# Tackling Class Imbalance Problem in Binary Classification using Augmented Neighborhood Cleaning Algorithm

Nadyah Obaid Al Abdouli, Zeyar Aung*, Wei Lee Woon, and Davor Svetinovic

Institute Center for Smart and Sustainable Systems (iSmart),
Department of Electrical Engineering and Computer Science,
Masdar Institute of Science and Technology, Abu Dhabi, UAE.
{nalabdouli,zaung,wwoon}@masdar.ac.ae,ds@davors.com

**Abstract.** Many natural processes generate some observations more frequently than others. These processes result in an imbalanced distributions which cause classifiers to bias toward the majority class because most classifiers assume a normal distribution. In order to address the problem of class imbalance, a number of data preprocessing techniques, which can be generally categorized into over-sampling and under-sampling methods, have been proposed throughout the years. The Neighborhood cleaning rule (NCL) method proposed by Laurikkala is among the most popular under-sampling methods. In this paper, we augment the original NCL algorithm by cleaning the unwanted samples using CHC evolutionary algorithm instead of a simple nearest neighbor-based cleaning as in NCL. We name our augmented algorithm as NCL+. The performance of NCL+ is compared to that of NCL on 9 imbalanced datasets using 11 different classifiers. Experimental results show noticeable accuracy improvements by NCL+ over NCL. Moreover, NCL+ is also compared to another popular over-sampling method called Synthetic minority over-sampling technique (SMOTE), and is found to offer better results as well.

**Keywords:** Data Preprocessing, Class Imbalance, Under-Sampling, Neighborhood Cleaning, Evolutionary Algorithm.

## 1 Introduction

Natural processes often generate some observations more frequently than others. Therefore, they produce samples that may not have a normal class distribution. In many cases the class distribution could be highly imbalanced (a.k.a. skewed). This phenomenon exists in many real-world datasets in different real-world applications such as text classification [19], fraud detection [21], intrusion detection [10], customer behavior prediction [17], and environmental event monitoring [4]. The ratio between the numbers of samples of the majority class and those of the minority class is called the imbalance ratio (IR).

* Corresponding author.

In general, most classification algorithms assume normal class distribution. However, in the case of an imbalanced dataset, it is the majority class that creates a bias in the classifiers' decision. The classifiers tend to focus on the majority class and ignore the minority class. There are many cases in which the minority class represents the most important class of interest. For example, in diseased tree detection [4], it is extremely important to correctly classify the minority class. Generally, a normal classifier would misclassify many samples of the minority class that represents diseased trees. The imbalance distribution poses many challenges to widely used-classifiers such as decision tree, induction models, and multilayer perceptron neural networks [15].

In order to address the challenge of class imbalance, a number of solutions have been proposed throughout the years. These can be categorized into method-level (internal) and data-level (external a.k.a. preprocessing) approaches. Data-level processing approaches can be further categorized into over-sampling and under-sampling methods. Among a number of under-sampling methods [16, 12, 25, 24, 6], the Neighborhood cleaning rule (NCL) algorithm [16] is a simple and effective one.

In this paper, our objective is to further improve the accuracy performance of NCL by employing CHC evolutionary algorithm [9] in removing the unwanted samples. We name our proposed method as NCL+. The performance of NCL+ is compared to that of NCL on 9 UCI benchmark datasets [1] in conjunction with 11 commonly-used classifiers. Experimental results show noticeable accuracy improvements by NCL+ over NCL. Furthermore, NCL+ is also compared to another popular over-sampling method called Synthetic minority over-sampling technique (SMOTE), and is found to offer better results as well.

This paper is an extended version of a portion of research work presented in the master's thesis [3] of the first author.

## 2 Related Work

### 2.1 Method-level (Internal) Approaches

Many normal classification algorithms can be modified to take imbalanced data in account. Specifically, the modification may focus on adjusting the cost function, changing the probability estimation, or adapting recognition-based learning [12]. The algorithms that work at method level could be efficient. However, in many cases these algorithms are application specific. Thus, there needs special knowledge about both the classifier itself and the application domain in order to use them effectively [12, 13].

### 2.2 Data-level (Preprocessing) Approaches

The purpose of preprocessing is to balance and normalize the class distribution to certain extent before passing the dataset to the classifier. Sampling is the most used approach for overcoming misclassification problems due to imbalanced datasets [18]. There are two broad categories in sampling: over-sampling and under-sampling.

