Information Sciences 509 (2020) 47–70

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Neighbourhood-based undersampling approach for handling imbalanced and overlapped data

Pattaramon Vuttipittayamongkol*, Eyad Elyan

School of Computing Sciences and Digital Media, Robert Gordon University, UK

ARTICLE INFO

Article history:
Received 9 February 2019
Revised 31 July 2019
Accepted 26 August 2019
Available online 3 September 2019

Keywords:
Imbalanced dataset
Undersampling
k-NN
Class overlap
Classification

ABSTRACT

Class imbalanced datasets are common across different domains including health, security, banking and others. A typical supervised learning algorithm tends to be biased towards the majority class when dealing with imbalanced datasets. The learning task becomes more challenging when there is also an overlap of instances from different classes. In this paper, we propose an undersampling framework for handling class imbalance in binary datasets by removing potential overlapped data points. Our methods are designed to identify and eliminate majority class instances from the overlapping region. Accurate identification and elimination of these instances maximise the visibility of the minority class instances and at the same time minimises excessive elimination of data, which reduces information loss. Four methods based on neighbourhood searching with different criteria to identify potential overlapped instances are proposed in this paper. Extensive experiments using simulated and real-world datasets were carried out. Results show comparable performance with state-of-the-art methods across different common metrics with exceptional and statistically significant improvements in sensitivity.

© 2019 Elsevier Inc. All rights reserved.

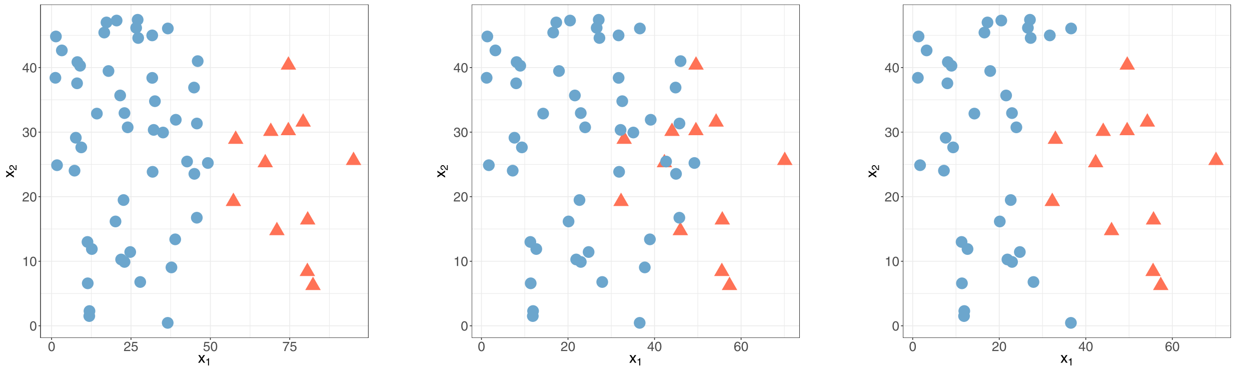
1. Introduction

In an imbalanced dataset, the class of interest is often underrepresented, and difficult to identify. Typical examples include fraud detection and medical predictions [9,20], where misclassifying an instance of the class of interest often comes at a high cost. This problem has attracted significant attention from the research community over the past years [16], and solutions addressing such a problem can be broadly categorised into data-level and algorithm-level methods [10]. Data-level methods involve data resampling, where the class distributions are adjusted [7], while algorithm-level methods involve creating new algorithms or modifying existing ones. Algorithm-level methods are fixed to the predetermined learning algorithms and require deep understanding of the algorithm and costs function. On the contrary, data resampling methods, which we are more interested in, are less complicated and able to be applied to any learning algorithms [3].

The most common resampling methods include random oversampling, random undersampling and Synthetic Minority Oversampling Technique (SMOTE) [7]. More recent resampling methods include k-means clustering [11,27], density-based clustering [4,6,27], neural networks [8], and ensemble [38]. These methods are designed to produce better data distribution. However, a number of studies showed that the classifiers' performance was affected more by class overlap rather than class distribution [9,13,36,39]. A recent study [41] supported these findings by showing significant improvement over state-of-the-

* Corresponding author.
E-mail addresses: p.vuttipittayamongkol@rgu.ac.uk (P. Vuttipittayamongkol), e.elyan@rgu.ac.uk (E. Elyan).

https://doi.org/10.1016/j.ins.2019.08.062
0020-0255/© 2019 Elsevier Inc. All rights reserved.



(a) (b) (c)

Fig. 1. Synthetic datasets with (a) imbalanced class distributions, (b) class imbalance and overlap problems, (c) negative instances removed from the overlapping region.

art k-means based method [27], by only focusing on removing negative instances from the overlapping region and without having to rebalance the data distribution. Consider Fig. 1a and b, which show two datasets with the same class distribution. Despite class imbalance, the learning task on the dataset in Fig. 1a is simple. Also, it is far easier than on the dataset in Fig. 1b due to the presence of class overlap. In real-world scenarios, datasets are often found imbalanced and overlapped. Therefore, undersampling majority class instances from the overlapping region is a reasonable approach to improve the learning algorithm performance.

In this paper, we propose a neighbourhood-based undersampling framework for identifying and eliminating overlapped negative instances. By applying this framework, we hypothesize that most of the negative instances will be removed from the overlapping region as illustrated in Fig. 1c. The benefits are twofold. First, it maximises the visibility of minority class instances. Second, by employing a neighbourhood search technique, more accurate identification of overlapped negative instances can be achieved, hence preventing excessive eliminations and minimising information loss. We introduce four different *k-NN* based methods to explore the local surroundings of individual instances and identify overlapped instances for elimination. The main contributions of this paper can be outlined as follows:

- Four *k-NN*-based methods for handling imbalanced datasets by accurately detecting and optimally removing potential overlapped negative instances are proposed. Different criteria to identify overlapped instances for removal are introduced. These methods are different from existing variations of *k-NN* in several aspects. First, we consider the entire overlapping region rather than just borderline instances. Second, the removal of potential overlapped negative instances is made based on the class overlap degree, not the class distribution. Finally, our methods proved to be capable of handling any degree of class overlap as can be seen in the Experiments Section.
- Extensive experiments using extremely imbalanced and overlapped simulated and real-world datasets were carried out.
- The methods presented provide a suitable framework for real-world application and domain-specific imbalanced problems, where high positive class accuracy is required and negative class accuracies can be compromised. This is evident by the significant improvement in sensitivity and other metrics achieved.

The rest of this paper is organised as follows: In Section 2, we review and discuss related work. Section 3 addresses the problem statement. In Section 4, the proposed methods are described and discussed in detail. Section 5 discusses the experimental setup. Section 6 presents the results and discussion. Finally, Section 7 concludes and discusses future work.

2. Related work

Solutions for class imbalanced problems at the data level aim at modifying the class distribution. A common practice is to resample data by either undersampling or oversampling, which reduces the majority class instances and increases the minority class instances, respectively. At the algorithmic level, the problem is handled by creating new learning algorithms or by modifying existing ones. Algorithm-level solutions have the advantage of directly incorporating the user's preferences into the model [3]. However, as opposed to data resampling, a learning algorithm needs to be predetermined and cannot be changed once implemented. Ensemble-based methods, which are combinations of data-level and algorithm-level methods, are also used. Because of the scope of this paper, our discussion will be focused on data-level solutions. For a detailed review of algorithm-level methods and ensemble-based methods, the reader is referred to Haixiang et al. [16]. Additional recent algorithm-level and ensemble-based methods can be seen in [14,28,32,35,42,46].

The class-imbalance issue has led to the development of well-established data-level solutions. However, for linearly separable or sufficiently large datasets, it was shown that results were not affected by any degree of class imbalance [20]. Other studies showed that class overlap had a higher negative impact on modelâs performance than class imbalance [9,13,36,39].

Thus, in this paper, we broadly categorise existing solutions as class distribution-based methods and class overlapped-based methods.

Random resampling is the most widely-used class distribution-based approach due to the simplicity of application. However, as balanced class distribution is typically the stopping criteria, random undersampling may lead to information loss while random oversampling is often prone to overfitting [29]. To replace random selection, Synthetic Minority Over-sampling Technique (SMOTE) [7] was introduced. The method generates new minority class instances by means of linear interpolation between neighbouring points. SMOTE has led to many well-known extensions [5,6,11,17,34]. Some of these extensions, e.g. Borderline-SMOTE [17] and SMOTE-IPF [34], focused on resampling instances within the overlapping region. In contrast, other extensions such as Safe-level-SMOTE [5] and DBSMOTE [6], avoided resampling hard-to-classify instances and showed less improvements compared to their counterpart overlap-based methods, in particular, Borderline-SMOTE [17] and DBMUTE [4]. Other recent methods based on clustering [27,31] and Neural Network [8] have also been proposed. These methods attempted to preserve the topology of the original data while rebalancing class distribution. A common drawback of class distribution-based methods is that the resampling rate is limited by imbalance degree. If the class imbalance is low, a small number of minority class instances generated may not be sufficient to emphasise the presence of the minority class near the class boundary. On the other hand, if the class imbalance is high, a significant amount of important information may be lost.

Since the class overlap problem of imbalanced datasets is focused in this paper, existing class overlap-based methods are extensively reviewed in the following subsection.

2.1. Class overlap-based methods

Class overlap-based methods deal with instances near the borderline or in the entire overlapping region. We define borderline instances as those near the physical borderline of their own class whereas overlapped instances can extend far from the borderlines. This concept is similar to that illustrated in [40]; however, we also consider borderline instances as a subset of overlapped instances. Literature shows that only few existing methods address the entire overlapping region. This might be due to the risk of losing information by excessive elimination of negative instances. However, there is a trade-off between sensitivity and specificity [45], and for some specific domains having a higher sensitivity is crucial and therefore scarifying some information becomes essential.

Examples of methods that address the entire overlapping region are [4,18,41]. A recent overlapped-based method, OBU [41], based on the removal of negative instances from the entire overlapping region led to significant improvements over a state-of-the-art class distribution-based method [27]. However, excessive eliminations were observed in some cases as a result of a global approach to identify the overlapping region. DBMUTE [4] utilised a density-based clustering algorithm to discover and undersample negative instances from the overlapping region. The method outperformed other existing methods such as DBSMOTE [6], which in contrast focused on instances outside the overlapping region. This evidences that between the two methods that employed the same technique, the variation that emphasised the class overlap issue led to better classification results. Another well-established method, Adaptive Synthetic sampling approach (ADASYN) [18], works by synthesizing more data points from positive examples surrounded by more negative neighbours. Results showed that ADASYN improved sensitivity. However, as opposed to undersampling, this method does not guarantee the maximum visibility of the positive class instances because negative instances are still present in the overlapping region.

Other methods focused on instances near the decision boundary. Edited Nearest Neighbour (ENN) is a long-standing undersampling method for imbalanced learning that was adapted from the study of Wilson [43]. ENN selectively removes majority class instances by considering its k nearest neighbours that belong to the other class, where $k = 3$. It has to be noted that the setting of the k value in this approach significantly impacts the performance. For example, a small k value can leave a lot of the overlapped majority class instances unremoved. Neighbourhood Cleaning Rule (NCL) [25] is an extension of this method, where the k nearest neighbours of both positive and negative instances were considered for the removal of negative instances. Results showed an improvement over a data-distribution based method proposed in [22]. Later overlap-based methods showed superior results over NCL. These include a combined cleaning and resampling approach, which consisted of neighbourhood cleaning and selectively oversampling positive instances in the overlapping region [23], and Evolutionary undersampling, which employed an Evolutionary algorithm to significantly reduce the dataset size and achieve optimal classification results [24].

A method to avoid erroneous eliminations of positive instances during noise cleaning was proposed in SMOTE-IPF [34]. This was done by first applying SMOTE, followed by applying a noise filtering algorithm. SMOTE-IPF allows new minority class instances to be generated before determining noisy instances. By doing so, fewer rare cases of the minority class will appear as noise, hence reducing erroneous eliminations, which is crucial for preserving highly important information. In Borderline-SMOTE [17] (BLSMOTE), synthetic instances were created from the borderline minority class instances and their nearest neighbours. Based on this approach, two methods were proposed. One considers only the minority-class nearest neighbours while the other includes the nearest neighbours of both classes in generating new instances. Results showed that the latter method whose synthetic instances were generated closer to the borderline achieved higher true positive rates. Redundancy-driven modified Tomek-link based undersampling [10] considered similarity and contribution factors as the undersampling criteria. This is achieved by removing redundant negative instances with the lowest contributions to classification. The undersampling process is terminated when a data balance is reached. In other words, the imbalance

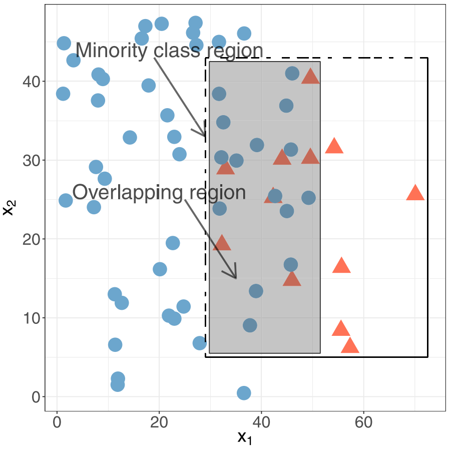


Fig. 2. Regions approximation.

degree indeed dominates the proposed elimination criteria, which could eventually lead to an insufficient elimination or an excessive elimination of negative instances.

