

# Neighbourhood-based undersampling approach for handling imbalanced and overlapped data.

VUTTIPITTAYAMONGKOL, P., ELYAN, E.

2020



# 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*

---

## 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 maximises 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.

*Keywords:* Imbalanced dataset, Undersampling, k-NN, Class overlap, Classification

---

\*Corresponding author

*Email addresses:* [p.vuttipittayamongkol@rgu.ac.uk](mailto:p.vuttipittayamongkol@rgu.ac.uk) (Pattaramon Vuttipittayamongkol), [e.elyan@rgu.ac.uk](mailto:e.elyan@rgu.ac.uk) (Eyad Elyan)

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

## 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].

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, 27, 6], 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 [39, 9, 36, 13]. A recent study [41] supported these findings by showing significant improvement over state-of-the-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 Fig. 1b, 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.

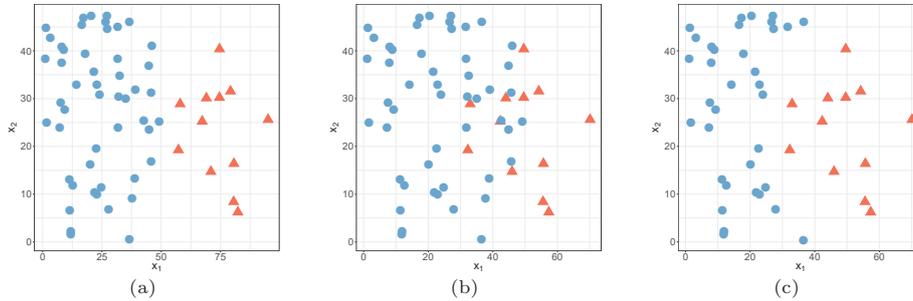


Figure 1: Synthetic datasets with (a) imbalanced class distributions, (b) class imbalance and overlap problems, (c) negative instances removed from the overlapping region

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.

- 1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20
- 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.

21  
22  
23  
24  
25  
26  
27  
28  
29

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.

## 30 31 **2. Related Work**

32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54

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 an advantage of directly incorporating the user's preferences into the model [3]. However, as opposed to data resampling, a learning algorithm need 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 [16]. Additional recent algorithm-level and ensemble-based methods can be seen in [46, 32, 29, 14, 35, 42].

55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

The class-imbalance issue has led to the development of well-established data-level solutions. However, for linearly separable or sufficiently large datasets,

1  
2  
3  
4  
5  
6  
7  
8  
9 it was shown that results were not affected by any degree of class imbalance  
10 [20]. Other studies showed that class overlap had a higher negative impact on  
11 models performance than class imbalance [39, 9, 36, 13]. Thus, in this paper,  
12 we broadly categorise existing solutions as class distribution-based methods and  
13  
14  
15  
85 class overlapped-based methods.

16  
17 Random resampling is the most widely-used class distribution-based approach  
18 due to the simplicity of application. However, as balanced class distribution is  
19 typically the stopping criteria, random undersampling may lead to information  
20 loss while random oversampling is often prone to overfitting [28]. To replace  
21  
22  
23  
90 random selection, Synthetic Minority Over-sampling Technique (SMOTE) [7]  
24 was introduced. The method generates new minority class instances by means  
25 of linear interpolation between neighbouring points. SMOTE has led to many  
26 well-known extensions [17, 5, 6, 34, 11]. Some of these extensions, *e.g.* Borderline-  
27 SMOTE [17] and SMOTE-IPF [34], focused on resampling instances within the  
28  
29  
30  
95 overlapping region. In contrast, other extensions such as Safe-level-SMOTE [5]  
31 and DBSMOTE [6], avoided resampling hard-to-classify instances and showed  
32 less improvements compared to their counterpart overlap-based methods, in  
33 particular, Borderline-SMOTE [17] and DBMUTE [4]. Other recent methods  
34 based on clustering [27, 31] and Neural Network [8] have also been proposed.  
35  
36  
37  
38  
100 These methods attempted to preserve the topology of the original data while  
39 rebalancing class distribution. A common drawback of class distribution-based  
40 methods is that the resampling rate is limited by imbalance degree. If the class  
41 imbalance is low, a small number of minority class instances generated may  
42 not be sufficient to emphasise the presence of the minority class near the class  
43  
44  
45  
46  
105 boundary. On the other hand, if the class imbalance is high, a significant amount  
47 of important information may be lost.  
48  
49

50 Since the class overlap problem of imbalanced datasets is focused in this  
51 paper, existing class overlap-based methods are extensively reviewed in the  
52 following subsection.  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

1  
2  
3  
4  
5  
6  
7  
8  
9  
110 *2.1. Class Overlap-based Methods*

10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
120  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

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 sacrificing some information becomes essential.

Examples of methods that address the entire overlapping region are [41, 18, 4]. A recent overlapped-based method, OBU [41], based on 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 with 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 [43]. ENN selectively removes

1  
2  
3  
4  
5  
6  
7  
8  
9 majority class instances by considering its  $k$  nearest neighbours that belong to  
10 the other class, where  $k = 3$ . It has to be noted that the setting of the  $k$  value  
11 in this approach significantly impacts the performance. For example, a small  
12  $k$  value can leave a lot of the overlapped majority class instances unremoved.  
13  
14  
15 Neighbourhood Cleaning Rule (NCL) [25] is an extension of this method, where  
16 the  $k$  nearest neighbours of both positive and negative instances were considered  
17 for the removal of negative instances. Results showed an improvement over a  
18 data-distribution based method proposed in [22]. Later overlap-based methods  
19 showed superior results over NCL. These include a combined cleaning and re-  
20 sampling approach, which consisted of neighbourhood cleaning and selectively  
21 oversampling positive instances in the overlapping region [23], and Evolution-  
22 ary undersampling, which employed an Evolutionary algorithm to significantly  
23 reduce the dataset size and achieve optimal classification results [24].  
24  
25  
26  
27  
28

29 A method to avoid erroneous eliminations of positive instances during noise  
30 cleaning was proposed in SMOTE-IPF [34]. This was done by first applying  
31 SMOTE, followed by applying a noise filtering algorithm. SMOTE-IPF allows  
32 new minority class instances to be generated before determining noisy instances.  
33 By doing so, fewer rare cases of the minority class will appear as noise, hence  
34 reducing erroneous eliminations, which is crucial for preserving highly important  
35 information. In Borderline-SMOTE [17] (BLSMOTE), synthetic instances were  
36 created from the borderline minority class instances and their nearest neigh-  
37 bours. Based on this approach, two methods were proposed. One considers  
38 only the minority-class nearest neighbours while the other includes the nearest  
39 neighbours of both classes in generating new instances. Results showed that the  
40 latter method whose synthetic instances were generated closer to the borderline  
41 achieved higher true positive rates. Redundancy-driven modified Tomek-link  
42 based undersampling [10] considered similarity and contribution factors as the un-  
43 dersampling criteria. This is achieved by removing redundant negative instances  
44 with the lowest contributions to classification. The undersampling process is  
45 terminated when a data balance is reached. In other words, the imbalance degree  
46 indeed dominates the proposed elimination criteria, which could eventually lead  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

1  
2  
3  
4  
5  
6  
7  
8  
9 to an insufficient elimination or an excessive elimination of negative instances.

10 In Majority Weighted Minority Oversampling Technique (MWMOTE) [1]  
11 and Adaptive Semi-Unsupervised Weighted Oversampling (A-SUWO) [30], new  
12 instances are synthesized within the sub-clusters of the minority class. Both  
13  
14 175 instances are synthesized within the sub-clusters of the minority class. Both  
15 methods used a bottom-up clustering technique that does not require the number  
16 of clusters to be fixed a prior. Each minority class instance was weighted with the  
17 selection probability based on its proximity to the majority class. This proximity  
18 was calculated differently in the two methods. MWMOTE assigned higher  
19 weights to minority class instances that are closer to the predefined borderline  
20 majority class instances while A-SUWO put higher weights on instances with  
21 more majority class nearest neighbours. Another weighting factor MWMOTE  
22 180 determined was the density of a sub-cluster. More synthetic positive instances  
23 were created in sparse sub-clusters. On the other hand, A-SUWO synthesised  
24 more instances in the sub-clusters with higher misclassification errors to handle  
25 small sub-clusters, which were the results of the within-class imbalance issue.  
26  
27  
28  
29  
30  
31  
32

### 34 **3. Problem Statement**

35  
36 It was illustrated earlier in Fig. 1a and Fig. 1b how class overlap adds  
37 difficulties to classification tasks of a dataset with imbalanced class distribution.  
38 Also, performance of a learning algorithm on imbalanced datasets was shown to  
39 190 be highly dependant on the level of class overlap [13]. In this section, we express  
40 how class overlap is quantified. Detailed discussion of how we extended solution  
41 that deals with borderline instances to cover instances in the entire overlapping  
42 region is also provided.  
43  
44  
45  
46  
47  
48

#### 49 195 *3.1. Quantification of Class Overlap*

50 Since class overlap is not yet mathematically well-defined [37], several methods  
51 to estimate class overlap have been proposed, such as in [44, 26, 37]. However,  
52 these methods have some limitations, including a prior assumption of normal  
53 distribution of data, which are not generally applicable to real-world datasets.  
54  
55  
56  
57  
58

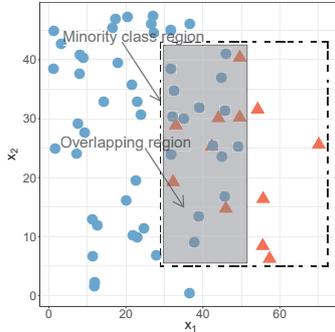


Figure 2: Regions approximation

200 To facilitate the measurement of class overlap degree later in the Experiments  
 201 Section, we adapted the formula used in [13]. The formula was modified to  
 202 suit class imbalanced problems as expressed in Eq. 1, and how the regions are  
 203 approximated for the calculation is shown in Fig. 2(a). Note that this formula is  
 204 only designed for binary-class datasets with 2 dimensional features for simplicity  
 205 of exploring the problem.

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

### 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  
 210 three nearest neighbours are of the minority class. Now that the minority class  
 instances in the overlapping region is 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  
 215 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

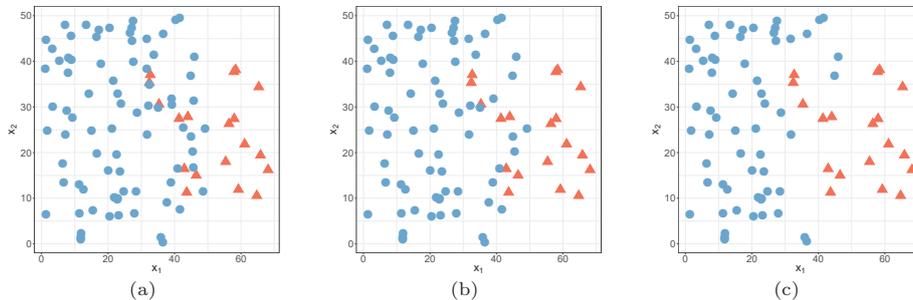


Figure 3: Undersampling solutions to (a) imbalanced and overlapped dataset with (b) borderline instances removed and (c) overlapped instances removed

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 as can be seen in Fig. 2. In order 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, 2, 3 and 4 depict these methods <sup>1</sup>.