**Over-Sampling:** The purpose is to increase the size of the minority class by adding new synthetic samples. Some of the well-known over-sampling methods include Synthetic minority over-sampling technique (SMOTE) [8], SMOTE + Tomek's links (SMOTE-TL) [6], Selective preprocessing of imbalanced data 2 (SPIDER2) [20], Random over-sampling (ROS) [12], and Adaptive synthetic sampling (ADASYN) [14].

**Under-Sampling:** The aim is to reduce the size of the majority class set by removing some of its samples. Some of the well-known over-sampling methods include Neighborhood cleaning rule (NCL) [16], Tomek links (TL) [6], Condensed nearest neighbor rule (CNN) + Tomek's links (CNN-TL) [12], Under-sampling based on clustering (SBC) [24], and Class purity maximization (CPM) [25].

Here, we will elaborate the Neighborhood cleaning rule (NCL) [16] method because our proposed algorithm in this paper is based and improved upon it. NCL employs Wilson's edited nearest neighbor rule (ENN) [23]. NCL model maintains all the samples of the class of interest $C$ and reduce those from the rest of the data $O$. This process is accomplished in two phases. In the first phase, ENN is used to find the noisy data $A_1$ in $O$. Specifically, 3-ENN is used to remove the samples with different class than the majority class of the three nearest neighbors. Subsequently, the neighborhoods are processed again and the set $A_2$ is initially created. Then, the three nearest neighbor samples that belong to $O$ and lead to $C$ samples misclassification are iteratively inserted in the set $A_2$. Finally, the data is reduced by eliminating the samples that belong to either sets $A_1$ or $A_2$ (i.e., $A_1 \cup A_2$). Figure 1 describes the NCL algorithm.

**Algorithm** NCL
1. Split data $T$ into the class of interest $C$ and the rest of data $O$.
2. Identify noisy data $A_1$ in $O$ with edited nearest neighbor rule.
3. For each class $C_i$ in $O$
   if ($x \in C_i$ in 3-nearest neighbors of misclassified $y \in C$) and ($|C_i| \geq 0.5 \cdot |C|$)
   then $A_2 = \{x\} \cup A_2$
4. Reduced data $S = T - (A_1 \cup A_2)$

**Fig. 1.** NCL algorithm (reproduced from [16]).

## 3 Proposed NCL+ Algorithm

The proposed augmented algorithm, named NCL+, follows a similar architecture of the original neighborhood cleaning rule (NCL) algorithm described above in Section 2.2 and Figure 1. The data reduction process is carried out in two phases. In the first phase, the edited nearest neighbor (ENN) algorithm [23] is used to

create a subset named $A_1$. Unlike the original NCL algorithm which uses simple 3 nearest neighbors, the second phase is augmented by utilizing an evolutionary instance selection algorithm called CHC [9] to create a more carefully scrutinized subset, named $A_2$, of samples to be removed. Finally, $A_1 \cup A_2$ will be removed from the original dataset. The algorithm is shown in Figure 2.

**Algorithm** NCL+
1. Split data $T$ into the class of interest $C$ and the rest of data $O$.
2. For the subset $O$ do the following:
   2.1. Using ENN to identify the noisy data and insert them to subset $A_1$.
   2.2. Run CHC with the following fitness function:
   $Fitness(S) = \alpha \cdot$ `clas_rate` $+ (1 - \alpha) \cdot$ `pred_red`, where:
   $S$: a subset of the data set.
   $\alpha$: Alfa equilibrate factor.
   `class_rate`: classification rate (% of correctly classified samples in $T$ using only $S$ to find nearest neighbors).
   `perc_red`: percentage of reduction (calculated as: $100 \cdot (|T| - |S|)/|T|$).
   2.3. Insert the result into subset $A_2$.
   2.4. Reduce the dataset $S = T - (A_1 \cup A_2)$.

**Fig. 2.** Proposed NCL+ algorithm.

In many cases, there could be too many samples in the majority class. However, generally, not all the samples are equally informative during the training phase. Thus, instance selection algorithms such as CHC are useful to interpret the data independently of their location in the search space. CHC chooses the most representative samples. Consequently, it provides a high reduction rate while maintaining the accuracy. Cano et al. [7] has showed that CHC offered the best ranking in data reduction rates.