In Majority Weighted Minority Oversampling Technique (MWMOTE) [1] and Adaptive Semi-Unsupervised Weighted Oversampling (A-SUWO) [30], new instances are synthesized within the sub-clusters of the minority class. Both methods used a bottom-up clustering technique that does not require the number of clusters to be fixed a prior. Each minority class instance was weighted with the selection probability based on its proximity to the majority class. This proximity was calculated differently in the two methods. MWMOTE assigned higher weights to minority class instances that are closer to the predefined borderline majority class instances while A-SUWO put higher weights on instances with more majority class nearest neighbours. Another weighting factor MWMOTE determined was the density of a sub-cluster. More synthetic positive instances were created in sparse sub-clusters. On the other hand, A-SUWO synthesised more instances in the sub-clusters with higher misclassification errors to handle small sub-clusters, which were the results of the within-class imbalance issue.

3. Problem statement

It was illustrated earlier in Fig. 1a and b how class overlap adds difficulties to classification tasks of a dataset with the imbalanced class distribution. Also, the performance of a learning algorithm on imbalanced datasets was shown to be highly dependant on the level of class overlap [13]. In this section, we express how class overlap is quantified. A detailed discussion of how we extended solution that deals with borderline instances to cover instances in the entire overlapping region is also provided.

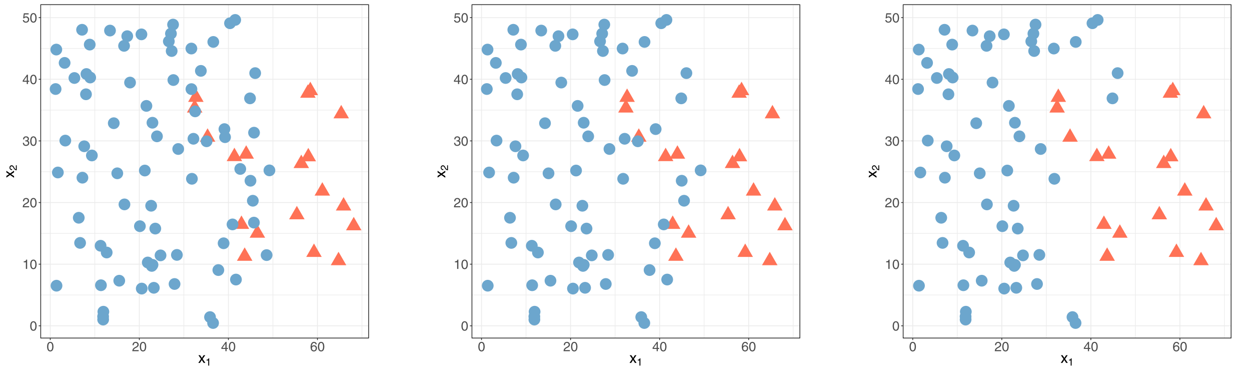
3.1. Quantification of class overlap

Since class overlap is not yet mathematically well-defined [37], several methods to estimate class overlap have been proposed, such as in [26,37,44]. However, these methods have some limitations, including a prior assumption of a normal distribution of data, which are not generally applicable to real-world datasets. To facilitate the measurement of class overlap degree later in the Experiments Section, we adapted the formula used in [13]. The formula was modified to suit class imbalanced problems as expressed in Eq. (1), and how the regions are approximated for the calculation is shown in Fig. 2(a). Note that this formula is only designed for binary-class datasets with 2-dimensional features for simplicity of exploring the problem.

$$overlap(\%) = \frac{overlapping\ area}{minority\ class\ area} * 100 \tag{1}$$

3.2. Borderline vs overlap

Fig. 3b gives an example of majority class instances that are close to the minority class borderline being removed from the original dataset (Fig. 3a). This is carried out by removing majority class instances that most of their three nearest neighbours are of the minority class. Now that the minority class instances in the overlapping region are more visible to the learner, the resulting dataset (Fig. 3b) is likely to produce better classification results. However, high classification errors in the minority class in the complex region may yet occur as the minority class is still under-represented. This issue can be addressed by further removal of the remaining majority class instances in such a region as demonstrated in Fig. 3c. To achieve this, the removal is performed on the majority class instances having a minority class instance as one of the three



(a) (b) (c)

Fig. 3. Undersampling solutions to (a) imbalanced and overlapped dataset with (b) borderline instances removed and (c) overlapped instances removed.

nearest neighbours. As a result, the visibility of the minority class as well as its class boundary are maximised. Such an approach is suitable for application domains where the accuracy of the class of interest cannot be compromised.

4. Proposed methods

Our approach is to maximise the visibility of the positive instances in the overlapping region by eliminating overlapped negative instances. To minimise excessive elimination, we utilised the *k-NN* rule to explore the local surroundings of each instance. By doing so, a near-optimal trade-off was achieved by minimising information loss and maximising sensitivity.

In this paper, we propose four neighbourhood-based (NB-based) undersampling methods. These are Basic Neighbourhood Search (NB-Basic), Modified Tomek Link Search (NB-Tomek), Common Nearest Neighbours Search (NB-Comm), and Recursive Search (NB-Rec). The methods vary in terms of local search criteria and negative instances elimination. NB-Basic is the first and simplest proposed method created to remove negative instances from the overlapping region without compromising any positive instance. This method showed an exceptional improvement in the minority class accuracy as will be discussed later. However, with such an approach, there is a risk of excessive elimination of negative instances, which could lead to a significant drop in accuracy. Three different methods were subsequently developed by varying the search criteria and queries. These methods were proposed to offer different trade-offs suitable for a wide range of real-world problems. NB-Tomek and NB-Comm were created to address the potential excessive elimination of negative instances. NB-Comm was then extended to NB-Rec aiming at improving the detection of overlapped negative instances. Algorithms 1–4 depict these methods.[1]

```
Algorithm 1: Basic Neighbourhood Search Undersampling.
  Data: training set, $k$
  Result: undersampled training set
  begin
      $T \leftarrow$ training set;
      $T_{neg} \leftarrow$ negative instances in $T$;
      foreach $x \in T_{neg}$ do
          $NN \leftarrow k$ nearest neighbours' class labels;
          if 'positive' $\in NN$ then
              $X \leftarrow X \cup \{x\}$;
      $\hat{T} \leftarrow T - X$;
      return $(\hat{T})$
```

4.1. Basic Neighbourhood Search

NB-Basic was implemented as in Algorithm 1. The method removes any negative query that has a positive neighbour.

As can be seen in Fig. 4(a), the query in the centre of the circle is marked as a potential overlapped instance because one of its nearest neighbours is a positive instance. Upon identifying all potential overlapped instances, the removal is executed. Only one positive neighbour is set as the elimination criterion to ensure the presence of every positive instance is clearly

[1] Source code available at https://github.com/fonkafon/NB-undersampling.git.

Algorithm 2: Modified Tomek Link Search Undersampling.

```
Data: training set, $k$
Result: undersampled training set
begin
    $T \leftarrow training\ set$;
    $T_{neg} \leftarrow negative\ instances\ in\ T$;
    foreach $x \in T_{neg}$ do
        $NN \leftarrow k\ nearest\ neighbours$;
        foreach $y \in NN$ do
            $c \leftarrow class(y)$;
            if $c ==$ '$positive$' then
                $NN_c \leftarrow k\ nearest\ neighbours\ of\ y$;
                if $x \in NN_c$ then
                    $X \leftarrow X \cup \{x\}$;
    $\hat{T} \leftarrow T - X$;
    return $(\hat{T})$
```

Algorithm 3: Common Nearest Neighbours Search Undersampling.

```
Data: training set, $k$
Result: undersampled training set
begin
    $T \leftarrow training\ set$;
    $T_{pos} \leftarrow positive\ instances\ in\ T$;
    $A \leftarrow frequency\ table$;
    foreach $x \in T_{pos}$ do
        $NN \leftarrow k\ nearest\ neighbours$;
        $NN_{neg} \leftarrow negative\ members\ of\ NN$;
        foreach $y \in NN_{neg}$ do
            $A_y.freq \leftarrow A_y.freq + 1$;
    foreach $x \in A.instance$ do
        if $A_x.freq > 1$ then
            $X \leftarrow X \cup \{x\}$;
    $\hat{T} \leftarrow T - X$;
    return $(\hat{T})$
```

visible to the learning algorithm. This is because the minority class information is considerably more valuable and losing part of it is highly undesirable.

4.2. Modified Tomek Link Search

Modified Tomek Link Search is proposed as an extension of NB-Basic to address potential excessive elimination of negative instances. As described in Algorithm 2, for every negative instance x with a positive neighbour y, x is removed only if it appears within the k nearest neighbours of the positive instance y. In other words, when the neighbourhood between a negative query and a positive query is established in both directions, the negative query in the modified Tomek Link is eliminated (Fig. 4(b)).

The rationale behind this second query is illustrated in Fig. 5 which shows that if q is within the k nearest neighbours of p, it does not necessarily mean that p is within the k nearest neighbours of q.

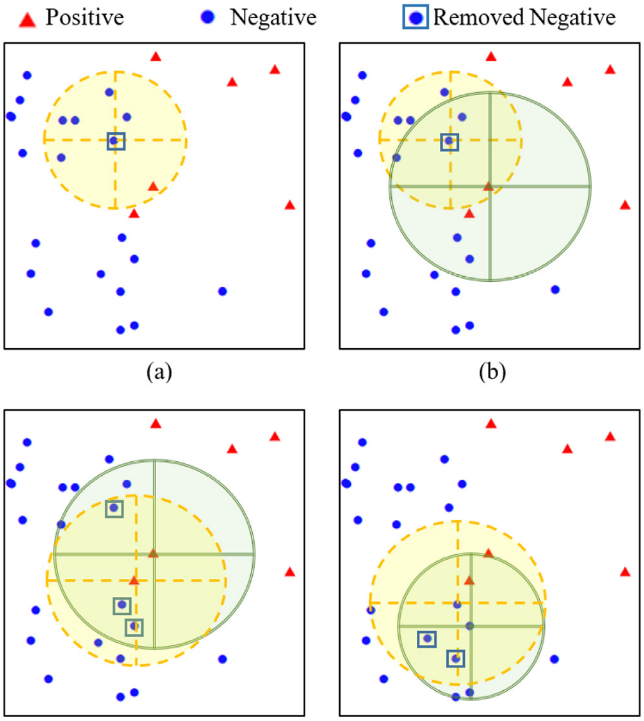
4.3. Common Nearest Neighbours Search

It was observed that when a majority class query was used, there was a higher probability that NB-Tomek would miss nearby positive instances. Therefore, in this variation, we propose an alternative method, NB-Comm, to remove common negative neighbours of positive instances. As defined in Algorithm 3, two positive queries will be used for considering the elimination of a negative instance. The common negative nearest neighbours of any two positive queries are identified as potential overlapped instances and removed (Fig. 4(c)).

NB-Comm, as discussed in the results section, provided competitive results. However, we hypothesise that the performance of the method can be dependant on the data density. In other words, when the density of the minority class is much lower than that of the majority class, fewer common nearest negative neighbours instances would be found.

Algorithm 4: Recursive Search Undersampling.

```
Data: training set, $k$, set $X$ from Algorithm 3
Result: undersampled training set
begin
    $T \leftarrow$ training set;
    $A' \leftarrow$ frequency table;
    foreach $x_1 \in X$ do
        $NN_2 \leftarrow k$ nearest neighbours;
        $NN_{2_{neg}} \leftarrow$ negative members of $NN_2$;
        foreach $y \in NN_{2_{neg}}$ do
            $A'_y.freq \leftarrow A'_y.freq + 1$;
    foreach $x_2 \in A'.instance$ do
        if $A'_{x_2}.freq > 1$ then
            $X_2 \leftarrow X_2 \cup \{x_2\}$;
    $\hat{T} \leftarrow T - (X \cup X_2)$;
    return $(\hat{T})$
```



(a) (b)

(c) (d)

Fig. 4. The proposed neighbourhood-based undersampling methods (a) NB-Basic (b) NB-Tomek (c) NB-Comm (d) NB-Rec.