#### 4.1. Basic Neighbourhood Search

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

---

**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 ( $\hat{T}$ )

```

---

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

---

<sup>1</sup>Source code available at <https://github.com/fonkafon/NB-undersampling.git>

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

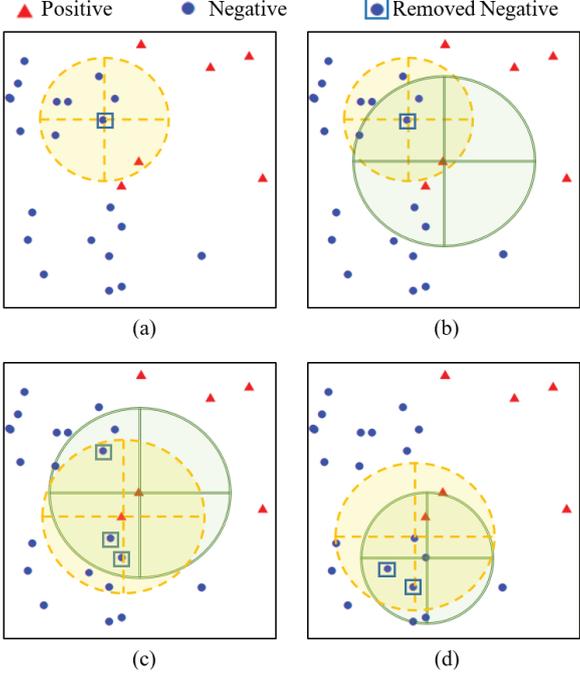


Figure 4: The proposed neighbourhood-based undersampling methods (a) NB-Basic (b) NB-Tomek (c) NB-Comm (d) NB-Rec

4.2. Modified Tomek Link Search

Modified Tomek Link Search is proposed as an extension of NB-Basic to  
 255 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  
 260 modified Tomek Link is eliminated (Fig. 4(b)).

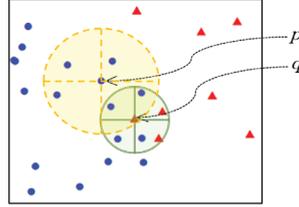


Figure 5: Neighbourhood is *not* established in both directions

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  $q$ .

---

**Algorithm 2:** ModifiedTomek 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 ( $\hat{T}$ )

```

---

### 4.3. Common Nearest Neighbours Search

265 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 an elimination of a negative instance. The  
 270 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

275 is much lower than that of the majority class, fewer common nearest negative  
 280 neighbours instances would be found.

