A^j}(X_p)). \tag{4}$$

3) Selection of the Majority Samples: After the rule base of the class negative is generated, the rules are randomly drawn based on the rule weight. The rule with a higher rule weight will have a higher probability to be chosen. Then, the sample matching this rule is selected randomly to form the new dataset. These processes are repeated until the number of negative samples is twice of positive samples.

B. Setting of CHC

After the under-sampling, the number of majority class samples is twice of the minority class samples, and CHC is then applied. There are two important issues that need to be addressed clearly before the algorithm is employed: the representation of each chromosome and the definition of fitness function.

1) Chromosome Representation: CHC is used to further reduce the data size. Therefore, the chromosomes are to represent subsets of these samples. It can be carried out by a binary representation. Each chromosome is an n_g-dimensional vector, which is a set of genes, where n_g is the number of genes. In this study, n_g is the number of samples in the training set. Each gene shows whether the corresponding sample exists in the subset of the training set or not. Therefore, there are two possible values for each gene: 0 and 1. If the gene value is 1, the corresponding sample is included in the subset of the training set. If the gene value is 0, the sample does not exist in the subset.

2) Fitness function: In this study, the SVM is used as the evaluation method of CHC to obtain the subset with the highest classification rate. Normally, accuracy (ratio of correctly classified samples to total number of samples) would be used as the measure of classification rate. However, it may cause difficulty for imbalanced datasets since the correct classification rate of the majority class samples may affect the accuracy more seriously than that of the minority class. This problem is more obvious if the ratio of the number of majority class to that of minority class is large. The worst case could be that even all the minority class samples are misclassified, the accuracy can still be very high. Therefore, some other measures are used in this paper. These measures are commonly employed to analyze problems with imbalanced datasets.

Firstly, precision and recall are introduced [21]. Their definitions are given as follows:

$$Precision = \frac{TP}{TP+FP} \tag{5}$$

$$Recall = \frac{TP}{TP+FN} \tag{6}$$

where TP is the number of true positives, FP is the number of false positives and FN is the number of false negatives. A high value of precision indicates that the predicted positive samples are most likely relevant. A high value of recall indicates that most of the positive samples can be predicted correctly.

Another measure is $F-measure$ [21], which is a function of precision and recall. It is a popular evaluation metric for imbalanced problems. In principle, $F-measure$ represents a harmonic mean between precision and recall. A high value of $F-measure$ means both the precision and recall values are high and do not differ very much. It is defined as follows:

$$F-measure = \frac{2 * precision * recall}{precision + recall} \tag{7}$$

The area under the receiver operating characteristic curve (AUC) is also commonly used to measure the performance of classification. The AUC measure [22] is the probability of correctly identifying a random sample, and it can be defined as follows:

$$AUC = \frac{1 + Recall - FP_{rate}}{2} \tag{8}$$

where $Recall$ is defined in (6) and $FP_{rate} = \frac{FP}{FP+TN}$, TN is the number of true negatives. FP_{rate} defines the percentage of true negatives cases misclassified as positives. A high value of AUC implies small values of FN and FP, meaning that the corresponding classifier is very effective.

Since both $F-measure$ and AUC are important measures on imbalanced datasets, a multi-objective fitness function is used here. If a chromosome X has a higher value of $F-measure$ ($F_X > F_Y$) and a lower value of AUC ($A_X < A_Y$) than that of chromosome Y, the difference between the chromosomes' $F-measure$ ($|F_X - F_Y|$) and the difference between the chromosomes' AUC ($|A_X - A_Y|$) will be compared. If $|F_X - F_Y| > |A_X - A_Y|$, chromosome X will be regarded as a better one; otherwise chromosome Y will be regarded as a better one.

IV. Experimental Study

In this section, we present the experiments that are carried out to compare our proposed method with other under-sampling methods. The dataset used can be found in UCI Repository [19].

The experiments involve RUS, CNN, TL, OSS, NCL, and our proposed method. To measure the performance of the preprocessing method, the same learning tool should be used among all the experimental methods. This tool is the Support Vector Machine (SVM) that attempts to obtain the classification model from the re-sampled training set. The program of all testing methods and the learning tool are based on KEEL, which is an open source software available in the Web [23]. $F-measure$ and AUC are used as measures to analyze the results of the experimental methods.