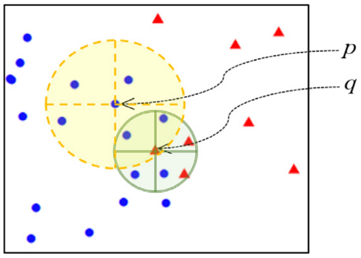


Fig. 5. Neighbourhood is *not* established in both directions.

4.4. Recursive search

NB-Rec is proposed as an extension of NB-Comm to ensure sufficient and accurate elimination of overlapped negative instances. From Algorithm 3, X is the set of potential negative instances to be eliminated by NB-Comm, all elements in X are used as the secondary queries in NB-Rec as described in Algorithm 4. The negative instances that are the common nearest neighbours of any pair of secondary queries are then to be eliminated along with all elements in X. We hypothesise that by introducing this extension, a finer-grained search criteria are provided. As a result, more overlapped negative instances will be detected, hence further improvements in sensitivity are expected (Fig. 4(d)).

5. Experiments

In order to provide conclusive results, we simulated 66 datasets representing a wide range of scenarios including extremely imbalanced and overlapped datasets and used them for evaluating our methods. Extensive experiments using 24 public real-world datasets were carried out for further evaluation. In addition, 2 large high-dimensional datasets were used in the final experiment to verify the consistency in the performance of our methods.

5.1. Setup

Three sets of experiments were carried out. In Experiment I, simulated datasets were used, and in Experiment II, small to medium-sized real-world datasets were used for evaluation. In Experiment III, further evaluation was carried out using large real-world datasets with high dimensions. The datasets used in Experiment II and III also include multi-class problems. To straightforwardly apply our methods on multi-class datasets without modifications, we treated one specific class as the minority class and employed the one-vs-all scheme, which is one of the most common strategies to handle multi-class problems [12] and was showed to have good performance [33]. Experimental results were compared with state-of-the-art and well-established methods for handling imbalanced datasets. These included class distribution-based methods namely, SMOTE [7] and k-means undersampling (kmUnder) [27], and class-overlap based methods including OBU [41], BLSMOTE [17] and ENN [43]. Support Vector Machine (SVM) and Random Forest (RF) were chosen as the learning algorithms. These classifiers are considered ones of the most widely used learning methods in imbalanced classification [16].

5.2. Datasets

In Experiment I, 66 uniformly-distributed binary-class datasets were simulated. These datasets capture wide ranges of class-overlap and imbalance degrees. The class imbalance degree is defined in Eq. (2) where N and P are the numbers of negative and positive instances in the dataset, respectively and values were set to 1.5, 3, 12, 30, 60, 120. For each imbalance degree, the class overlap degrees was varied between $0\% - 100\%$ in a step of 10. The number of negative instances was fixed at 6,000, and the number of positive instances was varied between $50 - 4,000$ with regard to the imbalance degree.

$$imbalance\ degree\ (imb) = \frac{N}{P} \tag{2}$$

Table 1 shows the public datasets that were used in Experiment II. These datasets were obtained from *UCI Repository*[2] and *KEEL Repository*.[3] The datasets vary in terms of imbalance degrees (*1.86–41.4*), number of features (*5–18*), and number of instances (*214–5,472*).

In Experiment III, we used the breast cancer dataset from *KDD Cup 2008*[4] and the handwritten digits dataset from *the MNIST database*.[5] The breast cancer dataset is 117-feature, binary-class and contains 102,294 samples with 101,671 negative and 623 positive samples, which makes $imb = 163.20$. The handwritten digits dataset is 10-class with 784 features, and contains 60,000 samples. Class 3 and class 5 were selected as the minority class in two scenarios, and were undersampled to obtained higher class imbalance degrees. In the first scenario, class 3 was undersampled such that $imb = 43.90$, which consists of 53,869 negative and 1227 positive instances. In the second scenario, class 5 was undersampled such that $imb = 20.13$, which consists of 53,869 negative and 2711 positive instances.

In all experiments, each dataset was partitioned into 80:20 ratio of training and testing sets. In Experiment I and II, 10-fold cross-validation was used in the training phase for the purpose of model selection whereas no cross-validation was applied to the large datasets in Experiment III.

5.3. Parameter settings

No parameters tuning or optimisation was performed in our experiments. This allows us to provide a fair comparison and assess our methods. In the proposed NB-based methods, k is an important parameter, where k-NN is used to investigate

[2] https://archive.ics.uci.edu/ml/index.php.

[3] http://sci2s.ugr.es/keel/imbalanced.php.

[4] https://www.kdd.org/kdd-cup/view/kdd-cup-2008.

[5] http://yann.lecun.com/exdb/mnist/.

Table 1
Datasets.

Dataset		Instances	Minority	Imbalance ratio	No. features
1	Wisconsin	683	239	1.86	9
2	Pima	768	268	1.87	8
3	Glass0	214	70	2.06	9
4	Vehicle1	846	217	2.9	18
5	Vehicle0	846	199	3.25	18
6	Ecoli1	336	77	3.36	7
7	New-thyroid1	215	35	5.14	5
8	New-thyroid2	215	35	5.14	5
9	Ecoli2	336	52	5.46	7
10	Segmemt0	2308	329	6.02	19
11	Yeast3	1484	163	8.1	8
12	Ecoli3	336	35	8.6	7
13	Yeast2vs4	514	51	9.08	8
14	Vowel0	988	90	9.98	13
15	Glass2	214	17	11.59	9
16	Yeast1vs7	459	30	14.3	7
17	Glass4	214	13	15.46	9
18	Ecoli4	336	20	15.8	7
19	Page-blocks13vs2	472	28	15.86	10
20	Abalone09-18	731	42	16.4	8
21	Glass5	214	9	22.78	9
22	Yeast4	1484	51	28.1	8
23	Ecoli0137vs26	281	7	39.14	7
24	Yeast6	1484	35	41.4	8

the surroundings of instances. A simple rule of thumb, where k is set to equal the square root of the dataset size, was considered. Furthermore, to take into account the class imbalance issue and promote the discovery of overlapped majority class instances, we adjusted the k value to also be proportional to the imbalance degree as can be seen in Eq. (3), where N is the number of instances in the dataset.

$$k = \sqrt{N} + \sqrt{imb} \tag{3}$$

Unlike the settings in our method, SMOTE requires a small k value to ensure better distribution of synthesized instances. In experiment I, k was set to equal 5, following Chawla et al. [7]. However, one of the real-world datasets used in Experiment II comprises too few positive instances, and assigning $k = 5$ was not applicable. To keep the same parameter settings for all methods and all datasets, $k = 3$ was assigned throughout for SMOTE-related procedures. To avoid biased results, we tested both $k = 3$ and $k = 5$ on all possible datasets, but no inferior results were obtained with $k = 3$. For ENN [43], kmUnder [27], and OBU [41], the same parameter settings as stated in the original work were used.

The default parameter settings of SVM and RF in *caret*[6] package in *R* were used. The Radial Bias Function kernel was used for SVM and the default *cost* $(C) = 1$ and $\gamma = \frac{1}{datadimension}$ remained unchanged. In RF, the number of trees was set to 500 and *mtry* was set to $\sqrt{datadimension}$.

5.4. Evaluation metrics

Sensitivity, G-mean, precision, and F1-score were selected for evaluating our methods. These are common metrics and widely used for imbalanced learning [2,3,15,19,21]. Sensitivity (Eq. (4)) measures the minority class accuracy and is considered essential metric for imbalanced problems. G-mean (Eq. (5)) is used to ensure a good balance between the accuracy of both classes [2,15] while it is not affected by the class distribution.

$$sensitivity = \frac{TP}{TP + FN} \tag{4}$$

$$G - mean = \sqrt{specificity * sensitivity}, \tag{5}$$

where

$$specificity = \frac{TN}{TN + FP} \tag{6}$$

Precision (Eq. (7)) and F1-score (Eq. (8)) provide good measures to evaluate the trade-offs between positive class accuracy anbd negative class errors. Precision, which considers the raw number of incorrectly classified negative instances,

[6] https://CRAN.R-project.org/package=caret.

might potentially be a misleading metric by underestimating the performance when data is highly imbalanced. Consider, for instance, a dataset of 10:1000 (positive:negative) distribution. Classification results of 10 true positives and 10 false positives resulting in 100% sensitivity and 99% specificity, wihch in general is highly desirable. However, precision in such case is 50% and might be very misleading. This observation also applies to F1-score where in this typical example, an F1-score of 67% can also be misleading. That said, both measures are still very useful to evaluate classifiers, especially if used to provide further insights on performance if G-mean and sensitivity were competitive across different methods.

$$precision = \frac{TP}{TP + FP} \tag{7}$$

$$F1 - score = \frac{2}{\frac{1}{sensitivity} + \frac{1}{precision}} \tag{8}$$

6. Results and discussion

Three experiments were carried out. In Experiment I, we discuss performance of our methods using simulated datasets. In Experiment II, we evaluate our methods using real-world datasets, and further in Experiment III, performance on the large, high-dimension real-world datasets is discussed.

6.1. Experiment I: Simulations

The main objective of this experiment is to assess the impact of class imbalance and class overlap on our methods performance across a wide range of degrees. Overall performance is discussed and compared against other existing methods. An experiments using 66 simulated datasets showed superior performance of the proposed methods across different metrics. In particular, our methods yielded highest sensitivity and competitive G-mean with nearly 100% true positive rates were achieved, which was interestingly relatively stable across all datasets regardless of class imbalance and class overlap degrees. This is clearly illustrated in Fig. 6, where results of our methods are presented with solid lines, the results of the other methods are marked with dashed lines, and the shaded areas are the areas under the performance curves of the baseline (SVM).

Among the proposed NB-based methods, NB-Rec showed the highest sensitivity (99.95%) and competitive G-mean, but least tolerable to information loss, resulting in the lowest precision and F1-scores. NB-Comm showed slightly better overall results than NB-Basic and NB-Tomek. A detailed discussion of these results is provided in the following subsections. Numerical results of this experiment are provided in the supplementary material.[7]

6.1.1. NB-based methods vs class-distribution based methods

The superior performance in sensitivity achieved by our methods in comparison with the class-distribution based methods namely, SMOTE [7][8] and kmUnder [27], strongly suggests that our NB-based methods promoted the visibility of the positive class across different class imbalance and class overlap degrees. Moreover, while our methods provided relatively stable sensitivity under different scenarios, sensitivity of other methods tended to drop when the class overlap degree increased.

The NB-based methods not only produced the highest sensitivity but also showed competitive G-mean with SMOTE, and produced higher overall G-mean than kmUnder. These improvements in terms of G-mean and sensitivity indicate that our methods had improved the trade-offs between sensitivity and specificity, which means we have reduced both false positives and false negatives, over state-of-the-art kmUnder across different ranges of class imbalance and class overlap degrees. It was observed that for low imbalanced datasets ($imb = 1.5, 3$), lower precision was obtained by the NB-based methods comparing to SMOTE and kmUnder; however, competitive F1-score was achieved. For datasets with higher degrees of class imbalance, our methods showed more favourable results over kmUnder in both precision and F1-score. It can be said that our methods yielded relatively better performance as the degree of class imbalance and class overlap increased. As for moderate to extreme imbalanced datasets (i.e. $imb = 12$ *to* $imb = 120$), our NB-based methods achieved comparable precision and F1-score with SMOTE in almost all datasets. It should be noted that with our methods, these results were obtained with less training data in comparison to SMOTE, which could potentially reduce training time, especially in case of large datasets.

6.1.2. NB-based methods vs class-overlap based methods

Our NB-based methods showed more favourable performance over other common and recent class-overlap based techniques (Fig. 6), which are BLSMOTE [17], ENN [43], and OBU [41]. All NB-based methods have competitive results in sensitivity and G-mean with OBU, but with higher precision and F1-score obtained. The improvements in precision and F1-score of our methods over OBU were significant, especially when the degrees of class imbalance and class overlap were higher.

[7] https://github.com/fonkafon/NB-undersampling_Results.git.