---

**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 ( $\hat{T}$ )

```

---

#### 4.4. Recursive Search

NB-Rec is proposed as an extension of NB-Comm to ensure sufficient and  
 280 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  
 285 criteria is 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  
 290 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.

---

**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;

$NN_{2neg} \leftarrow$  negative members of  $NN_2$ ;

**foreach**  $y \in NN_{2neg}$  **do**

$A'_y.freq \leftarrow A'_y.freq + 1$ ;

**foreach**  $x_2 \in A'.instance$  **do**

**if**  $A'_{x_2}.freq > 1$  **then**

$\hat{X}_2 \leftarrow X_2 \cup \{x_2\}$ ;

$\hat{T} \leftarrow T - (X \cup \hat{X}_2)$ ;

**return** ( $\hat{T}$ )

---

### 5.1. Setup

295 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  
300 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.  
305 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  
310 classification [16].

### 5.2. Datasets

In Experiment I, 66 uniformly-distributed binary-class datasets were simulated. These datasets captures wide ranges of class-overlap and imbalance

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	Segment0	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

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.

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

Table 1 shows the public datasets that were used in Experiment II. These datasets were obtained from *UCI Repository*<sup>2</sup> and *KEEL Repository*<sup>3</sup>. The datasets vary in terms of imbalance degrees (1.86-41.4), number of features

<sup>2</sup><https://archive.ics.uci.edu/ml/index.php>

<sup>3</sup><http://sci2s.ugr.es/keel/imbanced.php>

1  
2  
3  
4  
5  
6  
7  
8  
9 (5-18), and number of instances (214-5,472).

10  
11 In Experiment III, we used the breast cancer dataset from *KDD Cup 2008*<sup>4</sup>  
12 and the handwritten digits dataset from *the MNIST database*<sup>5</sup>. The breast  
13 cancer dataset is 117-feature, binary-class and contains 102,294 samples with  
14 101,671 negative and 623 positive samples, which makes  $imb = 163.20$ . The  
15 handwritten digits dataset is 10-class with 784 features, and contains 60,000  
16 samples. Class 3 and class 5 were selected as the minority class in two scenarios,  
17 and were undersampled to obtained higher class imbalance degrees. In the first  
18 scenario, class 3 was undersampled such that  $imb = 43.90$ , which consists of  
19 53,869 negative and 1,227 positive instances. In the second scenario, class 5  
20 was undersampled such that  $imb = 20.13$ , which consists of 53,869 negative and  
21 2,711 positive instances.  
22  
23  
24  
25  
26  
27

28 In all experiments, each dataset was partitioned into 80 : 20 ratio of training  
29 and testing sets. In Experiment I and II, 10-fold cross-validation was used in the  
30 training phase for the purpose of model selection whereas no cross-validation  
31 was applied to the large datasets in Experiment III.  
32  
33  
34

### 35 5.3. Parameter Settings

36  
37 No parameters tuning or optimisation was performed in our experiments.  
38 This allows us to provide a fair comparison and assess our methods. In the  
39 proposed NB-based methods,  $k$  is an important parameter, where  $k$ -NN is used to  
40 investigate the surroundings of instances. A simple rule of thumb, where  $k$  is set  
41 to equal the square root of the dataset size, was considered. Furthermore, to take  
42 into account the class imbalance issue and promote the discovery of overlapped  
43 majority class instances, we adjusted the  $k$  value to also be proportional to the  
44 imbalance degree as can be seen in Eq. 3, where  $N$  is the number of instances in  
45 the dataset.  
46  
47  
48  
49  
50  
51  
52

$$k = \sqrt{N} + \sqrt{imb} \quad (3)$$

53  
54  
55 <sup>4</sup><https://www.kdd.org/kdd-cup/view/kdd-cup-2008>

56 <sup>5</sup><http://yann.lecun.com/exdb/mnist/>

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

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 [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*<sup>6</sup> 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 [21, 15, 19, 2, 3]. 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.

$$sensitivity = \frac{TP}{TP + FN} \quad (4)$$

$$G - mean = \sqrt{specificity * sensitivity}, \quad (5)$$

where

$$specificity = \frac{TN}{TN + FP} \quad (6)$$

---

<sup>6</sup><https://CRAN.R-project.org/package=caret>

Precision (Eq. 7) and F1-score (Eq. 8) provide good measures to evaluate the trade-offs between positive class accuracy and negative class errors. Precision, which considers the raw number of incorrectly classified negative instances, might potentially be a misleading metric by underestimating the performance when data is highly imbalanced. Consider, for instance, a dataset of 10 : 1,000 (positive : negative) distribution. Classification results of 10 true positives and 10 false positives resulting in 100% sensitivity and 99% specificity, which 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.

$$precision = \frac{TP}{TP + FP} \quad (7)$$

$$F1 - score = \frac{2}{\frac{1}{sensitivity} + \frac{1}{precision}} \quad (8)$$

## 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

1  
2  
3  
4  
5  
6  
7  
8  
9 highest sensitivity and competitive G-mean with nearly 100% true positive  
10 rates were achieved, which was interestingly relatively stable across all datasets  
11  
12 395 regardless of class imbalance and class overlap degrees. This is clearly illustrated  
13 in Fig. 6, where results of our methods are presented with solid lines, the results  
14 of the other methods are marked with dashed lines, and the shaded areas are  
15 the areas under the performance curves of the baseline (SVM).  
16  
17

18  
19 Among the proposed NB-based methods, NB-Rec showed the highest sen-  
20 400 sitivity (99.95%) and competitive G-mean, but least tolerable to information  
21 loss, resulting in the lowest precision and F1-scores. NB-Comm showed slightly  
22 better overall results than NB-Basic and NB-Tomek. Detailed discussion of  
23 these results are provided in the following subsections. Numerical results of this  
24 experiment are provided in the supplementary material<sup>7</sup>.  
25  
26  
27  
28

#### 29 405 *6.1.1. NB-based methods VS class-distribution based methods*

30  
31 The superior performance in sensitivity achieved by our methods in com-  
32 parison with the class-distribution based methods namely, SMOTE [7]<sup>8</sup> and  
33 kmUnder [27], strongly suggests that our NB-based methods promoted the  
34 visibility of the positive class across different class imbalance and class overlap  
35 degrees. Moreover, while our methods provided relatively stable sensitivity under  
36 410 different scenarios, sensitivity of other methods tended to drop when the class  
37 overlap degree increased.  
38  
39  
40

41 The NB-based methods not only produced the highest sensitivity but also  
42 showed competitive G-mean with SMOTE, and produced higher overall G-mean  
43 than kmUnder. These improvements in terms of G-mean and sensitivity indicate  
44 415 that our methods had improved the trade-offs between sensitivity and specificity,  
45 which means we have reduced both false positives and false negatives, over  
46 state-of-the-art kmUnder across different ranges of class imbalance and class  
47 overlap degrees. It was observed that for low imbalanced datasets ( $imb = 1.5, 3$ ),  
48  
49  
50  
51  
52

53  
54 <sup>7</sup>[https://github.com/fonkafon/NB-undersampling\\_Results.git](https://github.com/fonkafon/NB-undersampling_Results.git)

55 <sup>8</sup>In Fig. 6, SMOTE has similar performance in sensitivity to kmUnder (hence the line is  
56 not visible)  
57

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

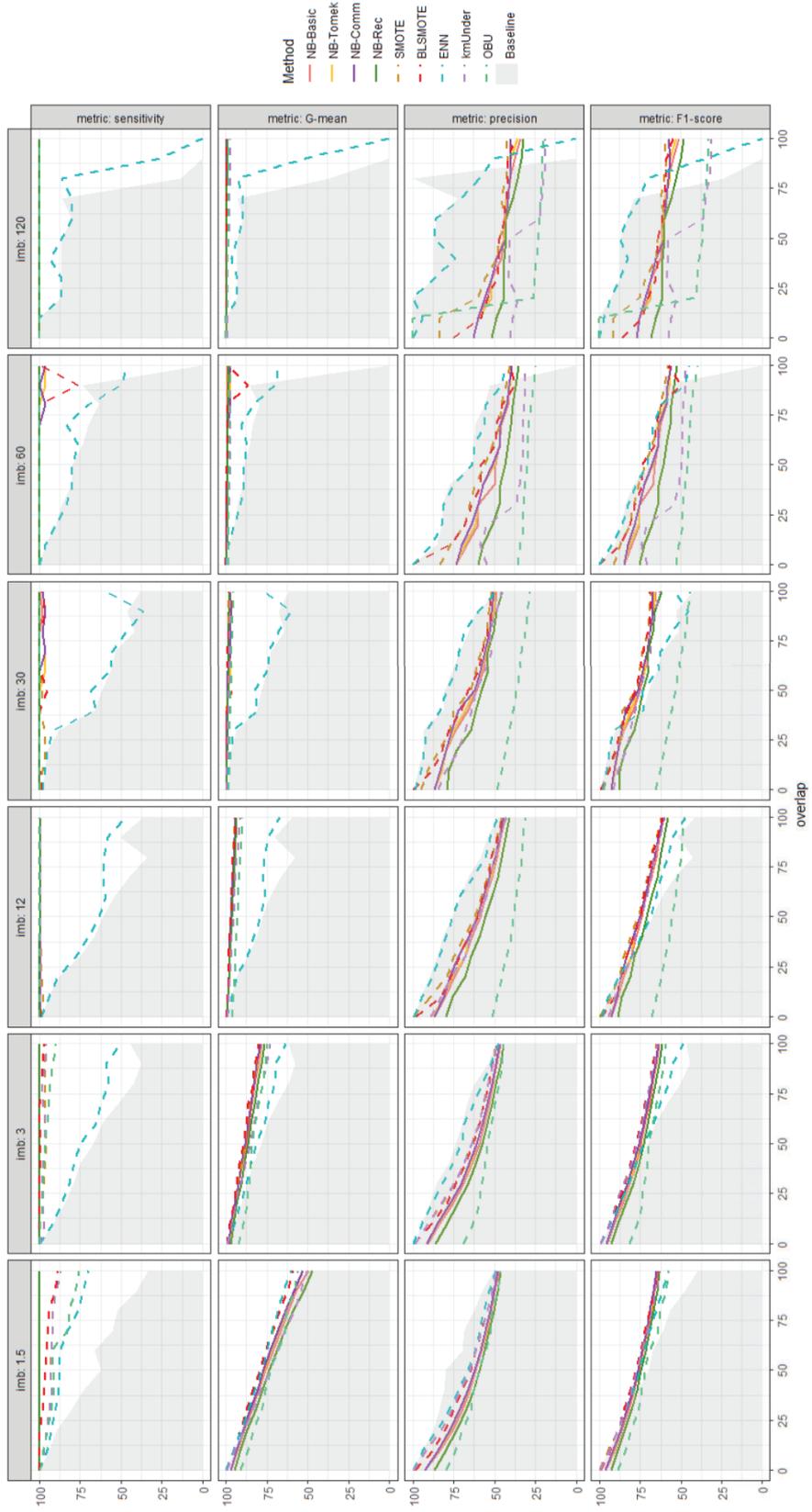


Figure 6: Performance of methods across different imbalance and overlap degrees

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

420 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  
425 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  
430 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 BLSTMOTE  
[17], ENN [43], and OBU [41]. All NB-based methods have competitive results  
435 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. It suggests that our methods had relatively reduced both  
false positives and false negatives by the more accurate detection of potential  
440 overlapped negative instances and minimisation of information loss over the  
similar approach offered by OBU. Compared to BLSTMOTE, 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  
445 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 amount 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.

1  
2  
3  
4  
5  
6  
7  
8  
9  
450 *6.1.3. Overall results*

10  
11 Table 2 shows the average performance of the methods across all imbalance  
12 and overlap degrees. Winning numbers are presented in **bold**. Our NB-based  
13 methods produced exceptionally high sensitivity. NB-Rec achieved the highest  
14 average sensitivity of 99.95%. Such high sensitivity is required across different  
15 domains, especially in medical. Comparable results with SMOTE and BLSMOTE  
16 were achieved in terms of G-means, but with lower precision and recall. It was also  
17 observed that NB-Rec produced the lowest precision and recall when compared  
18 with the other proposed NB-based methods. This suggests that maximising  