As mentioned before, the large re-sampled training datasets will increase the complexity of the classification model. Therefore, the under-sampling rate and the number of support vectors formed from SVM will also be compared.

TABLE II: Descriptions of the Selected Imbalanced Dataset.

Dataset	$N_{samp.}$	$N_{attr.}$	**Min., Maj.(%)**	**IR**
Census (Training/Testing)	57,008/57,008	41	(5.73, 94.27)	16.45
Census (Validation)	28,504	41	(5.73, 94.27)	16.45

A. Datasets

To evaluate the methods, a large dataset called Census from UCI is chosen. It has been divided into five parts evenly. The training set and testing set form two parts of them separately. The remainder part forms the validation set. Table II shows the details of the selected dataset, where the number of samples ($N_{samp.}$), the number of attributes ($N_{attr.}$), the distribution of minority and majority classes, and the imbalanced ratio (IR) can be found. IR is the ratio of the number of majority class to the number of minority class. When IR is larger, a larger difference between these two classes is represented.

B. Setup of Experiment

For CHC, the basic setting of the parameters are:

- Population size: 30.
- Divergence rate: 0.35.
- Threshold decreasing rate: 0.001.
- Kernel of SVM: Radial Basis Function.
- Number of Evaluations: 2,000.

In this paper, SVM is used to weigh the influence of each preprocessing methods. A radial basis function (RBF) is used as the SVM kernel since a non-linear classification model is needed and RBF is a common kernel to handle this problem. The RBF is defined as follows:

$$RBF = exp(-\frac{1}{\sigma}\|\mathbf{x}_i - \mathbf{x}\|^2) \tag{9}$$

where $\sigma > 0$ is the parameter to determine the width of the radial basis function. It controls the flexibility of the classifier. When σ decreases, the flexibility of the resulting classifier in fitting the training data increases, and this might lead to overfitting easily. The value of σ is set as 0.01 for the experiments.

C. Results

Table III shows the $F-measure$ and AUC of each sampling method. The proposed method can offer the best values of $F-measure$ and AUC. The performance of TL and NCL is similar since the ideas of them are similar to remove the noisy and borderline samples. The under-sampling rate and the support vectors' number of the classification model of different methods are also shown in the table. The under-sampling rate is defined as follows:

$$Rate_{under} = \frac{(N_{original} - N_{sampled})}{N_{original}} * 100\% \tag{10}$$

where $N_{sampled}$ is the number of samples in the re-sampled training set and $N_{original}$ is the number of samples in the original training set. The proposed method can obtain the

TABLE III: The Testing Results of Census.

Results	RUS	CNN	TL	OSS	NCL	Proposed Method
F-measure	0.1579	0.05753	0.02333	0.07913	0.02578	0.1702
AUC	0.6703	0.5095	0.5043	0.5163	0.5046	0.6869
Under-sampling Rate	0.8855	0.8455	0.05334	0.8592	0.1228	0.9083
Number of Support Vectors	6,396	8,799	40,083	8,024	36,690	4,381

highest under-sampling rate and the lowest number of support vectors. This shows that our method can use less training samples to achieve high performance and the classification model is simpler to apply.

V. Conclusion

An under-sampling method over large imbalanced datasets has been proposed. The samples of the majority class are selected based on fuzzy logic. CHC is used to further reduce the data size. The proposed method is compared to RUS, CNN, TL, OSS, and NCL. To evaluate the performance of these six sampling methods, the same SVM classifier has been used to obtain the experimental results. It shows that our method outperforms the other sampling methods on both $F-measure$ and AUC. The large data size may increase the computational power of the classification. Therefore, the under-sampling rate and the number of support vectors of the classification model are also compared. Our method achieves good results on all these measures, which means the proposed method can select the most representative samples to form the training sets.

Acknowledgment

The work described in this paper was substantially supported by a grant from The Hong Kong Polytechnic University (Project No. RPKP).

References

[1] N. Japkowicz and S. Stephen, "The class imbalance problem: a systematic study," *Intelligent Data Analysis*, vol. 6, no. 5, pp. 429–450, 2002.