[8] In Fig. 6, SMOTE has similar performance in sensitivity to kmUnder (hence the line is not visible)

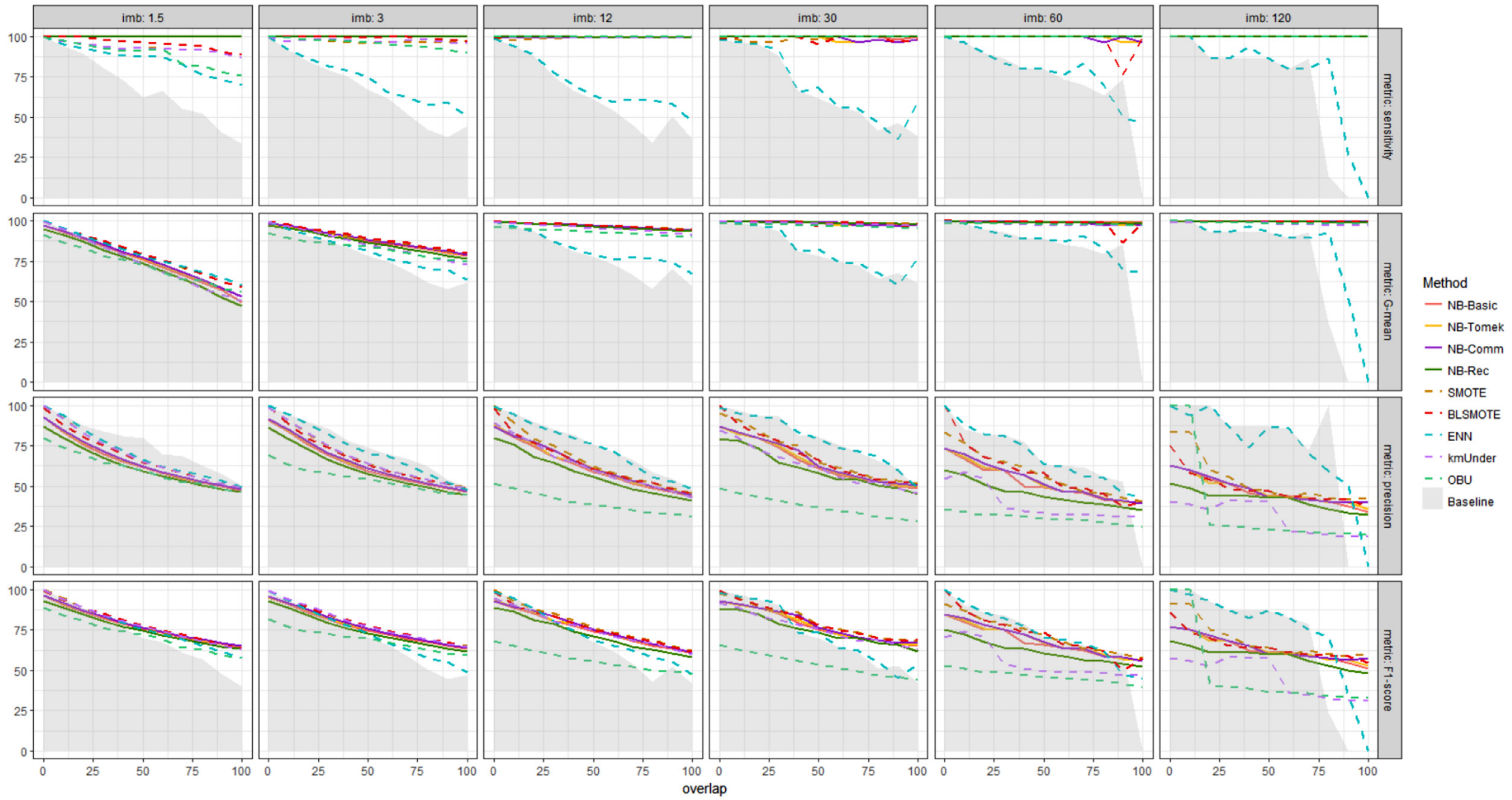


Fig. 6. Performance of methods across different imbalance and overlap degrees.

Table 2
Average classification results from Experiment I.

	NB-Basic	NB-Tomek	NB-Comm	NB-Rec	SMOTE	BLSMOTE	ENN	kmUnder	OBU	Baseline
sensitivity	99.86	99.59	99.64	**99.95**	98.11	98.56	75.52	98.35	97.46	67.75
G-mean	92.93	93.09	93.19	92.18	93.87	**93.78**	82.53	91.71	90.43	77.86
precision	58.83	59.67	60.12	54.59	64.21	62.83	72.8	55.3	43	**74.04**
F1-score	73.05	73.66	74.03	69.65	**76.41**	75.59	73.59	68.54	57.57	69.88

It suggests that our methods had relatively reduced both false positives and false negatives by the more accurate detection of potential overlapped negative instances and minimisation of information loss over the similar approach offered by OBU. Compared to BLSMOTE, our NB-based methods provided comparable G-mean, precision, and F1-score. However, our methods showed more stable sensitivity values throughout all class imbalance and class overlap degrees. This suggests that our NB-based methods had improved the positive class accuracy without sacrificing the performance in other metrics. In other words, by using our methods, lower false negatives can be achieved without increasing the number of false positives. Lastly, it can be said that our methods led to significantly better results than ENN in all scenarios as ENN barely improved the performance from the baseline.

6.1.3. Overall results

Table 2 shows the average performance of the methods across all imbalance and overlap degrees. Winning numbers are presented in **bold**. Our NB-based methods produced exceptionally high sensitivity. NB-Rec achieved the highest average sensitivity of 99.95%. Such high sensitivity is required across different domains, especially in medicine. Comparable results with SMOTE and BLSMOTE were achieved in terms of G-means, but with lower precision and recall. It was also observed that NB-Rec produced the lowest precision and recall when compared with the other proposed NB-based methods. This suggests that maximising the visibility of positive instances may cost an increase in the false positives. Our NB-based methods significantly outperformed ENN in sensitivity and G-mean with comparable F1-score (except F1-score of NB-Rec). More importantly, our methods outperformed state-of-the-art kmUnder and OBU in all measures, except for precision of NB-Rec that was competitive with kmUnder.

6.2. Experiment II: Real-world datasets

In this experiment, our methods were evaluated on real-world datasets. Tables 3–6 show the performance of our methods against other methods on the UCI datasets using SVM, where the datasets are sorted by imbalance ratio from low to high. These tables also show methods ranks and average ranking based on their performance (i.e. rank 1 means top performance and so on). Wilcoxon Signed Rank Tests were carried to assess statistical significance of the difference in methods performance. Results are presented in Table 7, with significance level of 0.05, the p values indicating a statistically significant difference between two methods are highlighted in **bold**.

As can be seen in Table 3, an overall superior performance over other methods in sensitivity was achieved by our NB-based methods as minimum false negatives occurred. These results are consistent with the results obtained from Experiment I. Amongst the four proposed methods, NB-Rec ranked top on average sensitivity, while NB-basic was the second best ranking method. NB-Comm and NB-Tomek had competive ranking with OBU, and higher ranking than kmUnder, SMOTE, BLSMOTE, and ENN. Improvements in sensitivity achieved by our methods over SMOTE, BLSMOTE and ENN is statically significant as shown in Table 7. Interestingly, both SMOTE and BLSMOTE did not improve the sensitivity and performed worse than the baseline in some cases.

On average, the highest ranking in G-mean was obtained using our NB-Comm, and it was significantly better than BLSMOTE (Table 7). The other NB-based methods outperformed SMOTE and BLSMOTE, and showed comparable performance with ENN, kmUnder and OBU in G-mean. This is also consistent with the results on synthetic datasets.

SMOTE, BLSMOTE, and ENN outperformed our methods in precision (Table 5) but with significantly lower sensitivity values. Such a trade-off is not generally desirable in some specific imbalanced domains. In contrast, all our methods, outperformed state-of-the-art kmUnder in both sensitivity and precision. Similarly, our methods outperformed OBU in precision with comparable results in sensitivity and G-mean.

In conclusion, NB-Comm on average ranked best in F1-score (Table 6) and G-mean. The method also showed high average ranking in terms of sensitivity, and relatively good ranking in precision. This high performance across the different measures reflects a good trade-off between true positive and false positive rates of NB-Comm. Even though relatively high false positives in comparison to low false negatives might be obtained due to the class imbalance nature, low false positive and low false negative rates can be achieved with regards to the total numbers of the majority class instances and minority class instances, respectively. This also shows better trade-off over other methods such as OBU, where high sensitivity was obtained but with lower precision and F1-score. NB-Basic and NB-Tomek showed competitive average ranking in F1-score. They provided comparable trade-offs between sensitivity, and G-means and F1-score, to OBU and kmUnder. In particular, NB-Basic and OBU had higher positive class accuracy but lower negative class accuracy than NB-Tomek and kmUnder. Thus, it can be said that NB-Basic, OBU, NB-Tomek, and kmUnder performed comparably on these datasets, and to consider which

Table 3
Sensitivity values and ranks with SVM baseline from Experiment II.

	Sensitivity value/rank																			
Dataset	NB-Basic		NB-Tomek		NB-Comm		NB-Rec		SMOTE		BLSMOTE		ENN		kmUnder		OBU		Baseline	
Wisconsin	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	97.87	6	97.87	6	97.87	6	97.87	6	**100**	**1**	97.87	6
Pima	98.11	2	96.23	3	92.45	5	**100**	**1**	43.4	10	47.17	8	52.83	7	79.25	6	96.23	3	47.17	8
Glass0	**85.71**	**1**	**85.71**	**1**	71.43	6	**85.71**	**1**	50	8	42.86	9	57.14	7	78.57	5	**85.71**	**1**	42.86	9
Vehicle1	**100**	**1**	**100**	**1**	95.35	3	95.35	3	25.58	10	27.91	8	34.88	7	83.72	6	86.05	5	27.91	8
Vehicle0	**100**	**1**	94.87	4	**100**	**1**	94.87	4	71.79	10	82.05	6	82.05	6	76.92	9	97.44	3	82.05	6
Ecoli1	**100**	**1**	93.33	3	93.33	3	86.67	5	53.33	10	66.67	7	66.67	7	80	6	**100**	**1**	60	9
New-thyroid1	71.43	5	71.43	5	**100**	**1**	**100**	**1**	57.14	7	57.14	7	57.14	7	**100**	**1**	**100**	**1**	57.14	7
New-thyroid2	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	85.71	10	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**
Ecoli2	**90**	**1**	**90**	**1**	**90**	**1**	**90**	**1**	70	10	80	8	**90**	**1**	**90**	**1**	**90**	**1**	80	8
Segmemt0	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	95.38	8	98.46	6	95.38	8	96.92	7	**100**	**1**	95.38	8
Yeast3	96.88	4	**100**	**1**	**100**	**1**	**100**	**1**	46.88	10	50	8	56.25	7	87.5	5	65.63	6	50	8
Ecoli3	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	28.57	8	14.29	9	57.14	7	**100**	**1**	**100**	**1**	14.29	9
Yeast2vs4	**80**	**1**	70	3	70	3	60	6	40	8	30	10	50	7	**80**	**1**	70	3	40	8
Vowel0	88.89	4	88.89	4	**100**	**1**	**100**	**1**	72.22	10	88.89	4	88.89	4	88.89	4	**100**	**1**	88.89	4
Glass2	**66.67**	**1**	**66.67**	**1**	**66.67**	**1**	**66.67**	**1**	33.33	7	33.33	7	0	9	**66.67**	**1**	**66.67**	**1**	0	9
Yeast1vs7	50	2	16.67	6	33.33	4	**83.33**	**1**	0	7	0	7	0	7	50	2	33.33	4	0	7
Glass4	50	3	50	3	50	3	**100**	**1**	50	3	50	3	50	3	**100**	**1**	50	3	50	3
Ecoli4	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	75	10	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**
Page-blocks13vs2	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	80	10	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**
Abalone09-18	50	4	37.5	6	50	4	**75**	**1**	12.5	7	0	8	0	8	62.5	3	**75**	**1**	0	8
Glass5	0	–	0	–	0	–	0	–	0	–	0	–	0	–	0	–	0	-	0	-
Yeast4	**80**	**1**	**80**	**1**	**80**	**1**	**80**	**1**	30	7	30	7	0	10	60	5	50	6	10	9
Ecoli0137vs26	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**
Yeast6	**85.71**	**1**	**85.71**	**1**	57.14	5	**85.71**	**1**	57.14	5	14.29	9	28.57	8	57.14	5	**85.71**	**1**	14.29	9
Average		1.74		2.22		2.17		**1.61**		6.74		7.3		5.65		3.43		2.09		6.39

Table 4
G-mean values and ranks with SVM baseline from Experiment II.