19 the visibility of positive instances may cost an increase in the false positives.  
20  
21 Our NB-based methods significantly outperformed ENN in sensitivity and G-  
22 mean with comparable F1-score (except F1-score of NB-Rec). More importantly,  
23 our methods outperformed state-of-the-art kmUnder and OBU in all measures,  
24 except for precision of NB-Rec that was competitive with kmUnder.  
25  
26  
27  
28  
29  
30  
31

32 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	<b>99.95</b>	98.11	98.56	75.52	98.35	97.46	67.75
G-mean	92.93	93.09	93.19	92.18	93.87	<b>93.78</b>	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	<b>74.04</b>
F1-score	73.05	73.66	74.03	69.65	<b>76.41</b>	75.59	73.59	68.54	57.57	69.88

33  
34  
35  
36  
37  
38  
39  
40  
41 *6.2. Experiment II: Real-world datasets*

42  
43 In this experiment, our methods were evaluated on real-world datasets.  
44 Tables 3 to Table 6 show performance of our methods against other methods  
45 on the UCI datasets using SVM, where the datasets are sorted by imbalance  
46 ratio from low to high. These tables also show methods ranks and average  
47 ranking based on their performance (i.e. rank 1 means top performance and so  
48 on). Wilcoxon Signed Rank Tests were carried to assess statistical significance  
49 of the difference in methods performance. Results are presented in Table 7,  
50 with significance level of 0.05, the  $p$  values indicating a statistically significant  
51 difference between two methods are highlighted in **bold**.  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

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

Dataset	Sensitivity Value/Rank																			
	NB-Basic	NB-Tomek	NB-Comm	NB-Rec	SMOTE	BLSMOTE	ENN	kmUnder	OBU	Baseline										
Wisconsin	100	1	100	1	100	1	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
Segment0	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
Ahalame09-18	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
<b>Average</b>	1.74		2.22		2.17		1.61		6.74		7.3		5.65		3.43		2.09		6.39	

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 competitive 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

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

Dataset	G-mean Value/Rank									
	NB-Basic	NB-Tomek	NB-Comm	NB-Rec	SMOTE	BLSMOTE	ENN	kmUnder	OBU	Baseline
Wisconsin	<b>97.12</b> 1	<b>97.12</b> 1	<b>97.12</b> 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	<b>67.8</b> 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	<b>80.34</b> 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	<b>70.4</b> 1	38.02 10	51.54 5
Vehicle0	85.82 7	<b>89.94</b> 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	<b>93.93</b> 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	<b>100</b> 1	89.75 4	75.59 7
New-thyroid2	95.74 4	95.74 4	95.74 4	88.19 10	<b>100</b> 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	<b>94.87</b> 1	92.29 6	78.15 10	89.44 7
Segment0	87.58 8	92.91 7	94 6	87.15 9	97.54 5	<b>98.85</b> 1	97.67 3	97.95 2	84.64 10	97.67 3
Yeast3	90.66 3	<b>93.74</b> 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	<b>94.87</b> 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	<b>88.47</b> 1	82.29 3	63.25 8
Vowel0	94.28 4	94.28 4	<b>99.72</b> 1	98.31 3	84.98 10	94.28 4	94.28 4	94.28 4	<b>99.72</b> 1	94.28 4
Glass2	73.96 2	<b>76.24</b> 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	<b>64.17</b> 1	51.26 5	0 7
Glass4	70.71 3	70.71 3	70.71 3	<b>82.16</b> 1	70.71 3	70.71 3	70.71 3	<b>82.16</b> 1	70.71 3	70.71 3
Ecoli4	99.2 7	99.2 7	<b>100</b> 1	98.4 9	<b>100</b> 1	86.6 10	<b>100</b> 1	<b>100</b> 1	<b>100</b> 1	<b>100</b> 1
Page-blocks13vs2	97.7 5	97.7 5	97.7 5	89.19 10	<b>100</b> 1	89.44 9	99.43 3	97.12 8	<b>100</b> 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	<b>67</b> 1	0 8
Glass5	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -	0 -
Yeast4	81.42 3	83.79 2	<b>84.29</b> 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	<b>100</b> 1	99.07 7	<b>100</b> 1	<b>100</b> 1	<b>100</b> 1	96.23 10	<b>100</b> 1	<b>100</b> 1
Yeast6	90.97 2	<b>91.13</b> 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
<b>Average</b>	4.52	4	<b>3.13</b>	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

Dataset	Precision Value/Rank									
	NB-Basic	NB-Tomek	NB-Comm	NB-Rec	SMOTE	BLSMOTE	ENN	kmUnder	OBU	Baseline
Wisconsin	91.18 8	91.1 9	91.77 7	<b>92.15</b> 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	<b>98.17</b> 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	<b>88.61</b> 1	68.14 7	56.38 10	85.37 2
Vehicle1	72.39 2	68.51 4	69.37 3	<b>87.47</b> 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	<b>93.44</b> 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	<b>100</b> 1
New-thyroid1	75.31 8	75.31 8	76.89 7	83.44 6	<b>100</b> 1	<b>100</b> 1	<b>100</b> 1	<b>100</b> 1	65.49 10	<b>100</b> 1
New-thyroid2	74.01 6	73.68 7	76.54 5	70.79 9	<b>100</b> 1	<b>100</b> 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	<b>100</b> 1	<b>100</b> 1	<b>100</b> 1	75.47 9	42.99 10	<b>100</b> 1
Segment0	52.22 8	62.07 7	65.29 6	51.49 10	98.43 3	95.57 4	<b>100</b> 1	94.09 5	51.98 9	<b>100</b> 1
Yeast3	57.27 7	60.79 5	52.58 9	56.05 8	71.79 4	<b>76.51</b> 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	<b>66.59</b> 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	<b>100</b> 1	<b>100</b> 1	80.21 4	30.33 10	<b>100</b> 1
Vowel0	<b>100</b> 1	<b>100</b> 1	96.1 9	89.65 10	<b>100</b> 1	<b>100</b> 1	<b>100</b> 1	<b>100</b> 1	97.28 8	<b>100</b> 1
Glass2	44.07 3	41.41 4	31.9 6	33.58 5	<b>100</b> 1	<b>100</b> 1	0 9	14.72 7	11.44 8	0 9
Yeast1vs7	17.91 2	8.88 5	15.44 4	<b>25.15</b> 1	0 7	0 7	0 7	16.54 3	4.56 6	0 7
Glass4	<b>100</b> 1	<b>100</b> 1	<b>100</b> 1	39.89 8	<b>100</b> 1	<b>100</b> 1	<b>100</b> 1	16.6 9	9.94 10	<b>100</b> 1
Ecoli4	82.62 8	82.35 9	<b>100</b> 1	72.31 10	<b>100</b> 1	<b>100</b> 1	<b>100</b> 1	<b>100</b> 1	<b>100</b> 1	<b>100</b> 1
Page-blocks13vs2	62.09 7	61.78 8	63.57 6	33.32 10	<b>100</b> 1	<b>100</b> 1	84.73 4	52.6 9	<b>100</b> 1	84.73 4
Abalone09-18	14.85 4	14.77 5	21.62 2	19.66 3	<b>34.29</b> 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	<b>100</b> 1
Ecoli0137vs26	47.93 8	46.69 9	<b>100</b> 1	66.67 7	<b>100</b> 1	<b>100</b> 1	<b>100</b> 1	25.64 10	<b>100</b> 1	<b>100</b> 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	<b>100</b> 1
<b>Average</b>	5.52	5.87	5.35	6.13	<b>3</b>	3.09	3.3	6.22	7.57	3.22

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

Dataset	F1-Score Value/Rank									
	NB-Basic	NB-Tomek	NB-Comm	NB-Rec	SMOTE	BLSMOTE	ENN	kmUnder	OBU	Baseline
Wisconsin	<b>94.95</b> 1	<b>94.95</b> 1	<b>94.95</b> 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	<b>61.31</b> 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	<b>73.33</b> 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	<b>55.38</b> 1	40.22 7	39.34 8
Vehicle0	69.64 9	77.89 5	71.56 8	74 7	81.16 4	<b>84.21</b> 1	<b>84.21</b> 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	<b>83.33</b> 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	<b>100</b> 1	66.67 10	72.73 4
New-thyroid2	82.35 5	82.35 5	82.35 5	63.64 10	<b>100</b> 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	<b>94.74</b> 1	81.82 9	48.65 10	88.89 5
Segment0	58.56 8	70.65 7	73.86 6	57.78 9	96.88 4	96.97 3	<b>97.64</b> 1	95.45 5	53.72 10	<b>97.64</b> 1
Yeast3	60.19 6	66.67 2	58.72 8	34.78 10	56.6 9	60.38 5	64.29 3	<b>70.89</b> 1	63.64 4	59.26 7
Ecoli3	60.87 5	63.64 2	<b>70</b> 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	<b>80</b> 1	70 2	57.14 7
Vowel0	94.12 3	94.12 3	<b>97.3</b> 1	85.71 9	83.87 10	94.12 3	94.12 3	94.12 3	<b>97.3</b> 1	94.12 3
Glass2	33.33 4	40 3	33.33 4	28.57 6	<b>50</b> 1	<b>50</b> 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	<b>25</b> 1	15.38 5	0 7
Glass4	<b>66.67</b> 1	<b>66.67</b> 1	<b>66.67</b> 1	23.53 9	66.67 1	<b>66.67</b> 1	<b>66.67</b> 1	23.53 9	66.67 1	<b>66.67</b> 1
Ecoli4	88.89 7	88.89 7	<b>100</b> 1	80 10	<b>100</b> 1	85.71 9	<b>100</b> 1	<b>100</b> 1	<b>100</b> 1	<b>100</b> 1
Page-blocks13vs2	71.43 6	71.43 6	71.43 6	35.71 10	<b>100</b> 1	88.89 5	90.91 3	66.67 9	<b>100</b> 1	90.91 3
Abalone09-18	12.9 7	13.04 6	<b>22.86</b> 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	<b>33.33</b> 1	<b>33.33</b> 1	0 10	22.22 6	21.28 7	18.18 8
Ecoli0137vs26	50 8	50 8	<b>100</b> 1	66.67 7	<b>100</b> 1	<b>100</b> 1	<b>100</b> 1	33.33 10	<b>100</b> 1	<b>100</b> 1
Yeast6	52.17 3	54.55 2	42.11 7	52.17 3	<b>66.67</b> 1	20 10	36.36 8	47.06 6	48 5	25 9
<b>Average</b>	4.87	4.17	<b>3.65</b>	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	<b>2.71E-03</b>	<b>3.82E-04</b>	<b>1.04E-02</b>	4.35E-01	8.81E-01	<b>3.66E-03</b>	
NB-Tomek	<b>1.04E-02</b>	<b>1.27E-03</b>	<b>1.90E-02</b>	6.07E-01	8.98E-01	<b>1.01E-02</b>	
NB-Comm	<b>5.16E-03</b>	<b>6.99E-04</b>	<b>1.08E-02</b>	5.11E-01	8.80E-01	<b>5.14E-03</b>	
NB-Rec	<b>3.99E-04</b>	<b>4.04E-05</b>	<b>1.20E-03</b>	1.13E-01	4.16E-01	<b>7.98E-04</b>	
	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	<b>4.88E-02</b>	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	<b>4.31E-02</b>	5.07E-02	1.53E-01	4.70E-01	9.88E-02	<b>3.27E-02</b>	
NB-Tomek	<b>4.76E-02</b>	<b>4.60E-02</b>	1.36E-01	5.36E-01	9.07E-02	<b>3.10E-02</b>	
NB-Comm	7.39E-02	9.56E-02	2.90E-01	4.15E-01	6.47E-02	5.73E-02	
NB-Rec	<b>1.00E-02</b>	<b>1.92E-02</b>	6.56E-02	7.57E-01	2.01E-01	<b>1.17E-02</b>	
	F1-score						
	SMOTE	BLSMOTE	ENN	kmUnder	OBU	Baseline	
NB-Basic	3.82E-01	3.78E-01	<b>2.70E-02</b>	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	<b>2.62E-03</b>	1.95E-01	6.13E-01	<b>4.45E-02</b>	

1  
2  
3  
4  
5  
6  
7  
8  
9 methods, outperformed state-of-the-art kmUnder in both sensitivity and precision.  