[2] M. Kubat, R. Holte, and S. Matwin, "Machine learning for the detection of oil spills in satellite radar images," *Machine Learning*, vol. 30, no. 2–3, pp. 195–215, 1998.

[3] K. Ezawa, M. Singh, and S. Norton, "Learning goal oriented bayesian networks for telecommunications risk management," in *Processdings of the International Conference on Machine Learning*. Bari, Italy: Morgan Kauffmann, 1996, pp. 139–147.

[4] L. Xu, M. Chow, and L. Taylor, "Power distribution fault cause identification with imbalanced data using the data mining-based fuzzy classification e-algorithm," *IEEE Transactions on Power Systems*, vol. 22, no. 1, pp. 164–171, 2007.

[5] G. Batista, R. Prati, and M. Monard, "A study of the behavior of several methods for balancing machine learning training data," *SIGKDD Explorations*, vol. 6, no. 1, pp. 20–29, 2004.

[6] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal Of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[7] H. Guo and H. Viktor, "Learning from imbalanced data sets with boosting and data generation: the databoost-im approach," *SIGKDD Explorations*, vol. 6, no. 1, pp. 30–39, 2004.

[8] G. Weiss and F. Provost, "Learning when training data are costly: the effect of class distribution on tree induction," *Journal of Artificial Intelligence Research*, vol. 19, pp. 315–354, 2003.

[9] F. Provost and T. Fawcett, "Robust classification for imprecise environments," *Machine Learning*, vol. 42, no. 3, pp. 203–231, 2001.

[10] Y. Lin, Y. Lee, and G. Wahba, "Support vector machines for classification in nonstandard situations," *Machine Learning*, vol. 46, no. 1–3, pp. 191–202, 2002.

[11] B. Raskutti and A. Kowalczyk, "Extreme rebalanceing for svms: a case study," *SIGKDD Explorations*, vol. 6, no. 1, pp. 60–69, 2004.

[12] A. Fernández, S. García, M. Jesus, and F. Herrera, "A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets," *Fuzzy Sets and Systems*, vol. 159, no. 18, pp. 2378–2398, 2008.

[13] I. Tomek, "Two modifications of cnn," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 6, pp. 769–772, 1976.

[14] J. Laurikkala, "Improving identification of difficult small classes by balancing class distribution," in *Proceedings of the 8th Conference on AI in Medicine in Europe: Artificial Intelligence Medicine*, ser. AIME '01. London, UK, UK: Springer-Verlag, 2001, pp. 63–66.

[15] L. Eshelman, "The chc adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination," in *Foundations of Genetic Algorithms*. Morgan Kaufmann, 1991, pp. 265–283.

[16] J. Cano, F. Herrera, and M. Lozano, "Using evolutionary algorithms as instance selection for data reduction in kdd: An experimental study," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 6, pp. 561–575, 2003.

[17] P. Hart, "The condensed nearest neighbor rule," *IEEE Transactions on Information Theory*, vol. 14, pp. 515–516, 1968.

[18] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: one-sided selection," in *Proceedings of the 14th International Conference on Machine Learning*, 1997, pp. 179–186.

[19] A. Asuncion and D. Newman. (2007) Uci machine learning repository. School of Information and Computer Sciences. University of California, Irvine. [Online]. Available: http://www.ics.uci.edu/ mlearn/MLRepository.html

[20] C. Chang and C. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 27, pp. 1–27, 2011. [Online]. Available: http://www.csie.ntu.edu.tw/ cjlin/libsvm

[21] Q. Gu, Z. Cai, L. Zhu, and B. Huang, "Data mining on imbalanced data sets," in *International Conference on Advanced Computer Theory and Engineering*, 2008, pp. 1020–1024.

[22] S. García, J. Derrac, I. Triguero, C. Carmona, and F. Herrera, "Evolutionary-based selection of generalized instances for imbalanced classification," *Knowledge-Based Systems*, vol. 25, no. 1, pp. 3–12, 2012.

[23] J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework." *Journal of Multiple-Valued Logic and Soft Computing*, vol. 17, no. 2–3, pp. 255–287, 2011. [Online]. Available: http://www.keel.es/