The CHC algorithm relies on reducing the data by means of evolutionary algorithm (EA) for instance selection. The EAs are adaptive models that rely on the principle of natural evolution. In CHC, an EA is used as instance selector to select the data to be removed. The C4.5 decision-tree induction algorithm [22] is used to build a decision tree using the selected instances. Then the new examples are classified using the resultant tree.

During each generation, CHC follows some basic steps that can be summarized as follows. First, it generates an intermediate population of size $N$ using the parent population of size $N$. Then, it randomly pairs them and use them to produce $N$ potential offspring. Then, a survival competition is held in order to select the next generation population. The best $N$ from the parent population and offspring are selected to form the next generation [7].

## 4 Experimental Results

In this section, we will present performance companions of the proposed NCL+ algorithm with its predecessor NCL [16] as well as a widely-used over-sampling algorithm, SMOTE [8].

### 4.1 Experimental Setup

We compare NCL+ to NCL and SMOTE on 9 imbalanced UCI datasets [1] using 11 commonly-used classifiers. The experimental setup is described below.

- **Software tool:** KEEL [5].
- **Classifiers:** 11 algorithms (Names are as given in KEEL.)
  - Cost-sensitive: `C_SVMCS-I`, `C4.5CS-I`, `NNCS-I`.
  - Ensemble: `AdaBoost-I`, `AdaBoostM2-I`, `AdaC2-I`, `Bagging-I`, `OverBagging2-I`, `MSMOTEBagging-I`, `UnderBagging2-I`, `UnderOverBagging-I`.
- **Datasets:** 9 UCI datasets [1] (Downloaded from KEEL dataset website [2]. The imbalance ratio (IR) for each dataset is given in Table 1.)
  - High IR ($> 9$): Abalone19, Yeast6, Glass6, Glass5, New-thyroid2.
  - Low IR (1.5 to 9): Vowel0, Vehicle1, Wisconsin, Ecoli-0_vs_1.
- **Preprocessing algorithms compared:** NCL+ (our proposed method), NCL [16] (`NCL-I` in KEEL), SMOTE [8] (`SMOTE-I` in KEEL).
- **Experimental mode:** 5-fold cross validation.
- **Evaluation criteria:** Average area under receiver operating characteristic curve (AUC) [11] for both classes.

### 4.2 NCL vs. NCL+

We can observe a recognizable improvement in the accuracy performance of NCL+ over the original NCL in terms of AUC. We can observe significant improvements in average AUC by 29.411% in the case of Abalone19 dataset (which has a very high IR of 128.87) and 11.231% in Vehicle dataset (with a relatively low IR of 3.23). Moderate improvements in average AUC between 0.809% and 5.192% are also observed for 4 datasets (Wisconsin, New-thyroid2, Vowel0, and Yeast6). However, for 2 datasets, NCL+ exhibits slight declines (less than 1%) in average AUC ($-0.268\%$ for Glass6 and $-0.723\%$ for Glass5). Table 1(a) presents the detailed comparison of accuracy results between NCL and NCL+ for each classifier.

Looking at the data distribution of processed data of both models and given the dataset size in each case, it is noticeable the CHC does not operate on quantity. It removes relatively fewer samples than NCL. However, the removed samples are the most defining ones whose absence can noticeably contribute to the improvement in classification accuracy. These results suggest that, in addition to CHC, other adaptive learning and evolutionary training sample selection algorithms, such as generational genetic algorithm (GGA) and population-based incremental learning (PBI), could also be used to further improve the quality of nearest neighbor-based under-sampling.

**Table 1.** Detailed comparisons of NCL and SMOTE vs. NCL+ on 9 different UCI datasets using 11 different classification algorithms in terms of average AUC values.