10 Similarly, our methods outperformed OBU in precision with comparable results  
11 in sensitivity and G-mean.  
12

13  
14 495 In conclusion, NB-Comm on average ranked best in F1-score (Table 6) and G-  
15 mean. The method also showed high average ranking in terms of sensitivity, and  
16 relatively good ranking in precision. This high performance across the different  
17 measures reflects a good trade-off between true positive and false positive rates of  
18 NB-Comm. Even though relatively high false positives in comparison to low false  
19 negatives might be obtained due to the class imbalance nature, low false positive  
20 and low false negative rates can be achieved with regards to the total numbers of  
21 the majority class instances and minority class instances, respectively. This also  
22 500 shows better trade-off over other methods such as OBU, where high sensitivity  
23 was obtained but with lower precision and F1-score. NB-Basic and NB-Tomek  
24 showed competitive average ranking in F1-score. They provided comparable  
25 trade-offs between sensitivity, and G-means and F1-score, to OBU and kmUnder.  
26 In particular, NB-Basic and OBU had higher positive class accuracy but lower  
27 negative class accuracy than NB-Tomek and kmUnder. Thus, it can be said  
28 that NB-Basic, OBU, NB-Tomek, and kmUnder performed comparably on these  
29 505 datasets, and to consider which method was more preferable, the error costs of  
30 each class must be specified. As a result of the trade-off for the highest positive  
31 accuracy, NB-Rec did not perform well in F1-score. NB-Rec is thus more desirable  
32 when the classification accuracy of the positive class (false negatives) cannot be  
33 compromised while misclassifying negative instances (false positives) is tolerable.  
34 515 Finally, it was interesting to observe that the two well-established methods  
35 SMOTE and BLSMOTE ranked best in precision but showed very low ranking  
36 in G-means, F1-score and sensitivity; also, ENN showed the least improvement  
37 over the baseline in sensitivity. Thus, these well-established methods are the  
38 least suitable solutions for handling the selected imbalanced problems.  
39

40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53 520 Table 8 - 11 shows the results of Random Forest applied with the same exper-  
54 iment settings. All our NB-based methods ranked top in sensitivity. However,  
55 low precision was observed in some cases. For example, NB-Rec achieved the  
56  
57  
58

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

Dataset	Sensitivity Value/Rank																				
	NB-Basic	NB-Tomek	NB-Comm	NB-Rec	SMOTE	BLSMOTE	ENN	kmUnder	OBU	Baseline											
Wisconsin	100	100	100	100	1	91.49	10	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	
Ecol1	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	
Ecol2	80	2	80	2	80	2	80	2	80	2	80	2	80	2	80	2	90	1	80	2	
Segment0	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	
Ecol3	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	
Ecol4	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	50	3	75	1	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	
Ecol10137vs26	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	3.33	1.63	4.92	4.92	5.17	5.92	2.5	2.42	6.21	2.5	2.42	6.21	2.5	2.42	6.21	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			
Ecol1	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	1	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			
Ecol2	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			
Segment0	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			
Ecol3	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	97.18	1	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	3	93.54	2	98.74	1	69.82	3	69.82	3	68.92	3	67.08	10	69.82	3			
Ecol4	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			
Ecol10137vs26	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	4.83	4.75	4.29	4.83	5.21	5.21	5.25	5.13	5.5	5.13	4.46	4.83	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

Dataset	Precision Value/Rank																			
	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
Ecol1	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
Ecol2	62.13	10	89.13	7	100	1	67.22	9	100	1	100	1	100	1	100	1	75.47	8	100	1
Segment0	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
Ecol3	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
Ecol4	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																			

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	<b>98.95</b> 1	<b>98.95</b> 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	<b>66.67</b> 1	50.26 10	64.08 5				
Class0	65.12 10	68.75 7	70.97 5	70.97 5	<b>78.26</b> 1	75 4	<b>78.26</b> 1	75.86 3	68.29 8	66.67 9				
Vehicle1	46.49 8	47.78 7	50 6	42.57 9	<b>61.54</b> 1	57.45 4	<b>60.76</b> 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	<b>92.86</b> 1	<b>92.86</b> 1				
Ecoli1	83.33 5	83.33 5	83.33 4	81.08 8	80 9	77.42 10	<b>85.71</b> 1	85.71 2	83.33 5	85.71 2				
New-thyroid1	92.31 7	<b>100</b> 1	<b>100</b> 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	<b>100</b> 1	82.35 10	<b>100</b> 1	<b>100</b> 1	<b>100</b> 1	87.5 9	93.33 6	<b>100</b> 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 3	<b>88.89</b> 1	<b>88.89</b> 1				
Segment0	89.66 9	99.22 3	96.97 7	95.59 8	98.46 5	<b>99.24</b> 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	<b>80.6</b> 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	<b>66.67</b> 1	54.55 5	41.38 10	60 3				
Yeast2vs4	81.82 5	85.71 2	<b>90</b> 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	<b>97.14</b> 1	<b>97.14</b> 1	<b>97.14</b> 1	78.26 7	72.34 9	97.14 4				
Class2	<b>37.5</b> 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	<b>66.67</b> 1	<b>66.67</b> 1				
Class4	50 2	50 2	40 8	44.44 7	<b>80</b> 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	<b>100</b> 1	85.71 3	66.67 8	66.67 4	<b>100</b> 1	66.67 4				
Page-blocks13vs2	61.54 7	61.54 7	66.67 6	58.82 9	<b>100</b> 1	<b>100</b> 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	<b>54.55</b> 1	17.02 10	21.05 9	54.55 2				
Glass5	<b>40</b> 1	0 5	0 5	<b>40</b> 1	0 5	0 5	0 5	28.57 4	<b>40</b> 1	0 5				
Yeast4	30.77 5	<b>38.89</b> 1	29.41 8	30 6	30 6	37.5 2	33.33 4	6.62 10	34.04 3	18.18 9				
Ecoli137vs26	<b>66.67</b> 1	<b>66.67</b> 1	66.67 4	66.67 4	66.67 4	66.67 4	50 8	5.26 10	<b>66.67</b> 1	50 8				
Yeast6	34.78 7	37.5 6	46.15 4	30 8	<b>61.54</b> 1	46.15 4	54.55 2	4.7 10	26.32 9	54.55 3				
<b>Average</b>	5.92	4.5	4.92	6.79	<b>3.75</b>	3.92	<b>3.75</b>	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	<b>1.76E-02</b>	<b>3.51E-03</b>	3.87E-01	4.67E-01	<b>2.08E-03</b>
NB-Tomek	4.87E-01	2.02E-01	7.98E-02	<b>3.67E-02</b>	5.91E-02	5.90E-02
NB-Comm	6.40E-01	3.14E-01	1.38E-01	<b>2.58E-02</b>	<b>4.50E-02</b>	9.86E-02
NB-Rec	<b>4.47E-03</b>	<b>7.83E-04</b>	<b>8.43E-05</b>	8.29E-01	7.53E-01	<b>8.02E-05</b>
	G-mean					
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					
NB-Basic	SMOTE	BLSMOTE	ENN	kmUnder	OBU	Baseline
NB-Basic	<b>4.16E-02</b>	6.89E-02	<b>1.04E-03</b>	6.57E-01	4.15E-01	<b>7.47E-04</b>
NB-Tomek	1.25E-01	1.66E-01	<b>3.85E-03</b>	4.70E-01	2.74E-01	<b>2.57E-03</b>
NB-Comm	2.02E-01	3.05E-01	<b>6.49E-03</b>	3.53E-01	1.83E-01	<b>5.53E-03</b>
NB-Rec	<b>7.72E-03</b>	<b>5.00E-03</b>	<b>8.21E-05</b>	6.75E-01	8.45E-01	<b>6.65E-05</b>
	F1-score					
NB-Basic	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	<b>86.29</b>	40.32	<b>86.29</b>	42.74	28.23
	G-mean	78.83	76.1	76.69	<b>79.65</b>	63.48	79.12	65.35	53.12
	precision	9.66	18.02	11.81	1.95	<b>81.97</b>	2.77	73.61	81.4
	F1-score	16.81	27.6	19.76	3.81	54.05	5.36	<b>54.08</b>	41.92
MNIST_3	sensitivity	94.29	92.65	93.47	<b>99.18</b>	82.45	95.92	82.45	82.45
	G-mean	<b>94.82</b>	94.56	94.38	77.19	90.71	59.22	90.71	90.71
	precision	31.56	37.58	31.11	5.35	<b>90.18</b>	3.32	<b>90.18</b>	<b>90.18</b>
	F1-score	47.29	53.47	46.69	10.15	<b>86.14</b>	6.43	<b>86.14</b>	<b>86.14</b>
MNIST_5	sensitivity	97.42	97.42	97.42	<b>99.26</b>	91.14	95.94	90.96	90.96
	G-mean	96.26	<b>96.85</b>	95.52	70.98	95.28	93.85	95.18	95.18
	precision	49.72	56.53	43.28	9.1	<b>91.99</b>	36.78	91.98	91.98
	F1-score	65.84	71.54	59.93	16.67	<b>91.57</b>	53.17	91.47	91.47

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

1  
2  
3  
4  
5  
6  
7  
8  
9 class imbalance nature on a large dataset as well as high class overlap (as can  
10 be observed that none of the methods with relatively high sensitivity could  
11 545 simultaneously yield high precision and F1-score, and vice versa). NB-Rec  
12 and kmUnder produced the highest sensitivity while the other metrics were  
13 comparable.  
14  
15

16  
17 On the handwritten digits with class 3 as the minority class (MNIST\_3), the  
18 550 NB-based methods improved both sensitivity and G-mean. NB-Rec achieved  
19 the highest sensitivity of 99.18% and outperformed kmUnder in all metrics. The  
20 other NB-based methods showed competitive sensitivity with significantly higher  
21 G-means, precision, and F1-scores than kmUnder. ENN and OBU did not show  
22 any improvement over the baseline.  
23  
24

25  
26 Similarly, on the handwritten digits with class 5 as the minority class  
27 555 (MNIST\_5), NB-based methods showed significant improvements in sensitiv-  
28 ity while G-mean results were competitive with that of the baseline. NB-Rec had  
29 the highest sensitivity of 99.26%. NB-Basic, NB-Tomek, and NB-Comm yielded  
30 better results in all metrics than kmUnder. They also performed better than  
31 ENN, which rarely showed improvement over the baseline, in both sensitivity  