	G-mean value/rank																			
Dataset	NB-Basic		NB-Tomek		NB-Comm		NB-Rec		SMOTE		BLSMOTE		ENN		kmUnder		OBU		Baseline	
Wisconsin	**97.12**	**1**	**97.12**	**1**	**97.12**	**1**	95.35	9	96.66	4	96.66	4	96.66	4	96.66	4	0	10	96.66	4
Pima	47.5	8	52.83	7	55.24	6	0	10	60.02	5	61.43	3	64.6	2	**67.8**	**1**	29.43	9	61.43	3
Glass0	67.76	7	69.99	5	73.19	3	72.14	4	68.14	6	64.29	9	74.23	2	**80.34**	**1**	67.76	7	64.29	9
Vehicle1	40	9	46.48	8	55.92	2	51.67	4	48.93	7	51.54	5	55.91	3	**70.4**	**1**	38.02	10	51.54	5
Vehicle0	85.82	7	**89.94**	**1**	87.16	6	87.88	5	84.07	9	88.81	2	88.81	2	84.59	8	80.6	10	88.09	4
Ecoli1	91.82	2	88.71	3	87.67	4	82.45	6	70.11	10	80.85	7	80.85	7	84.95	5	**93.93**	**1**	77.46	9
New-thyroid1	82.13	5	82.13	5	95.74	2	94.28	3	75.59	7	75.59	7	75.59	7	**100**	**1**	89.75	4	75.59	7
New-thyroid2	95.74	4	95.74	4	95.74	4	88.19	10	**100**	**1**	92.58	9	98.6	2	95.74	4	92.8	8	98.6	2
Ecoli2	94.02	2	94.02	2	94.02	2	93.16	5	83.67	9	89.44	7	**94.87**	**1**	92.29	6	78.15	10	89.44	7
Segmemt0	87.58	8	92.91	7	94	6	87.15	9	97.54	5	**98.85**	**1**	97.67	3	97.95	2	84.64	10	97.67	3
Yeast3	90.66	3	**93.74**	**1**	91.08	2	73.85	7	67.68	10	70.04	8	74.14	6	90.11	4	78.99	5	69.9	9
Ecoli3	92.2	4	93.09	2	**94.87**	**1**	93.09	2	52.55	8	36.84	10	74.32	7	89.44	6	92.2	4	37.16	9
Yeast2vs4	86.98	2	82.29	3	81.83	5	72.23	6	62.9	9	54.77	10	70.71	7	**88.47**	**1**	82.29	3	63.25	8
Vowel0	94.28	4	94.28	4	**99.72**	**1**	98.31	3	84.98	10	94.28	4	94.28	4	94.28	4	**99.72**	**1**	94.28	4
Glass2	73.96	2	**76.24**	**1**	73.96	2	71.61	4	57.74	6	57.74	6	0	9	66.67	5	55.47	8	0	9
Yeast1vs7	57.9	3	36.78	6	51.64	4	61.83	2	0	7	0	7	0	7	**64.17**	**1**	51.26	5	0	7
Glass4	70.71	3	70.71	3	70.71	3	**82.16**	**1**	70.71	3	70.71	3	70.71	3	**82.16**	**1**	70.71	3	70.71	3
Ecoli4	99.2	7	99.2	7	**100**	**1**	98.4	9	**100**	**1**	86.6	10	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**
Page-blocks13vs2	97.7	5	97.7	5	97.7	5	89.19	10	**100**	**1**	89.44	9	99.43	3	97.12	8	**100**	**1**	99.43	3
Abalone09-18	56.35	5	52.84	6	64.5	3	64.5	3	35.1	7	0	8	0	8	66.52	2	**67**	**1**	0	8
Glass5	0	–	0	–	0	–	0	–	0	–	0	–	0	–	0	–	0	–	0	–
Yeast4	81.42	3	83.79	2	**84.29**	**1**	77.19	4	54.29	7	54.29	7	0	10	72.13	5	66.64	6	31.62	9
Ecoli0137vs26	98.13	8	98.13	8	**100**	**1**	99.07	7	**100**	**1**	**100**	**1**	**100**	**1**	96.23	10	**100**	**1**	**100**	**1**
Yeast6	90.97	2	**91.13**	**1**	74.54	7	90.97	2	75.46	5	37.67	10	53.27	8	74.8	6	90.64	4	37.8	9
Average		4.52		4		**3.13**		5.43		6		6.39		4.65		3.78		5.3		5.78

Table 5
Precision values and ranks with SVM baseline from Experiment II.

	Precision value/rank																			
Dataset	NB-Basic		NB-Tomek		NB-Comm		NB-Rec		SMOTE		BLSMOTE		ENN		kmUnder		OBU		Baseline	
Wisconsin	91.18	8	91.1	9	91.77	7	**92.15**	**1**	92.06	2	92.06	2	92.06	2	92.06	2	34.99	10	92.06	2
Pima	91.33	2	83.4	3	81.13	4	**98.17**	**1**	57.77	5	55.83	7	57.42	6	50.28	9	42.23	10	55.83	7
Glass0	76.36	5	65.12	9	69.57	6	66.7	8	77.29	4	85.37	2	**88.61**	**1**	68.14	7	56.38	10	85.37	2
Vehicle1	72.39	2	68.51	4	69.37	3	**87.47**	**1**	57.96	7	66.73	5	53.64	8	41.45	9	32.38	10	66.73	5
Vehicle0	64.03	9	74.11	7	66	8	76.83	6	**93.44**	**1**	86.69	2	86.69	2	77.23	5	60.7	10	82.3	4
Ecoli1	72.49	6	70.28	8	69.05	9	74.12	5	66.91	10	91	2	91	2	70.81	7	83.05	4	**100**	**1**
New-thyroid1	75.31	8	75.31	8	76.89	7	83.44	6	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	65.49	10	**100**	**1**
New-thyroid2	74.01	6	73.68	7	76.54	5	70.79	9	**100**	**1**	**100**	**1**	87.5	3	70	10	72.97	8	87.5	3
Ecoli2	94.21	5	92.32	6	91.41	7	87.3	8	**100**	**1**	**100**	**1**	**100**	**1**	75.47	9	42.99	10	**100**	**1**
Segmemt0	52.22	8	62.07	7	65.29	6	51.49	10	98.43	3	95.57	4	**100**	**1**	94.09	5	51.98	9	**100**	**1**
Yeast3	57.27	7	60.79	5	52.58	9	56.05	8	71.79	4	**76.51**	**1**	75.33	2	60	6	27.82	10	73.08	3
Ecoli3	50.22	6	51.47	4	56.68	3	51.34	5	49.92	7	24.94	10	**66.59**	**1**	36.76	8	59.84	2	33.26	9
Yeast2vs4	68.85	8	74.58	6	69.04	7	54.41	9	80.21	5	**100**	**1**	**100**	**1**	80.21	4	30.33	10	**100**	**1**
Vowel0	**100**	**1**	**100**	**1**	96.1	9	89.65	10	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	97.28	8	**100**	**1**
Glass2	44.07	3	41.41	4	31.9	6	33.58	5	**100**	**1**	**100**	**1**	0	9	14.72	7	11.44	8	0	9
Yeast1vs7	17.91	2	8.88	5	15.44	4	**25.15**	**1**	0	7	0	7	0	7	16.54	3	4.56	6	0	7
Glass4	**100**	**1**	**100**	**1**	**100**	**1**	39.89	8	**100**	**1**	**100**	**1**	**100**	**1**	16.6	9	9.94	10	**100**	**1**
Ecoli4	82.62	8	82.35	9	**100**	**1**	72.31	10	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**
Page-blocks13vs2	62.09	7	61.78	8	63.57	6	33.32	10	**100**	**1**	**100**	**1**	84.73	4	52.6	9	**100**	**1**	84.73	4
Abalone09-18	14.85	4	14.77	5	21.62	2	19.66	3	**34.29**	**1**	0	8	0	8	11.54	6	10.22	7	0	8
Glass5	0	–	0	–	0	–	0	–	0	–	0	–	0	–	0	–	0	–	0	–
Yeast4	19.63	7	23.57	5	24.22	4	21.01	6	37.92	2	37.92	2	0	10	13.85	8	5.06	9	**100**	**1**
Ecoli0137vs26	47.93	8	46.69	9	**100**	**1**	66.67	7	**100**	**1**	**100**	**1**	**100**	**1**	25.64	10	**100**	**1**	**100**	**1**
Yeast6	42.41	6	43.23	5	35.37	8	46.56	4	79.96	2	33.27	9	49.93	3	39.93	7	18.11	10	**100**	**1**
Average		5.52		5.87		5.35		6.13		**3**		3.09		3.3		6.22		7.57		3.22

Table 6
F1-score values and ranks with SVM baseline from Experiment II.

	F1-score value/rank																			
Dataset	NB-Basic		NB-Tomek		NB-Comm		NB-Rec		SMOTE		BLSMOTE		ENN		kmUnder		OBU		Baseline	
Wisconsin	**94.95**	**1**	**94.95**	**1**	**94.95**	**1**	92.16	9	94.85	4	94.85	4	94.85	4	94.85	4	51.65	10	94.85	4
Pima	57.14	4	58.29	2	57.99	3	51.46	7	49.46	10	51.02	8	54.9	5	**61.31**	**1**	52.31	6	51.02	8
Glass0	61.54	6	63.16	5	64.52	4	64.86	3	60.87	8	57.14	9	69.57	2	**73.33**	**1**	61.54	6	57.14	9
Vehicle1	45.03	5	46.74	4	48.81	2	47.13	3	35.48	10	39.34	8	42.25	6	**55.38**	**1**	40.22	7	39.34	8
Vehicle0	69.64	9	77.89	5	71.56	8	74	7	81.16	4	**84.21**	**1**	**84.21**	**1**	76.92	6	63.33	10	82.05	3
Ecoli1	78.95	2	75.68	5	73.68	8	66.67	9	59.26	10	76.92	3	76.92	3	75	6	**83.33**	**1**	75	6
New-thyroid1	71.43	8	71.43	8	82.35	2	77.78	3	72.73	4	72.73	4	72.73	4	**100**	**1**	66.67	10	72.73	4
New-thyroid2	82.35	5	82.35	5	82.35	5	63.64	10	**100**	**1**	92.31	4	93.33	2	82.35	5	73.68	9	93.33	2
Ecoli2	90	2	90	2	90	2	85.71	7	82.35	8	88.89	5	**94.74**	**1**	81.82	9	48.65	10	88.89	5
Segmemt0	58.56	8	70.65	7	73.86	6	57.78	9	96.88	4	96.97	3	**97.64**	**1**	95.45	5	53.72	10	**97.64**	**1**
Yeast3	60.19	6	66.67	2	58.72	8	34.78	10	56.6	9	60.38	5	64.29	3	**70.89**	**1**	63.64	4	59.26	7
Ecoli3	60.87	5	63.64	2	**70**	**1**	63.64	2	36.36	8	18.18	10	61.54	4	53.85	7	60.87	5	20	9
Yeast2vs4	69.57	4	70	2	66.67	5	42.86	10	53.33	8	46.15	9	66.67	5	**80**	**1**	70	2	57.14	7
Vowel0	94.12	3	94.12	3	**97.3**	**1**	85.71	9	83.87	10	94.12	3	94.12	3	94.12	3	**97.3**	**1**	94.12	3
Glass2	33.33	4	40	3	33.33	4	28.57	6	**50**	**1**	**50**	**1**	0	9	22.22	7	15.38	8	0	9
Yeast1vs7	16.22	3	8.7	6	16	4	17.54	2	0	7	0	7	0	7	**25**	**1**	15.38	5	0	7
Glass4	**66.67**	**1**	**66.67**	**1**	**66.67**	**1**	23.53	9	66.67	1	**66.67**	**1**	**66.67**	**1**	23.53	9	66.67	1	**66.67**	**1**
Ecoli4	88.89	7	88.89	7	**100**	**1**	80	10	**100**	**1**	85.71	9	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**
Page-blocks13vs2	71.43	6	71.43	6	71.43	6	35.71	10	**100**	**1**	88.89	5	90.91	3	66.67	9	**100**	**1**	90.91	3
Abalone09-18	12.9	7	13.04	6	**22.86**	**1**	16	5	18.18	3	0	8	0	8	18.87	2	17.39	4	0	8
Glass5	0	–	0	–	0	–	0	–	0	–	0	–	0	–	0	–	0	–	0	–
Yeast4	23.88	5	30.19	4	32	3	17.58	9	**33.33**	**1**	**33.33**	**1**	0	10	22.22	6	21.28	7	18.18	8
Ecoli0137vs26	50	8	50	8	**100**	**1**	66.67	7	**100**	**1**	**100**	**1**	**100**	**1**	33.33	10	**100**	**1**	**100**	**1**
Yeast6	52.17	3	54.55	2	42.11	7	52.17	3	**66.67**	**1**	20	10	36.36	8	47.06	6	48	5	25	9
Average		4.87		4.17		**3.65**		6.91		5		5.17		4		4.43		5.39		5.35

Table 7
p-values of the Wilcoxon Signed Rank Tests with SVM baseline from Experiment II.