| Dataset | Abalone19 | | Yeast6 | | Glass6 | | Glass5 | | New-thyroid2 | | Vowel0 | | Vehicle1 | | Wisconsin | | Ecoli-0_vs_1 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IR | 128.87 | | 32.78 | | 22.81 | | 15.47 | | 10.1 | | 5.14 | | 3.23 | | 1.86 | | 1.86 | |
| (a) NCL vs. NCL+. | | | | | | | | | | | | | | | | | | |
| Classifier | NCL | NCL+ | NCL | NCL+ | NCL | NCL+ | NCL | NCL+ | NCL | NCL+ | NCL | NCL+ | NCL | NCL+ | NCL | NCL+ | NCL | NCL+ |
| C_SVMCS-I | 0.760 | 0.798 | 0.874 | 0.880 | 0.912 | 0.912 | 0.954 | 0.973 | 0.997 | 0.997 | 0.963 | 0.972 | 0.815 | 0.818 | 0.971 | 0.979 | 0.976 | 0.980 |
| C4.5CS-I | 0.548 | 0.800 | 0.846 | 0.907 | 0.923 | 0.890 | 0.988 | 0.943 | 0.949 | 0.989 | 0.942 | 0.983 | 0.755 | 0.864 | 0.957 | 0.971 | 0.959 | 0.987 |
| NNCS-I | 0.507 | 0.508 | 0.718 | 0.621 | 0.894 | 0.902 | 0.854 | 0.880 | 0.933 | 0.852 | 0.664 | 0.709 | 0.649 | 0.602 | 0.964 | 0.970 | 0.969 | 0.980 |
| AdaBoost-I | 0.515 | 0.799 | 0.793 | 0.896 | 0.923 | 0.885 | 0.938 | 0.948 | 0.946 | 0.997 | 0.970 | 0.988 | 0.761 | 0.909 | 0.970 | 0.981 | 0.973 | 0.980 |
| AdaBoostM2-I | 0.516 | 0.799 | 0.793 | 0.896 | 0.886 | 0.875 | 0.938 | 0.948 | 0.949 | 0.997 | 0.970 | 0.988 | 0.754 | 0.911 | 0.970 | 0.979 | 0.973 | 0.983 |
| AdaC2-I | 0.554 | 0.800 | 0.793 | 0.886 | 0.867 | 0.900 | 0.926 | 0.973 | 0.952 | 0.986 | 0.958 | 0.987 | 0.784 | 0.888 | 0.979 | 0.979 | 0.969 | 0.980 |
| Bagging-I | 0.500 | 0.500 | 0.782 | 0.798 | 0.872 | 0.830 | 0.940 | 0.840 | 0.940 | 0.949 | 0.930 | 0.988 | 0.761 | 0.884 | 0.969 | 0.974 | 0.983 | 0.987 |
| OverBagging2-I | 0.530 | 0.792 | 0.814 | 0.889 | 0.915 | 0.920 | 0.890 | 0.888 | 0.926 | 0.963 | 0.964 | 0.996 | 0.764 | 0.874 | 0.970 | 0.980 | 0.976 | 0.987 |
| MSMOTEBagging-I | 0.579 | 0.795 | 0.852 | 0.888 | 0.918 | 0.936 | 0.978 | 0.888 | 0.943 | 0.983 | 0.959 | 0.993 | 0.732 | 0.804 | 0.959 | 0.971 | no res. | 0.987 |
| UnderBagging2-I | 0.713 | 0.733 | 0.867 | 0.888 | 0.909 | 0.926 | 0.949 | 0.949 | 0.918 | 0.958 | 0.947 | 0.977 | 0.766 | 0.854 | 0.962 | 0.973 | 0.980 | 0.983 |
| UnderOverBagging-I | 0.547 | 0.792 | 0.836 | 0.884 | 0.909 | 0.926 | 0.940 | 0.990 | 0.941 | 0.969 | 0.961 | 0.990 | 0.780 | 0.845 | 0.970 | 0.973 | 0.976 | 0.983 |
| Average | 0.570 | 0.738 | 0.815 | 0.858 | 0.902 | 0.900 | 0.936 | 0.929 | 0.945 | 0.967 | 0.930 | 0.961 | 0.756 | 0.841 | 0.968 | 0.975 | 0.973 | 0.983 |