32 and G-mean. OBU did not show any improvement over the baseline.  
33  
34 560

35  
36 In summary, the performance of our methods on the large high-dimensional  
37 datasets was consistent with the previous experiments. NB-Rec performed best  
38 in sensitivity on all of the large high-dimensional datasets and had reasonable  
39 true negative rates (as can be observed from G-mean), however highly suffered  
40 565 from high false negatives due to the trade-off nature on the large and highly  
41 imbalanced datasets. NB-Basic, NB-Tomek, and NB-Comm showed significantly  
42 higher improvements over ENN and OBU. They were competitive with kmUnder  
43 on average.  
44  
45  
46  
47  
48  
49

## 50 570 **7. Conclusions**

51  
52 In this paper, we proposed a novel undersampling approach to handle classi-  
53 fication of imbalanced and overlapped datasets by identifying and eliminating  
54  
55  
56  
57  
58

1  
2  
3  
4  
5  
6  
7  
8  
9 potential negative instances in the overlapping region. Four different variants  
10 of the proposed approach have been created and extensive experiments using  
11 simulated and real-world datasets were carried out. The proposed methods  
12 575 simulated and real-world datasets were carried out. The proposed methods  
13 were compared against well-established and state-of-the-art methods. Results  
14 showed that our methods achieved highest sensitivity with competitive G-means  
15 across all imbalance degrees on both simulated and real-world datasets. Our  
16 methods also showed competitive performance across all degrees of class overlap  
17 on the simulated datasets. The four variants of the proposed approach provided  
18 580 different benefits and trade-offs: 1) NB-Rec yielded exceptionally high sensitivity  
19 at all degrees of class imbalance and class overlap but had higher false positives  
20 at higher class imbalance degrees; 2) NB-Basic resulted in competitive sensitivity  
21 with better trade-offs of fewer negative class prediction errors (false positives)  
22 than the state-of-the-art methods; 3) NB-Tomek and NB-Comm showed similar  
23 trade-offs and were comparable to state-of-the-art in all metrics. These methods  
24 provide different options that suit various problem domains.  
25  
26  
27  
28  
29  
30  
31  
32

33 From our experimental results, a more consistent performance was observed  
34 across all simulated datasets whereas some variations were observed in real-world  
35 590 datasets. The difference may be due to the difference in data uniformity. The  
36 simulated datasets are uniformly distributed (i.e. data density is uniform across  
37 the entire data space), but this cannot be guaranteed in real-world scenarios.  
38 Such an issue has not been considered in this work. Thus, a possible future  
39 direction includes integrating a density factor into the neighbourhood search  
40 criteria of our methods. Another potential solution is to create an adaptive  
41 method for setting  $k$  value in the  $k$ -NN rule, where the value will be dependent  
42 on the local minority class density. For example, a higher  $k$  value can be used  
43 when the local minority class density is lower than the local majority class  
44 595 density, otherwise a lower  $k$  may be considered. In this work, we only consider  
45 the undersampling criteria for binary-class problems. Multi-class datasets were  
46 treated as a binary-class problem using one-vs-all scheme. However, the searching  
47 criteria of our methods can be modified and extended to handle imbalanced  
48 datasets with more than one minority class. Finally, another interesting direction  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

1  
2  
3  
4  
5  
6  
7  
8  
9 would be to apply a global algorithm to roughly separate the overlapping and  
10 non-overlapping regions, followed by performing a local search. Such an approach  
11 605 could potentially lead to a significant reduction of processing time, which is  
12 required for large datasets.  
13  
14  
15

## 16 **References**

- 17  
18  
19 [1] Barua, S., Islam, M.M., Yao, X., Murase, K., 2014. Mwmote-majority  
20 weighted minority oversampling technique for imbalanced data set learning.  
21 610 IEEE Transactions on Knowledge and Data Engineering 26, 405–425.  
22  
23  
24 [2] Bekkar, M., Djemaa, H.K., Alitouche, T.A., 2013. Evaluation measures for  
25 models assessment over imbalanced data sets. J Inf Eng Appl 3.  
26  
27  
28 [3] Branco, P., Torgo, L., Ribeiro, R.P., 2016. A survey of predictive modeling  
29 on imbalanced domains. ACM Computing Surveys (CSUR) 49, 31.  
30 615  
31  
32 [4] Bunkhumpornpat, C., Sinapiromsaran, K., 2017. Dbmute: density-based  
33 majority under-sampling technique. Knowledge and Information Systems  
34 50, 827–850.  
35  
36  
37 [5] Bunkhumpornpat, C., Sinapiromsaran, K., Lursinsap, C., 2009. Safe-level-  
38 smote: Safe-level-synthetic minority over-sampling technique for handling  
39 620 the class imbalanced problem, in: Pacific-Asia conference on knowledge  
40 discovery and data mining, Springer. pp. 475–482.  
41  
42  
43 [6] Bunkhumpornpat, C., Sinapiromsaran, K., Lursinsap, C., 2012. Dbsmote:  
44 density-based synthetic minority over-sampling technique. Applied Intelli-  
45 625 gence 36, 664–684.  
46  
47  
48 [7] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. Smote:  
49 synthetic minority over-sampling technique. Journal of artificial intelligence  
50 research 16, 321–357.  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

- 1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65
- [8] Chetchotsak, D., Pattanapairoj, S., Arnonkijpanich, B., 2015. Integrating  
630 new data balancing technique with committee networks for imbalanced data:  
Grsom approach. *Cognitive neurodynamics* 9, 627–638.
- [9] Das, S., Datta, S., Chaudhuri, B.B., 2018. Handling data irregularities in  
classification: Foundations, trends, and future challenges. *Pattern Recogni-  
tion* 81, 674–693.
- 635 [10] Devi, D., Purkayastha, B., et al., 2017. Redundancy-driven modified  
tomek-link based undersampling: A solution to class imbalance. *Pattern  
Recognition Letters* 93, 3–12.
- [11] Douzas, G., Bacao, F., Last, F., 2018. Improving imbalanced learning  
through a heuristic oversampling method based on k-means and smote.  
640 *Information Sciences* 465, 1–20.
- [12] Galar, M., Fernández, A., Barrenechea, E., Bustince, H., Herrera, F.,  
2011. An overview of ensemble methods for binary classifiers in multi-  
class problems: Experimental study on one-vs-one and one-vs-all schemes.  
*Pattern Recognition* 44, 1761–1776.
- 645 [13] García, V., Mollineda, R.A., Sánchez, J.S., 2008. On the k-nn performance  
in a challenging scenario of imbalance and overlapping. *Pattern Analysis  
and Applications* 11, 269–280.
- [14] Gong, J., Kim, H., 2017. Rhsboost: Improving classification performance  
in imbalance data. *Computational Statistics & Data Analysis* 111, 1–13.
- 650 [15] Gu, Q., Zhu, L., Cai, Z., 2009. Evaluation measures of the classification  
performance of imbalanced data sets, in: *International symposium on  
intelligence computation and applications*, Springer. pp. 461–471.
- [16] Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., Bing,  
G., 2017. Learning from class-imbalanced data: Review of methods and  
655 applications. *Expert Systems with Applications* 73, 220–239.

- 1  
2  
3  
4  
5  
6  
7  
8  
9 [17] Han, H., Wang, W.Y., Mao, B.H., 2005. Borderline-smote: a new over-  
10 sampling method in imbalanced data sets learning, in: International Con-  
11 ference on Intelligent Computing, Springer. pp. 878–887.  
12  
13  
14 [18] He, H., Bai, Y., Garcia, E.A., Li, S., 2008. Adasyn: Adaptive synthetic  
15 sampling approach for imbalanced learning, in: Neural Networks, 2008.  
660 IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE  
17 International Joint Conference on, IEEE. pp. 1322–1328.  
18  
19  
20  
21 [19] He, H., Garcia, E.A., 2008. Learning from imbalanced data. IEEE Transactions  
22 on Knowledge & Data Engineering , 1263–1284.  
23  
24  
25 [20] Japkowicz, N., Stephen, S., 2002. The class imbalance problem: A systematic  
26 study. Intelligent data analysis 6, 429–449.  
27  
28  
29 [21] Jeni, L.A., Cohn, J.F., De La Torre, F., 2013. Facing imbalanced data–  
30 recommendations for the use of performance metrics, in: 2013 Humaine  
31 Association Conference on Affective Computing and Intelligent Interaction,  
32 IEEE. pp. 245–251.  
33  
670  
34  
35 [22] Jo, T., Japkowicz, N., 2004. Class imbalances versus small disjuncts. ACM  
36 Sigkdd Explorations Newsletter 6, 40–49.  
37  
38  
39 [23] Koziarski, M., Woźniak, M., 2017. Ccr: A combined cleaning and resam-  
40 pling algorithm for imbalanced data classification. International Journal of  
41 Applied Mathematics and Computer Science 27, 727–736.  
42  
43  
675  
44  
45 [24] Krawczyk, B., Galar, M., Jeleń, L., Herrera, F., 2016. Evolutionary under-  
46 sampling boosting for imbalanced classification of breast cancer malignancy.  
47 Applied Soft Computing 38, 714–726.  
48  
49  
50  
51 [25] Laurikkala, J., 2001. Improving identification of difficult small classes by  
52 balancing class distribution, in: Conference on Artificial Intelligence in  
680 Medicine in Europe, Springer. pp. 63–66.  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

- 1  
2  
3  
4  
5  
6  
7  
8  
9 [26] Lee, H.K., Kim, S.B., 2018. An overlap-sensitive margin classifier for  
10 imbalanced and overlapping data. *Expert Systems with Applications* .  
11  