			Sensitivity			
	SMOTE	BLSMOTE	ENN	kmUnder	OBU	Baseline
NB-Basic	**2.71E−03**	**3.82E−04**	**1.04E−02**	4.35E−01	8.81E−01	**3.66E−03**
NB-Tomek	**1.04E−02**	**1.27E−03**	**1.90E−02**	6.07E−01	8.98E−01	**1.01E−02**
NB-Comm	**5.16E−03**	**6.99E−04**	**1.08E−02**	5.11E−01	8.80E−01	**5.14E−03**
NB-Rec	**3.99E−04**	**4.04E−05**	**1.20E−03**	1.13E−01	4.16E−01	**7.98E−04**
			G-mean			
	SMOTE	BLSMOTE	ENN	kmUnder	OBU	Baseline
NB-Basic	2.48E−01	9.89E−02	4.76E−01	7.34E−01	4.09E−01	2.52E−01
NB-Tomek	2.65E−01	9.89E−02	4.64E−01	8.45E−01	3.70E−01	2.27E−01
NB-Comm	1.60E−01	**4.88E−02**	3.07E−01	9.42E−01	2.70E−01	1.37E−01
NB-Rec	2.70E−01	1.49E−01	6.20E−01	5.03E−01	6.43E−01	3.12E−01
			precision			
	SMOTE	BLSMOTE	ENN	kmUnder	OBU	Baseline
NB-Basic	**4.31E−02**	5.07E−02	1.53E−01	4.70E−01	9.88E−02	**3.27E−02**
NB-Tomek	**4.76E−02**	**4.60E−02**	1.36E−01	5.36E−01	9.07E−02	**3.10E−02**
NB-Comm	7.39E−02	9.56E−02	2.90E−01	4.15E−01	6.47E−02	5.73E−02
NB-Rec	**1.00E−02**	**1.92E−02**	6.56E−02	7.57E−01	2.01E−01	**1.17E−02**
			F1-score			
	SMOTE	BLSMOTE	ENN	kmUnder	OBU	Baseline
NB-Basic	3.82E−01	3.78E−01	**2.70E−02**	5.83E−01	8.26E−01	2.33E−01
NB-Tomek	5.32E−01	4.81E−01	5.63E−02	7.42E−01	5.38E−01	3.12E−01
NB-Comm	8.02E−01	8.23E−01	1.97E−01	8.35E−01	2.96E−01	6.52E−01
NB-Rec	7.45E−02	7.40E−02	**2.62E−03**	1.95E−01	6.13E−01	**4.45E−02**

method was more preferable, the error costs of each class must be specified. As a result of the trade-off for the highest positive accuracy, NB-Rec did not perform well in F1-score. NB-Rec is thus more desirable when the classification accuracy of the positive class (false negatives) cannot be compromised while misclassifying negative instances (false positives) is tolerable. Finally, it was interesting to observe that the two well-established methods SMOTE and BLSMOTE ranked best in precision but showed very low ranking in G-means, F1-score and sensitivity; also, ENN showed the least improvement over the baseline in sensitivity. Thus, these well-established methods are the least suitable solutions for handling the selected imbalanced problems.

Tables 8–12 present the results of Random Forest applied with the same experiment settings. Our NB-based methods ranked top in sensitivity. In particular, Table 12 shows that NB-Rec and NB-Basic provided significantly higher sensitivity than BLSMOTE, ENN and the baseline. However, low precision was observed in some cases. For example, NB-Rec achieved the highest average ranking in sensitivity among all methods but with low precision. NB-Basic provided competitive sensitivity and G-mean with state-of-the-art kmUnder and OBU, however with higher precision and F1-score ranks. NB-Tomek and NB-Comm yielded comparable trade-offs between sensitivity and, G-mean and F1-score, with kmUnder and OBU. Finally, SMOTE, BLSMOTE, and ENN produced the least favourable performance amongst all methods. These results are consistent with the results obtained using SVM, which indicates a stable performance of our methods across different learning algorithms.

6.3. Experiment III: Large and high-dimensional datasets

In this experiment, we aimed at validating the stability of our methods on large and high-dimensional real-world datasets. Table 13 shows the performance of the methods using SVM. In this experiment we compared our methods with the top performing methods in Experiment II based on SVM with emphasis on the class of interest accuracy, namely ENN, kmUnder, and OBU.

On the breast cancer dataset, all of our NB-based methods significantly improved both sensitivity and G-mean from the baseline. They also outperformed ENN and OBU in the two metrics, where NB-Rec yielded the highest results (86.29% in sensitivity and 79.65% in G-mean). As a result of the trade-off, they suffered more from higher false positives as can be seen from lower precision and F1-score. However, their false positive rates were reasonable as evidenced by fair G-mean. Low precision and F1-score obtained were due to the extremely class imbalance nature on a large dataset as well as high class overlap (as can be observed that none of the methods with relatively high sensitivity could simultaneously yield high precision and F1-score, and vice versa). NB-Rec and kmUnder produced the highest sensitivity while the other metrics were comparable.

On the handwritten digits with class 3 as the minority class (MNIST_3), the NB-based methods improved both sensitivity and G-mean. NB-Rec achieved the highest sensitivity of 99.18% and outperformed kmUnder in all metrics. The other NB-

Table 8
Sensitivity values and ranks with RF baseline from Experiment II.

	Sensitivity value/rank																			
Dataset	NB-Basic		NB-Tomek		NB-Comm		NB-Rec		SMOTE		BLSMOTE		ENN		kmUnder		OBU		Baseline	
Wisconsin	100	**1**	**100**	**1**	**100**	**1**	**100**	**1**	91.49	10	95.74	6	95.74	6	95.74	6	**100**	**1**	95.74	6
Pima	**100**	**1**	96.23	3	94.34	4	**100**	**1**	71.7	8	73.58	7	66.04	9	75.47	6	90.57	5	62.26	10
Glass0	**100**	**1**	78.57	3	78.57	3	78.57	3	64.29	7	64.29	7	64.29	7	78.57	3	**100**	**1**	50	10
Vehicle1	**100**	**1**	**100**	**1**	93.02	4	**100**	**1**	65.12	7	62.79	8	55.81	9	74.42	6	90.7	5	51.16	10
Vehicle0	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**
Ecoli1	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	80	6	80	6	80	6	80	6	**100**	**1**	80	6
New-thyroid1	85.71	6	**100**	**1**	**100**	**1**	**100**	**1**	85.71	6	85.71	6	85.71	6	**100**	**1**	**100**	**1**	85.71	6
New-thyroid2	98.2	4	**99.1**	**1**	**99.1**	**1**	**99.1**	**1**	91.89	6	90.99	7	87.39	9	96.4	5	87.39	9	88.29	8
Ecoli2	80	2	80	2	80	2	80	2	80	2	80	2	80	2	80	2	**90**	**1**	80	2
Segmemt0	**100**	**1**	98.46	5	98.46	5	**100**	**1**	98.46	5	**100**	**1**	98.46	5	98.46	5	**100**	**1**	98.46	5
Yeast3	93.75	3	93.75	3	84.38	5	**100**	**1**	68.75	8	84.38	5	62.5	9	**100**	**1**	78.13	7	62.5	9
Ecoli3	**85.71**	**1**	**85.71**	**1**	71.43	7	**85.71**	**1**	**85.71**	**1**	57.14	8	57.14	8	**85.71**	**1**	**85.71**	**1**	42.86	10
Yeast2vs4	90	2	90	2	90	2	90	2	70	7	60	8	50	9	**100**	**1**	80	6	50	9
Vowel0	94.44	3	94.44	3	94.44	3	**100**	**1**	94.44	3	94.44	3	94.44	3	**100**	**1**	94.44	3	94.44	3
Glass2	**100**	**1**	33.33	5	0	6	66.67	4	0	6	0	6	0	6	**100**	**1**	**100**	**1**	0	6
Yeast1vs7	50	3	50	3	50	3	**100**	**1**	33.33	9	33.33	9	50	3	**100**	**1**	50	3	50	3
Glass4	50	4	50	4	50	4	**100**	**1**	**100**	**1**	50	4	50	4	50	4	**100**	**1**	50	4
Ecoli4	50	6	50	6	50	6	75	4	**100**	**1**	75	4	50	6	**100**	**1**	**100**	**1**	50	6
Page-blocks13vs2	80	7	80	7	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	80	7	**100**	**1**	**100**	**1**	80	7
Abalone09-18	50	3	50	3	50	3	**75**	**1**	37.5	7	37.5	7	37.5	7	50	3	**75**	**1**	37.5	7
Glass5	**100**	**1**	0	5	0	5	**100**	**1**	0	5	0	5	0	5	**100**	**1**	**100**	**1**	0	5
Yeast4	80	3	70	5	50	6	90	2	30	7	30	7	20	9	**100**	**1**	80	3	10	10
Ecoli0137vs26	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**
Yeast6	57.14	3	42.86	5	42.86	5	42.86	5	57.14	3	42.86	5	42.86	5	**100**	**1**	71.43	2	42.86	5
Average		2.5		3		3.33		**1.63**		4.92		5.17		5.92		2.5		2.42		6.21

Table 9
G-mean values and ranks with RF baseline from Experiment II.

Dataset	G-mean value/rank																			
	NB-Basic		NB-Tomek		NB-Comm		NB-Rec		SMOTE		BLSMOTE		ENN		kmUnder		OBU		Baseline	
Wisconsin	**99.43**	**1**	**99.43**	**1**	98.86	3	98.28	4	94.56	9	96.73	7	97.29	5	96.73	7	0	10	97.29	5
Pima	48.99	8	54.62	7	57.46	6	0	10	**74.3**	**1**	74.29	2	73.59	4	74.23	3	30.09	9	71.89	5
Glass0	68.14	10	76.76	7	78.57	5	78.57	5	80.18	2	78.73	4	80.18	2	**82.07**	**1**	73.19	8	70.71	9
Vehicle1	45.61	8	49.8	7	59.77	6	26.83	10	73.96	2	71.23	3	71.03	4	**74.81**	**1**	39.95	9	67.4	5
Vehicle0	88.48	9	90.65	7	90.22	8	76.76	10	97.25	3	96.45	5	96.85	4	94.83	6	**97.65**	**1**	**97.65**	**1**
Ecoli1	**93.93**	**1**	**93.93**	**1**	**93.93**	**1**	92.88	5	86.77	9	85.86	10	88.56	6	88.56	6	**93.93**	**1**	88.56	6
New-thyroid1	92.58	6	**100**	**1**	**100**	**1**	98.6	3	92.58	6	92.58	6	92.58	6	95.74	5	97.18	4	92.58	6
New-thyroid2	94.6	4	95.2	2	89.17	9	64.71	10	95.03	3	94.56	5	92.91	7	**95.34**	**1**	92.91	7	93.43	6
Ecoli2	85.36	10	88.64	8	89.44	2	86.19	9	89.44	2	89.44	2	89.44	2	89.44	2	**92.29**	**1**	89.44	2
Segmemt0	98.08	9	99.23	3	98.85	7	99.24	2	99.1	6	**99.87**	**1**	99.23	3	98.85	7	73.43	10	99.23	3
Yeast3	93.08	3	94.03	2	89.92	5	**94.35**	**1**	81.65	8	90.45	4	78.3	10	89.82	6	83.92	7	78.46	9
Ecoli3	80.18	6	84.52	3	80.92	5	83.67	4	**87.83**	**1**	71.71	9	74.96	8	85.36	2	79.28	7	65.47	10
Yeast2vs4	93.31	3	93.83	2	**94.35**	**1**	93.31	3	83.21	7	77.04	8	70.71	9	92.67	5	88.96	6	70.71	9
Vowel0	95.26	7	95.54	6	94.71	8	91.24	10	**97.18**	**1**	**97.18**	**1**	**97.18**	**1**	97.17	5	93.87	9	**97.18**	**1**
Glass2	**86.23**	**1**	53.91	5	0	6	70.41	2	0	6	0	6	0	6	67.94	3	64.05	4	0	6
Yeast1vs7	64.17	7	66.42	6	69.87	5	**90.1**	**1**	56.01	9	57.05	8	70.71	2	48.51	10	70.71	2	70.71	2
Glass4	69.82	3	69.82	3	68.92	8	93.54	2	**98.74**	**1**	69.82	3	69.82	3	68.92	8	67.08	10	69.82	3
Ecoli4	70.71	6	70.71	6	70.71	6	83.81	5	**100**	**1**	86.6	4	70.71	6	96.77	3	**100**	**1**	70.71	6
Page-blocks13vs2	87.39	8	87.39	8	97.12	4	95.94	5	**100**	**1**	**100**	**1**	89.44	6	99.43	3	26.11	10	89.44	6
Abalone09-18	65.62	5	66.18	4	66.45	3	**76.54**	**1**	59.88	10	60.34	9	61.24	6	61.01	8	71.74	2	61.24	6
Glass5	**96.27**	**1**	0	5	0	5	**96.27**	**1**	0	5	0	5	0	5	93.7	4	**96.27**	**1**	0	5
Yeast4	83.96	3	80.84	4	68.32	5	**87.81**	**1**	54.1	7	54.48	6	44.72	8	11.83	10	84.79	2	31.62	9
Ecoli0137vs26	**99.07**	**1**	**99.07**	**1**	**99.07**	**1**	**99.07**	**1**	**99.07**	**1**	**99.07**	**1**	98.13	8	57.74	10	**99.07**	**1**	98.13	8
Yeast6	74.01	3	64.78	8	65.12	6	64.32	9	75.33	2	65.12	6	65.35	4	13.15	10	**80.62**	**1**	65.35	4
Average		5.13		4.46		4.83		4.75		**4.29**		4.83		5.21		5.25		5.13		5.5

Table 10
Precision values and ranks with RF baseline from Experiment II.