| Improvement (%) | | 29.411 | | 5.192 | | -0.268 | | -0.723 | | 2.363 | | 3.352 | | 11.231 | | 0.809 | | 1.004 |
| (b) SMOTE (denoted as SM) vs. NCL+. | | | | | | | | | | | | | | | | | | |
| Classifier | SM | NCL+ | SM | NCL+ | SM | NCL+ | SM | NCL+ | SM | NCL+ | SM | NCL+ | SM | NCL+ | SM | NCL+ | SM | NCL+ |
| C_SVMCS-I | 0.765 | 0.798 | 0.887 | 0.880 | 0.926 | 0.912 | no res. | 0.973 | 0.978 | 0.997 | 0.967 | 0.972 | 0.807 | 0.818 | 0.971 | 0.979 | 0.980 | 0.980 |
| C4.5CS-I | 0.596 | 0.800 | 0.834 | 0.907 | 0.892 | 0.890 | 0.954 | 0.943 | 0.944 | 0.989 | 0.972 | 0.983 | 0.684 | 0.864 | 0.958 | 0.971 | 0.983 | 0.987 |
| NNCS-I | 0.790 | 0.508 | 0.819 | 0.621 | 0.858 | 0.902 | 0.879 | 0.880 | 0.983 | 0.852 | 0.895 | 0.709 | 0.590 | 0.602 | 0.948 | 0.970 | 0.973 | 0.980 |
| AdaBoost-I | 0.539 | 0.799 | 0.816 | 0.896 | 0.855 | 0.885 | 0.880 | 0.948 | 0.955 | 0.997 | 0.976 | 0.988 | 0.733 | 0.909 | 0.964 | 0.981 | 0.973 | 0.980 |
| AdaBoostM2-I | 0.539 | 0.799 | 0.815 | 0.896 | 0.855 | 0.875 | 0.880 | 0.948 | 0.963 | 0.997 | 0.975 | 0.988 | 0.720 | 0.911 | 0.964 | 0.979 | 0.973 | 0.983 |
| AdaC2-I | 0.537 | 0.800 | 0.816 | 0.886 | 0.855 | 0.900 | 0.880 | 0.973 | 0.955 | 0.986 | 0.976 | 0.987 | 0.724 | 0.888 | 0.966 | 0.979 | 0.973 | 0.980 |
| Bagging-I | 0.527 | 0.500 | 0.834 | 0.798 | 0.898 | 0.830 | 0.966 | 0.840 | 0.958 | 0.949 | 0.981 | 0.988 | 0.742 | 0.884 | 0.962 | 0.974 | 0.980 | 0.987 |
| OverBagging2-I | 0.527 | 0.792 | 0.834 | 0.889 | 0.898 | 0.920 | 0.966 | 0.888 | 0.958 | 0.963 | 0.981 | 0.996 | 0.742 | 0.874 | 0.962 | 0.980 | 0.980 | 0.987 |
| MSMOTEBagging-I | no res. | 0.795 | 0.840 | 0.888 | 0.920 | 0.936 | 0.932 | 0.888 | 0.969 | 0.983 | 0.954 | 0.993 | no res. | 0.804 | 0.963 | 0.971 | no res. | 0.987 |
| UnderBagging2-I | 0.527 | 0.733 | 0.834 | 0.888 | 0.898 | 0.926 | 0.966 | 0.949 | 0.958 | 0.958 | 0.981 | 0.977 | 0.742 | 0.854 | 0.962 | 0.973 | 0.980 | 0.983 |
| UnderOverBagging-I | 0.550 | 0.792 | 0.829 | 0.884 | 0.918 | 0.926 | 0.968 | 0.990 | 0.966 | 0.969 | 0.983 | 0.990 | 0.750 | 0.845 | 0.971 | 0.973 | 0.969 | 0.983 |
| Average | 0.590 | 0.738 | 0.833 | 0.858 | 0.888 | 0.900 | 0.927 | 0.929 | 0.963 | 0.967 | 0.967 | 0.961 | 0.723 | 0.841 | 0.963 | 0.975 | 0.976 | 0.983 |
| Improvement (%) | | 25.111 | | 3.001 | | 1.317 | | 0.200 | | 0.495 | | -0.634 | | 16.292 | | 1.301 | | 0.715 |