12 [27] Lin, W.C., Tsai, C.F., Hu, Y.H., Jhang, J.S., 2017. Clustering-based  
13 undersampling in class-imbalanced data. *Information Sciences* 409, 17–26.  
14 685  
15  
16 [28] López, V., Fernández, A., García, S., Palade, V., Herrera, F., 2013. An  
17 insight into classification with imbalanced data: Empirical results and  
18 current trends on using data intrinsic characteristics. *Information Sciences*  
19 250, 113–141.  
20  
21 [29] López, V., del Río, S., Benítez, J.M., Herrera, F., 2015. Cost-sensitive linguistic  
22 fuzzy rule based classification systems under the mapreduce framework  
23 for imbalanced big data. *Fuzzy Sets and Systems* 258, 5–38.  
24 690  
25  
26 [30] Nekooimehr, I., Lai-Yuen, S.K., 2016. Adaptive semi-unsupervised weighted  
27 oversampling (a-suwo) for imbalanced datasets. *Expert Systems with Ap-  
28 plications* 46, 405–416.  
29 695  
30  
31 [31] Ng, W.W., Hu, J., Yeung, D.S., Yin, S., Roli, F., 2015. Diversified sensitivity-  
32 based undersampling for imbalance classification problems. *IEEE transac-  
33 tions on cybernetics* 45, 2402–2412.  
34  
35 [32] Patel, H., Thakur, G.S., 2017. Classification of imbalanced data using  
36 a modified fuzzy-neighbor weighted approach. *International Journal of  
37 Intelligent Engineering and Systems* 10, 56–64.  
38 700  
39  
40 [33] Rifkin, R., Klautau, A., 2004. In defense of one-vs-all classification. *Journal  
41 of machine learning research* 5, 101–141.  
42  
43 [34] Sáez, J.A., Luengo, J., Stefanowski, J., Herrera, F., 2015. Smote-ipf:  
44 Addressing the noisy and borderline examples problem in imbalanced classi-  
45 fication by a re-sampling method with filtering. *Information Sciences* 291,  
46 184–203.  
47  
48  
49  
50  
51 705  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

- 1  
2  
3  
4  
5  
6  
7  
8  
9 [35] Seiffert, C., Khoshgoftaar, T.M., Van Hulse, J., Napolitano, A., 2010.  
10 Rusboost: A hybrid approach to alleviating class imbalance. *IEEE Transactions*  
11 *on Systems, Man, and Cybernetics-Part A: Systems and Humans* 40,  
12 710 185–197.  
13  
14  
15  
16 [36] Stefanowski, J., 2013. Overlapping, rare examples and class decomposition  
17 in learning classifiers from imbalanced data, in: *Emerging paradigms in*  
18 *machine learning*. Springer, pp. 277–306.  
19  
20  
21 [37] Sun, H., Wang, S., 2011. Measuring the component overlapping in the  
22 715 gaussian mixture model. *Data mining and knowledge discovery* 23, 479–502.  
23  
24  
25 [38] Sun, Z., Song, Q., Zhu, X., Sun, H., Xu, B., Zhou, Y., 2015. A novel  
26 ensemble method for classifying imbalanced data. *Pattern Recognition* 48,  
27 1623–1637.  
28  
29  
30 [39] Visa, S., Ralescu, A., 2003. Learning imbalanced and overlapping classes  
31 720 using fuzzy sets, in: *Proceedings of the ICML*.  
32  
33  
34 [40] Vorraboot, P., Rasmeequan, S., Chinnasarn, K., Lursinsap, C., 2015. Improv-  
35 ing classification rate constrained to imbalanced data between overlapped  
36 and non-overlapped regions by hybrid algorithms. *Neurocomputing* 152,  
37 429–443.  
38 725  
39  
40  
41 [41] Vuttipittayamongkol, P., Elyan, E., Petrovski, A., Jayne, C., 2018. Overlap-  
42 based undersampling for improving imbalanced data classification, in: *Inter-*  
43 *national Conference on Intelligent Data Engineering and Automated*  
44 *Learning*, Springer. pp. 689–697.  
45  
46  
47  
48 [42] Wang, S., Yao, X., 2009. Diversity analysis on imbalanced data sets by  
49 730 using ensemble models, in: *Computational Intelligence and Data Mining,*  
50 *2009. CIDM'09. IEEE Symposium on, IEEE*. pp. 324–331.  
51  
52  
53 [43] Wilson, D.L., 1972. Asymptotic properties of nearest neighbor rules using  
54 edited data. *IEEE Transactions on Systems, Man, and Cybernetics* , 408–  
55 421.  
56 735  
57  
58  
59  
60  
61  
62  
63  
64  
65

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

[44] Yu, H., Liu, K., 2017. 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. pp. 51–54.

740 [45] Zhao, Y., Cen, Y., 2013. Data mining applications with R. Academic Press.

[46] Zhu, Y., Wang, Z., Zha, H., Gao, D., 2018. Boundary-eliminated pseudoinverse linear discriminant for imbalanced problems. IEEE transactions on neural networks and learning systems 29, 2581–2594.

Results on Simulated Datasets (Experiment I)

Sensitivity

overlap	imb	Baseline	NB-Basic	NB-Tomek	NB-Comm	NB-Rec	SMOTE	BLSMOTE	ENN	kmUnder	OBU
0	1.5	0.998333	1	1	1	1	0.999167	0.999167	0.998333	1	0.999167
10	1.5	0.941667	1	1	1	1	0.968333	1	0.95	0.970833	0.975833
20	1.5	0.895	1	1	1	1	0.960833	0.996667	0.9275	0.9575	0.954167
30	1.5	0.806667	1	1	1	1	0.934167	0.983333	0.905	0.9425	0.930833
40	1.5	0.7275	1	1	1	1	0.924167	0.968333	0.880833	0.925833	0.91
50	1.5	0.625	1	1	1	1	0.935833	0.965833	0.875833	0.93	0.911667
60	1.5	0.661667	1	1	1	1	0.921667	0.955	0.873333	0.925	0.928333
70	1.5	0.553333	1	1	1	1	0.916667	0.944167	0.829167	0.915833	0.826667
80	1.5	0.524167	1	1	1	1	0.915833	0.94	0.764167	0.9175	0.816667
90	1.5	0.4075	1	1	1	1	0.900833	0.903333	0.733333	0.9	0.779167
100	1.5	0.333333	1	1	1	1	0.888333	0.8875	0.699167	0.871667	0.760833
0	3	0.996667	1	1	1	1	1	0.991667	0.996667	1	1
10	3	0.916667	1	1	1	1	0.971667	1	0.918333	0.97	0.991667
20	3	0.848333	1	1	1	1	0.986667	1	0.861667	0.98	0.98
30	3	0.793333	1	1	1	1	0.973333	0.998333	0.82	0.986667	0.976667
40	3	0.751667	1	1	1	1	0.965	0.995	0.79	0.978333	0.968333
50	3	0.67	1	1	1	1	0.96	0.995	0.753333	0.963333	0.961667
60	3	0.616667	1	1	1	1	0.965	1	0.65	0.965	0.951667
70	3	0.52	1	1	1	1	0.963333	0.998333	0.625	0.975	0.943333
80	3	0.421667	1	1	1	1	0.965	0.983333	0.573333	0.98	0.933333
90	3	0.376667	1	1	1	1	0.965	0.98	0.595	0.963333	0.92
100	3	0.448333	1	1	1	1	0.97	0.97	0.5	0.96	0.901667
0	12	0.986667	0.993333	0.993333	0.993333	1	1	0.993333	0.986667	1	1
10	12	0.94	0.993333	0.993333	0.993333	1	0.973333	1	0.94	0.993333	1
20	12	0.88	0.993333	0.993333	0.993333	1	0.986667	0.993333	0.886667	0.993333	1
30	12	0.753333	0.993333	0.993333	0.993333	1	0.986667	1	0.773333	0.993333	1
40	12	0.666667	0.993333	0.993333	0.993333	1	0.993333	0.993333	0.7	0.993333	1
50	12	0.613333	0.993333	0.993333	0.993333	0.993333	0.993333	0.993333	0.633333	0.993333	1
60	12	0.54	0.993333	0.993333	0.993333	0.993333	0.993333	1	0.593333	0.993333	1
70	12	0.453333	0.993333	0.993333	0.993333	0.993333	0.993333	0.993333	0.613333	0.993333	1
80	12	0.34	0.993333	0.993333	0.993333	0.993333	0.993333	0.993333	0.606667	0.993333	1
90	12	0.506667	0.993333	0.993333	0.993333	0.993333	0.993333	0.993333	0.58	0.993333	1
100	12	0.366667	0.993333	0.993333	0.993333	1	0.993333	0.993333	0.473333	0.993333	1
0	30	0.983333	1	1	1	1	1	0.983333	0.983333	1	1
10	30	0.966667	1	1	1	1	0.983333	1	0.966667	1	1
20	30	0.95	1	1	1	1	0.966667	1	0.95	1	1
30	30	0.883333	1	1	1	1	0.966667	1	0.916667	1	1
40	30	0.666667	1	1	1	1	1	1	0.65	1	1
50	30	0.616667	1	1	1	1	0.983333	0.95	0.683333	1	1
60	30	0.566667	1	0.966667	1	1	1	1	0.566667	1	1
70	30	0.533333	1	0.966667	0.966667	1	1	1	0.55	1	1
80	30	0.416667	1	0.983333	0.983333	1	1	1	0.466667	1	1
90	30	0.466667	0.983333	0.966667	0.966667	1	1	0.966667	0.366667	1	1
100	30	0.383333	1	0.983333	0.983333	1	1	0.983333	0.6	1	1
0	60	1	1	1	1	1	1	1	1	1	1
10	60	0.966667	1	1	1	1	1	1	0.966667	1	1
20	60	0.9	1	1	1	1	1	1	0.9	1	1
30	60	0.866667	1	1	1	1	1	1	0.833333	1	1
40	60	0.8	1	1	1	1	1	1	0.8	1	1
50	60	0.8	1	1	1	1	1	1	0.8	1	1
60	60	0.733333	1	1	1	1	1	1	0.766667	1	1
70	60	0.7	1	1	1	1	1	1	0.833333	1	1
80	60	0.633333	1	1	0.966667	1	1	1	0.7	1	1
90	60	0.733333	1	0.966667	1	1	1	0.766667	0.5	1	1
100	60	0	1	0.966667	0.966667	1	1	1	0.466667	1	1
0	120	1	1	1	1	1	1	1	1	1	1
10	120	1	1	1	1	1	1	1	1	1	1
20	120	0.866667	1	1	1	1	1	1	0.866667	1	1
30	120	0.866667	1	1	1	1	1	1	0.866667	1	1
40	120	0.866667	1	1	1	1	1	1	0.933333	1	1
50	120	0.866667	1	1	1	1	1	1	0.866667	1	1
60	120	0.8	1	1	1	1	1	1	0.8	1	1
70	120	0.866667	1	1	1	1	1	1	0.8	1	1
80	120	0.133333	1	1	1	1	1	1	0.866667	1	1
90	120	0	1	1	1	1	1	1	0.266667	1	1
100	120	0	1	1	1	1	1	1	0	1	1

G-mean

overlap	imb	Baseline	NB-Basic	NB-Tomek	NB-Comm	NB-Rec	SMOTE	BLSMOTE	ENN	kmUnder	OBU
0	1.5	0.998889	0.972968	0.974394	0.974109	0.947804	0.998194	0.995131	0.998611	0.9975	0.912186
10	1.5	0.950789	0.929456	0.93482	0.936305	0.912262	0.956314	0.951023	0.953328	0.948513	0.866203
20	1.5	0.90467	0.886629	0.893806	0.895669	0.871461	0.912829	0.912866	0.909124	0.890346	0.829494
30	1.5	0.851267	0.844919	0.854725	0.856349	0.820569	0.868079	0.878277	0.867849	0.836134	0.782154
40	1.5	0.802143	0.800694	0.804846	0.809321	0.778175	0.830689	0.834985	0.823851	0.788489	0.759499
50	1.5	0.748146	0.759751	0.767029	0.76956	0.737865	0.789537	0.791656	0.784548	0.73169	0.732369
60	1.5	0.730578	0.714532	0.727629	0.729155	0.690813	0.751516	0.757668	0.747618	0.685974	0.682432
70	1.5	0.678599	0.665415	0.678233	0.681094	0.640312	0.715406	0.718797	0.717959	0.641952	0.639328
80	1.5	0.644404	0.617792	0.630256	0.635085	0.588784	0.669134	0.671714	0.684298	0.586331	0.630072
90	1.5	0.569773	0.572033	0.584523	0.584998	0.527046	0.627586	0.627657	0.646214	0.5425	0.581067
100	1.5	0.510809	0.496096	0.528625	0.528625	0.470815	0.592361	0.590415	0.599735	0.506155	0.560406
0	3	0.997222	0.983757	0.984039	0.984322	0.973253	0.997775	0.994164	0.9975	0.996249	0.922858
10	3	0.94861	0.967241	0.968389	0.970109	0.955394	0.969163	0.978377	0.950009	0.9675	0.894396
20	3	0.906089	0.944281	0.947511	0.950146	0.935711	0.956769	0.955394	0.911346	0.94451	0.872117
30	3	0.870674	0.921954	0.926463	0.927661	0.909823	0.930734	0.937597	0.878471	0.913427	0.862493
40	3	0.835343	0.9	0.902466	0.907683	0.888194	0.907451	0.916335	0.850722	0.881922	0.84809
50	3	0.785822	0.878762	0.886002	0.886002	0.867307	0.887543	0.896515	0.820097	0.852545	0.840411