	Precision value/rank																			
Dataset	NB-Basic		NB-Tomek		NB-Comm		NB-Rec		SMOTE		BLSMOTE		ENN		kmUnder		OBU		Baseline	
Wisconsin	**97.93**	**1**	**97.93**	**1**	95.95	5	94.04	9	95.59	8	95.78	6	97.84	3	95.78	7	34.99	10	97.84	4
Pima	41.36	8	42.78	7	43.76	6	34.9	10	62.56	3	61.21	4	**66.29**	**1**	59.97	5	35.04	9	66.25	2
Glass0	47.57	10	60.44	8	64.06	6	64.06	6	**100**	**1**	89.74	4	**100**	**1**	72.78	5	51.15	9	**100**	**1**
Vehicle1	30.34	8	31.45	7	34.25	6	27.1	10	58.4	3	53.01	4	**66.73**	**1**	50.87	5	27.52	9	61.18	2
Vehicle0	58.63	9	63.3	7	62.31	8	42.81	10	85	3	81.51	5	83.22	4	75.32	6	**86.86**	**1**	**86.86**	**1**
Ecoli1	71.65	7	71.65	7	71.65	6	68.41	10	80.17	4	75.2	5	92.38	3	**92.38**	**1**	71.65	7	**92.38**	**1**
New-thyroid1	**100**	**1**	**100**	**1**	**100**	**1**	87.5	8	**100**	**1**	**100**	**1**	**100**	**1**	70	10	77.78	9	**100**	**1**
New-thyroid2	87.5	6	87.5	6	**100**	**1**	70	10	**100**	**1**	**100**	**1**	**100**	**1**	77.78	9	87.5	6	**100**	**1**
Ecoli2	62.13	10	89.13	7	**100**	**1**	67.22	9	**100**	**1**	**100**	**1**	**100**	**1**	**100**	**1**	75.47	8	**100**	**1**
Segmemt0	81.41	9	**100**	**1**	95.57	6	91.63	8	98.48	5	98.5	4	**100**	**1**	95.57	7	26.51	10	**100**	**1**
Yeast3	60.43	7	67.06	6	71.42	5	52.9	8	73.68	4	77.46	3	80.28	2	38.98	10	49.46	9	**83.58**	**1**
Ecoli3	28.5	9	37.42	7	49.92	3	35.22	8	49.92	4	39.92	5	79.95	2	39.92	6	27.21	10	**100**	**1**
Yeast2vs4	75.25	9	82.02	7	90.12	3	75.25	8	87.64	5	85.88	6	**100**	**1**	43.81	10	89.02	4	**100**	**1**
Vowel0	70.76	6	73.85	5	65.31	7	37.42	10	**100**	**1**	**100**	**1**	**100**	**1**	64.21	8	58.54	9	**100**	**1**
Glass2	**25.18**	**1**	18.33	2	0	6	18.33	3	0	6	0	6	0	6	13.81	4	12.76	5	0	6
Yeast1vs7	16.54	9	22.91	8	59.77	4	27.09	7	28.38	6	49.77	5	**100**	**1**	8.38	10	**100**	**1**	**100**	**1**
Glass4	56.4	2	56.4	2	39.27	7	34.1	9	**72.12**	**1**	56.4	2	56.4	2	39.27	7	10.52	10	56.4	2
Ecoli4	**100**	**1**	**100**	**1**	**100**	**1**	42.78	10	**100**	**1**	**100**	**1**	**100**	**1**	49.92	9	**100**	**1**	**100**	**1**
Page-blocks13vs2	52.6	6	52.6	6	52.6	8	44.22	9	**100**	**1**	**100**	**1**	**100**	**1**	84.73	5	6.34	10	**100**	**1**
Abalone09-18	18.02	7	19.72	6	20.7	5	17.27	8	34.29	4	43.91	3	**100**	**1**	10.66	10	12.71	9	**100**	**1**
Glass5	37.5	2	0	5	0	5	**37.5**	**1**	0	5	0	5	0	5	26.47	4	37.5	2	0	5
Yeast4	19.32	8	27.27	5	21.13	7	18.26	9	30.37	4	50.44	3	**100**	**1**	3.48	10	21.92	6	**100**	**1**
Ecoli0137vs26	57.98	5	57.98	5	**57.98**	**1**	**57.98**	**1**	**57.98**	**1**	**57.98**	**1**	40.82	9	3.69	10	57.98	5	40.82	8
Yeast6	24.95	7	33.27	6	49.93	4	23.03	8	66.61	3	49.93	4	**74.95**	**1**	2.4	10	16.09	9	74.95	2
Average		6.17		5.13		4.67		7.88		3.17		3.38		2.13		7.04		7		**1.96**

Table 11
F1-score values and ranks with RF baseline from Experiment II.

Dataset	F1-score value/rank																			
	NB-Basic		NB-Tomek		NB-Comm		NB-Rec		SMOTE		BLSMOTE		ENN		kmUnder		OBU		Baseline	
Wisconsin	**98.95**	**1**	**98.95**	**1**	97.92	3	96.91	4	93.48	9	95.74	7	96.77	6	95.74	8	51.65	10	96.77	5
Pima	58.24	8	58.96	7	59.52	6	51.46	9	66.67	2	66.67	2	66.04	4	**66.67**	**1**	50.26	10	64.08	5
Glass0	65.12	10	68.75	7	70.97	5	70.97	5	**78.26**	**1**	75	4	**78.26**	**1**	75.86	3	68.29	8	66.67	9
Vehicle1	46.49	8	47.78	7	50	6	42.57	9	**61.54**	**1**	57.45	4	**60.76**	**2**	60.38	3	42.16	10	55.7	5
Vehicle0	73.58	9	77.23	7	76.47	8	59.54	10	91.76	3	89.66	5	90.7	4	85.71	6	**92.86**	**1**	**92.86**	**1**
Ecoli1	83.33	5	83.33	5	83.33	4	81.08	8	80	9	77.42	10	**85.71**	**1**	85.71	2	83.33	5	85.71	2
New-thyroid1	92.31	7	**100**	**1**	**100**	**1**	93.33	3	92.31	4	92.31	4	92.31	4	82.35	10	87.5	9	92.31	7
New-thyroid2	93.33	6	93.33	6	**100**	**1**	82.35	10	**100**	**1**	**100**	**1**	**100**	**1**	87.5	9	93.33	6	**100**	**1**
Ecoli2	69.57	10	84.21	7	88.89	3	72.73	9	88.89	3	88.89	3	88.89	3	**88.89**	**1**	81.82	8	**88.89**	**1**
Segmemt0	89.66	9	99.22	3	96.97	7	95.59	8	98.46	5	**99.24**	**1**	99.22	2	96.97	6	41.67	10	99.22	3
Yeast3	73.17	4	77.92	2	77.14	3	68.82	8	70.97	6	**80.6**	**1**	70.18	7	55.65	10	60.24	9	71.43	5
Ecoli3	42.86	9	52.17	6	58.82	4	50	7	63.16	2	47.06	8	**66.67**	**1**	54.55	5	41.38	10	60	3
Yeast2vs4	81.82	5	85.71	2	**90**	**1**	81.82	4	77.78	6	70.59	7	66.67	9	60.61	10	84.21	3	66.67	8
Vowel0	80.95	6	82.93	5	77.27	8	54.55	10	**97.14**	**1**	**97.14**	**1**	**97.14**	**1**	78.26	7	72.34	9	97.14	4
Glass2	**37.5**	**1**	22.22	3	0	6	26.67	2	0	6	0	6	0	6	22.22	3	20.69	5	0	6
Yeast1vs7	25	9	31.58	7	54.55	4	42.86	5	30.77	8	40	6	66.67	3	15.58	10	**66.67**	**1**	**66.67**	**1**
Glass4	50	2	50	2	40	8	44.44	7	**80**	**1**	50	2	50	2	40	8	15.38	10	50	2
Ecoli4	66.67	4	66.67	4	66.67	8	54.55	10	**100**	**1**	85.71	3	66.67	8	66.67	4	**100**	**1**	66.67	4
Page-blocks13vs2	61.54	7	61.54	7	66.67	6	58.82	9	**100**	**1**	**100**	**1**	88.89	5	90.91	3	10.87	10	88.89	4
Abalone09-18	25.81	8	27.59	6	28.57	5	27.27	7	35.29	4	40	3	**54.55**	**1**	17.02	10	21.05	9	54.55	2
Glass5	**40**	**1**	0	5	0	5	**40**	**1**	0	5	0	5	0	5	28.57	4	**40**	**1**	0	5
Yeast4	30.77	5	**38.89**	**1**	29.41	8	30	6	30	6	37.5	2	33.33	4	6.62	10	34.04	3	18.18	9
Ecoli0137vs26	**66.67**	**1**	**66.67**	**1**	66.67	4	66.67	4	66.67	4	66.67	4	50	8	5.26	10	**66.67**	**1**	50	8
Yeast6	34.78	7	37.5	6	46.15	4	30	8	**61.54**	**1**	46.15	4	54.55	2	4.7	10	26.32	9	54.55	3
Average		5.92		4.5		4.92		6.79		**3.75**		3.92		**3.75**		6.38		6.58		4.29

Table 12
p-values of the Wilcoxon Signed Rank Tests with RF baseline from Experiment II.

			Sensitivity			
	SMOTE	BLSMOTE	ENN	kmUnder	OBU	Baseline
NB-Basic	8.00E−02	**1.76E−02**	**3.51E−03**	3.87E−01	4.67E−01	**2.08E−03**
NB-Tomek	4.87E−01	2.02E−01	7.98E−02	**3.67E−02**	5.91E−02	5.90E−02
NB-Comm	6.40E−01	3.14E−01	1.38E−01	**2.58E−02**	**4.50E−02**	9.86E−02
NB-Rec	**4.47E−03**	**7.83E−04**	**8.43E−05**	8.29E−01	7.53E−01	**8.02E−05**
			G-mean			
	SMOTE	BLSMOTE	ENN	kmUnder	OBU	Baseline
NB-Basic	8.08E−01	6.62E−01	3.62E−01	1.00E+00	4.85E−01	2.86E−01
NB-Tomek	6.48E−01	9.00E−01	5.93E−01	8.69E−01	7.64E−01	4.85E−01
NB-Comm	5.35E−01	1.00E+00	7.49E−01	7.93E−01	9.38E−01	6.00E−01
NB-Rec	9.46E−01	4.21E−01	1.68E−01	7.15E−01	3.67E−01	1.45E−01
			precision			
	SMOTE	BLSMOTE	ENN	kmUnder	OBU	Baseline
NB-Basic	**4.16E−02**	6.89E−02	**1.04E−03**	6.57E−01	4.15E−01	**7.47E−04**
NB-Tomek	1.25E−01	1.66E−01	**3.85E−03**	4.70E−01	2.74E−01	**2.57E−03**
NB-Comm	2.02E−01	3.05E−01	**6.49E−03**	3.53E−01	1.83E−01	**5.53E−03**
NB-Rec	**7.72E−03**	**5.00E−03**	**8.21E−05**	6.75E−01	8.45E−01	**6.65E−05**
			F1-score			
	SMOTE	BLSMOTE	ENN	kmUnder	OBU	Baseline
NB-Basic	1.83E−01	2.88E−01	2.83E−01	7.41E−01	5.43E−01	3.97E−01
NB-Tomek	3.12E−01	5.09E−01	4.57E−01	6.35E−01	4.03E−01	5.91E−01
NB-Comm	2.97E−01	6.65E−01	6.42E−01	5.50E−01	3.37E−01	7.96E−01
NB-Rec	7.27E−02	1.94E−01	1.24E−01	8.93E−01	6.65E−01	1.76E−01

Table 13
Results on large and high-dimensional datasets.