### 4.3 SMOTE vs. NCL+

When compared to SMOTE, NCL+ also offers better results in terms of average AUC in 8 out of 9 datasets with improvements ranging from 0.200% to 25.111%. However, for one dataset (Vowel0), its performance is slightly inferior with −0.634% decline in average AUC. Table 1(b) shows the detailed comparison of accuracy results between SMOTE and NCL+ for each classifier.

## 5 Conclusions and Future Work

Class imbalance is a serious problem for classification applications when the minority class is important. In this research, an improvement to the well-known neighborhood cleaning rule (NCL) under-sampling algorithm by Laurikkala [16] was suggested by employing an effective CHC-based instance selection approach. The new NCL+ method was compared to both the original NCL method as well as another popular preprocessing method named SMOTE [8]. Experimental results on 9 imbalanced datasets with 11 commonly-used classifiers showed that the proposed NCL+ generally provided significantly better results than both NCL and SMOTE did.

This research was focused on imbalanced binary datasets. The proposed NCL+ in its current form cannot be directly applied to multi-class classification because of the substantial difference between binary and multi-class classifications. Thus, for future work, we have a plan to extend NCL+ to adapt it to the multi-class scenario.

## References

1. `http://archive.ics.uci.edu` (2014)
2. `http://sci2s.ugr.es/keel/datasets.php` (2014)
3. Al Abdouli, N.O.: Handling the Class Imbalance Problem in Binary Classification. Master's thesis, Masdar Institute of Science and Technology, Abu Dhabi, UAE (2014)
4. Alan, J.B., Ryutaro, T., Hoan, N.: A hybrid pansharpening approach and multiscale object-based image analysis for mapping diseased pine and oak trees. International Journal of Remote Sensing 34, 6969–6982 (2013)
5. Alcalá-Fdez, J., Fernandez, A., Luengo, J., Derrac, J., Garcia, S., Sanchez, L., Herrera, F.: KEEL data-mining software tool: Data set repository. Journal of Multiple-Valued Logic and Soft Computing 17, 255–287 (2011)
6. Batista, G.E., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explorations 6, 20–29 (2004)
7. Cano, J., Herrera, F., Lozano, M.: Using evolutionary algorithms as instance selection for data reduction in KDD: An experimental study. IEEE Transactions on Evolutionary Computing 7, 561–575 (2003)
8. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 16, 321–357 (2002)

9. Eshelman, L.J.: The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In: Proc. 1st Workshop on Foundations of Genetic Algorithms. pp. 265–283 (1990)
10. Faisal, M.A., Aung, Z., Williams, J., Sanchez, A.: Data-stream-based intrusion detection system for advanced metering infrastructure in smart grid: A feasibility study. IEEE Systems Journal (2014), in press
11. Fawcett, T.: An introduction to ROC analysis. Pattern Recognition Letters 27, 861–874 (2006)
12. Fernándeza, A., Garcíaa, S., Jesusb, M., Herreraa, F.: A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets. Fuzzy Sets and Systems 159, 2378–2398 (2008)
13. Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., Herrera, F.: A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. IEEE Transactions on Systems, Man, and Cybernetics Part C 42, 463–484 (2011)
14. He, H., Bai, Y., Garcia, E., Li, S.: ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: Proc. 2008 International Joint Conference on Neural Networks. pp. 1322–1328 (2008)
15. Jo, T., Japkowicz, N.: A multiple resampling method for learning from imbalanced data sets. Computational Intelligence 20, 18–36 (2004)
16. Laurikkala, J.: Improving identification of difficult small classes by balancing class distribution. In: Proc. 8th Conference on AI in Medicine in Europe. pp. 63–66 (2001)
17. Liu, N., Woon, W.L., Aung, Z., Afshari, A.: Handling class imbalance in customer behavior prediction. In: Proc. 2014 IEEE International Conference on Collaboration Technologies and Systems. pp. 100–103 (2014)
18. Lokanayaki, K., Malathi, A.: Data preprocessing for liver dataset using SMOTE. International Journal of Advanced Research in Computer Science and Software Engineering 3, 559–562 (2013)
19. Mladenii, D., Grobelnik, M.: Feature selection for unbalanced class distribution and naive Bayes. In: Proc. 16th International Conference on Machine Learning. pp. 258–267 (1999)
20. Napieralla, K., Stefanowski, J., Wilk, S.: Learning from imbalanced data in presence of noisy and borderline examples. In: Proc. 7th International Conference on Rough Sets and Current Trends in Computing. pp. 158–167 (2010)
21. Perera, K.S., Neupane, B., Faisal, M.A., Aung, Z., Woon, W.L.: A novel ensemble learning-based approach for click fraud detection in mobile advertising. In: Proc. 2013 International Conference on Mining Intelligence and Knowledge Exploration. Lecture Notes in Computer Science, vol. 8284, pp. 370–382 (2013)
22. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers (1993)
23. Wilson, D.R., Martinez, T.R.: Reduction techniques for instance-based learning algorithms. Machine Learning 38, 257–286 (2000)
24. Yen, S.J., Lee, Y.S.: Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset. In: Proc. 2006 International Conference on Intelligent Computing. pp. 731–740 (2006)
25. Yoon, K., Kwek, S.: An unsupervised learning approach to resolving the data imbalanced issue in supervised learning problems in functional genomics. In: Proc. 5th International Conference on Hybrid Intelligent Systems. pp. 303–308 (2005)