Dataset	Metric	NB-Basic	NB-Tomek	NB-Comm	NB-Rec	ENN	kmUnder	OBU	Baseline
Breast Cancer	sensitivity	64.52	58.87	60.48	**86.29**	40.32	**86.29**	42.74	28.23
	G-mean	78.83	76.1	76.69	**79.65**	63.48	79.12	65.35	53.12
	precision	9.66	18.02	11.81	1.95	**81.97**	2.77	73.61	81.4
	F1-score	16.81	27.6	19.76	3.81	54.05	5.36	**54.08**	41.92
MNIST_3	sensitivity	94.29	92.65	93.47	**99.18**	82.45	95.92	82.45	82.45
	G-mean	**94.82**	94.56	94.38	77.19	90.71	59.22	90.71	90.71
	precision	31.56	37.58	31.11	5.35	**90.18**	3.32	**90.18**	**90.18**
	F1-score	47.29	53.47	46.69	10.15	**86.14**	6.43	**86.14**	**86.14**
MNIST_5	sensitivity	97.42	97.42	97.42	**99.26**	91.14	95.94	90.96	90.96
	G-mean	96.26	**96.85**	95.52	70.98	95.28	93.85	95.18	95.18
	precision	49.72	56.53	43.28	9.1	**91.99**	36.78	91.98	91.98
	F1-score	65.84	71.54	59.93	16.67	**91.57**	53.17	91.47	91.47

based methods showed competitive sensitivity with significantly higher G-means, precision, and F1-scores than kmUnder. ENN and OBU did not show any improvement over the baseline.

Similarly, on the handwritten digits with class 5 as the minority class (MNIST_5), NB-based methods showed significant improvements in sensitivity while G-mean results were competitive with that of the baseline. NB-Rec had the highest sensitivity of 99.26%. NB-Basic, NB-Tomek, and NB-Comm yielded better results in all metrics than kmUnder. They also performed better than ENN, which rarely showed improvement over the baseline, in both sensitivity and G-mean. OBU did not show any improvement over the baseline.

In summary, the performance of our methods on the large high-dimensional datasets was consistent with the previous experiments. NB-Rec performed best in sensitivity on all of the large high-dimensional datasets and had reasonable true negative rates (as can be observed from G-mean), however highly suffered from high false positives due to the trade-off nature on a large and highly imbalanced dataset. NB-Basic, NB-Tomek, and NB-Comm showed significantly higher improvements over ENN and OBU. They were competitive with kmUnder on average.

7. Conclusions

In this paper, we proposed a novel undersampling approach to handle classification of imbalanced and overlapped datasets by identifying and eliminating potential negative instances in the overlapping region. Four different variants of the proposed approach have been created and extensive experiments using simulated and real-world datasets were carried out. The proposed methods were compared against well-established and state-of-the-art methods. Results showed that our methods achieved the highest sensitivity with competitive G-means across all imbalance degrees on both simulated and real-world datasets. Our methods also showed competitive performance across all degrees of class overlap on the simulated datasets. The four variants of the proposed approach provided different benefits and trade-offs: (1) NB-Rec yielded exceptionally high sensitivity at all degrees of class imbalance and class overlap but had higher false positives at higher class imbalance degrees; (2) NB-Basic resulted in competitive sensitivity with better trade-offs of fewer negative class prediction errors (false positives) than the state-of-the-art methods; (3) NB-Tomek and NB-Comm showed similar trade-offs and were comparable to the state-of-the-art methods in all metrics. These methods provide different options that suit various problem domains.

From our experimental results, a more consistent performance was observed across all simulated datasets whereas some variations were observed in real-world datasets. The difference may be due to the difference in data uniformity. The simulated datasets are uniformly distributed (i.e. data density is uniform across the entire data space), but this cannot be guaranteed in real-world scenarios. Such an issue has not been considered in this work. Thus, a possible future direction includes integrating a density factor into the neighbourhood search criteria of our methods. Another potential solution is to create an adaptive method for setting *k* value in the *k-NN* rule, where the value will be dependent on the local minority class density. For example, a higher *k* value can be used when the local minority class density is lower than the local majority class density, otherwise a lower *k* may be considered. In this work, we only consider the undersampling criteria for binary-class problems. Multi-class datasets were treated as a binary-class problem using the one-vs-all scheme. However, the searching criteria of our methods can be modified and extended to handle imbalanced datasets with more than one minority class. Finally, another interesting direction would be to apply a global algorithm to roughly separate the overlapping and non-overlapping regions, followed by performing a local search. Such an approach could potentially lead to a significant reduction of processing time, which is required for large datasets.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ins.2019.08.062.

References

[1] S. Barua, M.M. Islam, X. Yao, K. Murase, Mwmote–majority weighted minority oversampling technique for imbalanced data set learning, IEEE Trans. Knowl. Data Eng. 26 (2) (2014) 405–425.

[2] M. Bekkar, H.K. Djemaa, T.A. Alitouche, Evaluation measures for models assessment over imbalanced data sets, J. Inf. Eng. Appl. 3 (10) (2013).

[3] P. Branco, L. Torgo, R.P. Ribeiro, A survey of predictive modeling on imbalanced domains, ACM Comput. Surv. (CSUR) 49 (2) (2016) 31.

[4] C. Bunkhumpornpat, K. Sinapiromsaran, Dbmute: density-based majority under-sampling technique, Knowl. Inf. Syst. 50 (3) (2017) 827–850.

[5] C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap, Safe-level-smote: safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2009, pp. 475–482.

[6] C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap, Dbsmote: density-based synthetic minority over-sampling technique, Appl. Intell. 36 (3) (2012) 664–684.

[7] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, Smote: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.

[8] D. Chetchotsak, S. Pattanapairoj, B. Arnonkijpanich, Integrating new data balancing technique with committee networks for imbalanced data: grsom approach, Cognit. Neurodyn. 9 (6) (2015) 627–638.

[9] S. Das, S. Datta, B.B. Chaudhuri, Handling data irregularities in classification: foundations, trends, and future challenges, Pattern Recognit. 81 (2018) 674–693.

[10] D. Devi, B. Purkayastha, et al., Redundancy-driven modified tomek-link based undersampling: a solution to class imbalance, Pattern Recognit. Lett. 93 (2017) 3–12.

[11] G. Douzas, F. Bacao, F. Last, Improving imbalanced learning through a heuristic oversampling method based on k-means and smote, Inf. Sci. 465 (2018) 1–20.

[12] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, An overview of ensemble methods for binary classifiers in multi-class problems: experimental study on one-vs-one and one-vs-all schemes, Pattern Recognit. 44 (8) (2011) 1761–1776.

[13] V. García, R.A. Mollineda, J.S. Sánchez, On the k-nn performance in a challenging scenario of imbalance and overlapping, Pattern Anal. Appl. 11 (3–4) (2008) 269–280.

[14] J. Gong, H. Kim, Rhsboost: improving classification performance in imbalance data, Comput. Stat. Data Anal. 111 (2017) 1–13.

[15] Q. Gu, L. Zhu, Z. Cai, Evaluation measures of the classification performance of imbalanced data sets, in: International Symposium on Intelligence Computation and Applications, Springer, 2009, pp. 461–471.

[16] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, G. Bing, Learning from class-imbalanced data: review of methods and applications, Expert Syst. Appl. 73 (2017) 220–239.

[17] H. Han, W.-Y. Wang, B.-H. Mao, Borderline-smote: a new over-sampling method in imbalanced data sets learning, in: International Conference on Intelligent Computing, Springer, 2005, pp. 878–887.

[18] H. He, Y. Bai, E.A. Garcia, S. Li, Adasyn: adaptive synthetic sampling approach for imbalanced learning, in: Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on, IEEE, 2008, pp. 1322–1328.

[19] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. (9) (2008) 1263–1284.

[20] N. Japkowicz, S. Stephen, The class imbalance problem: a systematic study, Intell. Data Anal. 6 (5) (2002) 429–449.

[21] L.A. Jeni, J.F. Cohn, F. De La Torre, Facing imbalanced data–recommendations for the use of performance metrics, in: 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction, IEEE, 2013, pp. 245–251.

[22] T. Jo, N. Japkowicz, Class imbalances versus small disjuncts, ACM Sigkdd Explorations Newslett. 6 (1) (2004) 40–49.

[23] M. Koziarski, M. Woźniak, Ccr: a combined cleaning and resampling algorithm for imbalanced data classification, Int. J. Appl. Math. Comput. Sci. 27 (4) (2017) 727–736.

[24] B. Krawczyk, M. Galar, Ł. Jeleń, F. Herrera, Evolutionary undersampling boosting for imbalanced classification of breast cancer malignancy, Appl. Soft Comput. 38 (2016) 714–726.

[25] J. Laurikkala, Improving identification of difficult small classes by balancing class distribution, in: Conference on Artificial Intelligence in Medicine in Europe, Springer, 2001, pp. 63–66.

[26] H.K. Lee, S.B. Kim, An overlap-sensitive margin classifier for imbalanced and overlapping data, Expert Syst. Appl. (2018).

[27] W.-C. Lin, C.-F. Tsai, Y.-H. Hu, J.-S. Jhang, Clustering-based undersampling in class-imbalanced data, Inf. Sci. 409 (2017) 17–26.

[28] V. López, S. del Río, J.M. Benítez, F. Herrera, Cost-sensitive linguistic fuzzy rule based classification systems under the mapreduce framework for imbalanced big data, Fuzzy Sets Syst. 258 (2015) 5–38.

[29] V. López, A. Fernández, S. García, V. Palade, F. Herrera, An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics, Inf. Sci. 250 (2013) 113–141.

[30] I. Nekooeimehr, S.K. Lai-Yuen, Adaptive semi-unsupervised weighted oversampling (a-suwo) for imbalanced datasets, Expert Syst. Appl. 46 (2016) 405–416.

[31] W.W. Ng, J. Hu, D.S. Yeung, S. Yin, F. Roli, Diversified sensitivity-based undersampling for imbalance classification problems, IEEE Trans. Cybern. 45 (11) (2015) 2402–2412.

[32] H. Patel, G.S. Thakur, Classification of imbalanced data using a modified fuzzy-neighbor weighted approach, Int. J. Intell. Eng. Syst. 10 (1) (2017) 56–64.

[33] R. Rifkin, A. Klautau, In defense of one-vs-all classification, J. Mach. Learn. Res. 5 (January) (2004) 101–141.

[34] J.A. Sáez, J. Luengo, J. Stefanowski, F. Herrera, Smote–ipf: addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering, Inf. Sci. 291 (2015) 184–203.

[35] C. Seiffert, T.M. Khoshgoftaar, J. Van Hulse, A. Napolitano, Rusboost: a hybrid approach to alleviating class imbalance, IEEE Trans. Syst. ManCybern.-Part A 40 (1) (2010) 185–197.

[36] J. Stefanowski, Overlapping, rare examples and class decomposition in learning classifiers from imbalanced data, in: Emerging Paradigms in Machine Learning, Springer, 2013, pp. 277–306.

[37] H. Sun, S. Wang, Measuring the component overlapping in the gaussian mixture model, Data Min. Knowl. Discov. 23 (3) (2011) 479–502.

[38] Z. Sun, Q. Song, X. Zhu, H. Sun, B. Xu, Y. Zhou, A novel ensemble method for classifying imbalanced data, Pattern Recognit. 48 (5) (2015) 1623–1637.

[39] S. Visa, A. Ralescu, Learning imbalanced and overlapping classes using fuzzy sets, in: Proceedings of the ICML, 3, 2003.

[40] P. Vorraboot, S. Rasmequan, K. Chinnasarn, C. Lursinsap, Improving classification rate constrained to imbalanced data between overlapped and non-overlapped regions by hybrid algorithms, Neurocomputing 152 (2015) 429–443.

[41] P. Vuttipittayamongkol, E. Elyan, A. Petrovski, C. Jayne, Overlap-based undersampling for improving imbalanced data classification, in: International Conference on Intelligent Data Engineering and Automated Learning, Springer, 2018, pp. 689–697.

[42] S. Wang, X. Yao, Diversity analysis on imbalanced data sets by using ensemble models, in: Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on, IEEE, 2009, pp. 324–331.

[43] D.L. Wilson, Asymptotic properties of nearest neighbor rules using edited data, IEEE Trans. Syst. Man Cybern. (3) (1972) 408–421.

[44] H. Yu, K. Liu, Classification of multi-class microarray datasets using a minimizing class-overlapping based ecoc algorithm, in: Proceedings of the 5th International Conference on Bioinformatics and Computational Biology, ACM, 2017, pp. 51–54.

[45] Y. Zhao, Y. Cen, Data mining applications with R, Academic Press, 2013.

[46] Y. Zhu, Z. Wang, H. Zha, D. Gao, Boundary-eliminated pseudoinverse linear discriminant for imbalanced problems, IEEE Trans. Neural Netw. Learn. Syst. 29 (6) (2018) 2581–2594.