60	3	0.749568	0.860878	0.867948	0.868588	0.84951	0.870667	0.878446	0.765325	0.828779	0.822
70	3	0.689573	0.841625	0.847873	0.85049	0.827647	0.853142	0.8637	0.740214	0.805995	0.800916
80	3	0.62096	0.82226	0.828318	0.828989	0.808634	0.835153	0.837849	0.701052	0.780156	0.779886
90	3	0.579716	0.802773	0.810007	0.81035	0.784927	0.817309	0.821983	0.697077	0.755679	0.757613
100	3	0.616108	0.782091	0.787753	0.789163	0.765216	0.801807	0.799788	0.63901	0.733328	0.749358
0	12	0.993035	0.990273	0.991109	0.990273	0.989388	1	0.99583	0.993035	0.992497	0.960035
10	12	0.967649	0.986644	0.986924	0.986924	0.986577	0.981631	0.991351	0.967379	0.985413	0.954521
20	12	0.934166	0.982159	0.983844	0.984685	0.980363	0.982491	0.984124	0.93691	0.979573	0.948683
30	12	0.863112	0.977089	0.977089	0.980472	0.976957	0.979418	0.983475	0.874003	0.968715	0.942514
40	12	0.811263	0.972844	0.974828	0.974828	0.970681	0.977372	0.975394	0.830361	0.9637	0.935414
50	12	0.776602	0.967155	0.969719	0.96915	0.963153	0.971425	0.969435	0.78894	0.956589	0.931248
60	12	0.72808	0.963439	0.964584	0.96487	0.957406	0.966584	0.968676	0.762756	0.947377	0.925263
70	12	0.666722	0.957118	0.95827	0.959709	0.951044	0.961719	0.960858	0.771319	0.941928	0.920447
80	12	0.577235	0.952204	0.95423	0.954519	0.946682	0.95827	0.957982	0.763151	0.937315	0.912262
90	12	0.699036	0.947847	0.949593	0.949011	0.941128	0.952204	0.951044	0.744894	0.92682	0.909212
100	12	0.595352	0.941128	0.943178	0.943471	0.937787	0.946099	0.94464	0.673116	0.915469	0.902466
0	30	0.991632	0.997497	0.997497	0.997497	0.995546	0.999166	0.991632	0.991356	0.995415	0.982344
10	30	0.982646	0.996382	0.996382	0.996661	0.995267	0.989978	0.997775	0.982372	0.993748	0.980079
20	30	0.973596	0.995825	0.995825	0.995825	0.99387	0.980183	0.996382	0.973596	0.990412	0.977809
30	30	0.938813	0.99415	0.994429	0.994708	0.990791	0.978538	0.995267	0.956096	0.988326	0.975819
40	30	0.814453	0.991351	0.992192	0.993591	0.989388	0.994429	0.993031	0.804432	0.985822	0.973253
50	30	0.783316	0.989107	0.989669	0.989669	0.987702	0.9825	0.966796	0.82411	0.984153	0.970681
60	30	0.750469	0.98714	0.971654	0.987983	0.98545	0.989388	0.988826	0.750259	0.979977	0.968102
70	30	0.727655	0.986013	0.969994	0.970548	0.98545	0.987702	0.987421	0.738937	0.976636	0.965517
80	30	0.643162	0.984886	0.977203	0.976924	0.983475	0.985732	0.98545	0.680659	0.977471	0.961769
90	30	0.679706	0.975245	0.9675	0.968055	0.982344	0.985168	0.968609	0.603002	0.973711	0.959745
100	30	0.614651	0.982061	0.974964	0.975525	0.979229	0.984322	0.976924	0.766377	0.970367	0.956556
0	60	1	0.99694	0.99694	0.99694	0.994429	0.998332	1	1	0.989577	0.984886
10	60	0.981826	0.995825	0.996104	0.996382	0.993591	0.997497	0.997497	0.982099	0.991246	0.983757
20	60	0.947101	0.994429	0.994708	0.995267	0.991912	0.996104	0.996104	0.947101	0.989577	0.982344
30	60	0.929396	0.994429	0.994429	0.994429	0.990511	0.995546	0.995267	0.911348	0.978724	0.982344
40	60	0.892686	0.991632	0.993591	0.993591	0.99023	0.994708	0.99387	0.892437	0.976218	0.980929
50	60	0.890942	0.991351	0.991632	0.991912	0.989107	0.993311	0.99387	0.890942	0.974547	0.980079
60	60	0.853251	0.990511	0.990511	0.99023	0.988264	0.991912	0.991351	0.872183	0.973711	0.979229
70	60	0.833633	0.989949	0.99023	0.99023	0.98714	0.99023	0.991071	0.907785	0.973711	0.978661
80	60	0.7925	0.988264	0.988826	0.972206	0.986577	0.989949	0.989669	0.8327	0.972875	0.977809
90	60	0.851578	0.987702	0.971377	0.987983	0.985732	0.988826	0.866057	0.703562	0.972039	0.976388
100	60	0	0.986858	0.971101	0.970824	0.984604	0.987702	0.987702	0.679515	0.971621	0.974109
0	120	1	0.997497	0.997497	0.997497	0.996104	0.999166	0.99861	1	0.990829	1
10	120	1	0.997218	0.997218	0.997218	0.995546	0.999166	0.99694	0.999722	0.989994	1
20	120	0.930949	0.996382	0.996104	0.996661	0.994708	0.997218	0.996382	0.930949	0.988743	0.987702
30	120	0.930432	0.996104	0.996104	0.996104	0.994708	0.996661	0.995546	0.930432	0.991246	0.987421
40	120	0.930432	0.994987	0.995546	0.995546	0.994708	0.995546	0.995267	0.964749	0.990829	0.986858
50	120	0.930432	0.994708	0.994708	0.994429	0.994429	0.995267	0.995267	0.930432	0.990829	0.98545
60	120	0.89393	0.994708	0.994429	0.994429	0.994429	0.994429	0.994708	0.89393	0.977889	0.985168
70	120	0.929655	0.99415	0.994429	0.99415	0.993311	0.994987	0.994429	0.893184	0.976218	0.984886
80	120	0.365148	0.99387	0.993591	0.99387	0.992472	0.994429	0.99415	0.928619	0.974547	0.983757
90	120	0	0.993031	0.993591	0.993591	0.991632	0.99415	0.99415	0.515824	0.972457	0.983475
100	120	0	0.991912	0.992472	0.99387	0.991071	0.994429	0.993031	0	0.972039	0.982627

## Precision

overlap	imb	Baseline	NB-Basic	NB-Tomek	NB-Comm	NB-Rec	SMOTE	BLSMOTE	ENN	kmUnder	OBU
0	1.5	0.999166	0.925926	0.929512	0.928793	0.867679	0.995847	0.986831	0.998333	0.995025	0.799333
10	1.5	0.9401	0.83045	0.840925	0.843882	0.798935	0.920761	0.874636	0.935961	0.916601	0.73787
20	1.5	0.874593	0.757098	0.768246	0.771208	0.734844	0.828305	0.802146	0.850267	0.829004	0.695203
30	1.5	0.841008	0.699708	0.712166	0.714286	0.671141	0.763104	0.752551	0.782421	0.760081	0.644175
40	1.5	0.807586	0.650054	0.654308	0.658979	0.628272	0.708626	0.697479	0.719048	0.707193	0.623644
50	1.5	0.799574	0.611933	0.618238	0.620476	0.594059	0.651392	0.647125	0.662673	0.65225	0.596185
60	1.5	0.695271	0.576646	0.586224	0.587372	0.560486	0.613422	0.614807	0.617925	0.613938	0.553953
70	1.5	0.687371	0.544712	0.552486	0.554273	0.530504	0.580475	0.581622	0.593675	0.58056	0.521556
80	1.5	0.627119	0.518807	0.525164	0.527704	0.505051	0.544329	0.546512	0.568154	0.54478	0.514436
90	1.5	0.57193	0.497719	0.503145	0.503356	0.48	0.516237	0.516436	0.531722	0.516994	0.478261
100	1.5	0.505689	0.4693	0.480577	0.480577	0.461361	0.494664	0.493513	0.489784	0.492931	0.463452
0	3	0.993355	0.911854	0.913242	0.914634	0.863309	0.986842	0.990017	0.995008	0.985222	0.692042
10	3	0.943396	0.837989	0.842697	0.849858	0.792602	0.906687	0.886263	0.946735	0.906542	0.630965
20	3	0.897707	0.754717	0.765306	0.774194	0.728155	0.819945	0.792602	0.888316	0.829337	0.59334
30	3	0.856115	0.689655	0.701754	0.705053	0.659341	0.746803	0.735872	0.822742	0.74842	0.57734
40	3	0.777586	0.636943	0.642398	0.654308	0.612245	0.686833	0.679954	0.7584	0.685748	0.556513
50	3	0.740331	0.594059	0.607903	0.607903	0.573614	0.640712	0.633086	0.700775	0.636564	0.546919
60	3	0.698113	0.562852	0.574713	0.575816	0.544959	0.6	0.593472	0.68662	0.599379	0.522415
70	3	0.669528	0.533333	0.542495	0.546448	0.514139	0.56778	0.568311	0.628141	0.56741	0.495622
80	3	0.621622	0.507185	0.515021	0.515907	0.490597	0.537106	0.533937	0.572379	0.536007	0.471778
90	3	0.538095	0.483871	0.492207	0.492611	0.464756	0.511033	0.512642	0.519651	0.509251	0.449146
100	3	0.493578	0.461894	0.467654	0.469116	0.445765	0.489487	0.487029	0.47619	0.486898	0.443443
0	12	0.993289	0.866279	0.881657	0.866279	0.797872	1	0.980263	0.993289	0.892857	0.515464
10	12	0.952703	0.805405	0.809783	0.809783	0.757576	0.890244	0.828729	0.946309	0.827778	0.483871
20	12	0.897959	0.741294	0.764103	0.776042	0.681818	0.791444	0.768041	0.880795	0.768041	0.454545
30	12	0.849624	0.680365	0.680365	0.719807	0.646552	0.747475	0.717703	0.84058	0.677273	0.42735
40	12	0.813008	0.636752	0.656388	0.656388	0.590551	0.683486	0.662222	0.795455	0.642241	0.4
50	12	0.754098	0.586614	0.608163	0.603239	0.55597	0.623431	0.605691	0.753968	0.598394	0.385604
60	12	0.710526	0.558052	0.56654	0.568702	0.517361	0.582031	0.574713	0.717742	0.549815	0.366748
70	12	0.660194	0.515571	0.522807	0.532143	0.480645	0.545788	0.539855	0.630137	0.524648	0.352941
80	12	0.586207	0.486928	0.498328	0.5	0.458462	0.522807	0.520979	0.558282	0.505085	0.331858
90	12	0.542857	0.464174	0.473016	0.470032	0.43314	0.486928	0.480645	0.527273	0.465625	0.324675
100	12	0.478261	0.43314	0.442136	0.443452	0.408719	0.455657	0.448795	0.47973	0.429395	0.309917
0	30	1	0.869565	0.869565	0.869565	0.789474	0.952381	1	0.983333	0.84507	0.487805
10	30	0.966667	0.821918	0.821918	0.833333	0.779221	0.907692	0.882353	0.95082	0.8	0.458015
20	30	0.934426	0.8	0.8	0.8	0.731707	0.84058	0.821918	0.934426	0.722892	0.431655
30	30	0.929825	0.740741	0.75	0.759494	0.645161	0.773333	0.779221	0.916667	0.681818	0.410959
40	30	0.816327	0.659341	0.681818	0.722892	0.612245	0.75	0.705882	0.829787	0.638298	0.387097
50	30	0.804348	0.606061	0.618557	0.618557	0.576923	0.641304	0.662791	0.788462	0.612245	0.365854
60	30	0.755556	0.566038	0.58	0.582524	0.535714	0.612245	0.6	0.73913	0.555556	0.346821
70	30	0.711111	0.545455	0.54717	0.557692	0.535714	0.576923	0.571429	0.717391	0.517241	0.32967
80	30	0.657895	0.526316	0.531532	0.526786	0.504202	0.540541	0.535714	0.682927	0.526316	0.307692
90	30	0.608696	0.5	0.504348	0.513274	0.487805	0.530973	0.522523	0.594595	0.487805	0.29703
100	30	0.469388	0.483871	0.495798	0.504274	0.447761	0.517241	0.526786	0.486486	0.458015	0.28169
0	60	1	0.731707	0.731707	0.731707	0.6	0.833333	1	1	0.545455	0.357143
10	60	0.852941	0.666667	0.681818	0.697674	0.566038	0.769231	0.769231	0.878788	0.588235	0.340909
20	60	0.818182	0.6	0.612245	0.638298	0.508475	0.681818	0.681818	0.818182	0.545455	0.322581
30	60	0.8125	0.6	0.6	0.6	0.46875	0.652174	0.638298	0.806452	0.37037	0.322581
40	60	0.774194	0.5	0.566038	0.566038	0.461538	0.612245	0.576923	0.75	0.344828	0.306122
50	60	0.631579	0.491803	0.5	0.508475	0.434783	0.555556	0.576923	0.631579	0.32967	0.29703
60	60	0.628571	0.46875	0.46875	0.461538	0.416667	0.508475	0.491803	0.621622	0.322581	0.288462
70	60	0.617647	0.454545	0.461538	0.461538	0.394737	0.461538	0.483871	0.555556	0.322581	0.283019
80	60	0.558824	0.416667	0.428571	0.42029	0.384615	0.454545	0.447761	0.552632	0.315789	0.275229
90	60	0.52381	0.405405	0.402778	0.410959	0.37037	0.428571	0.370968	0.454545	0.309278	0.263158
100	60	0	0.38961	0.39726	0.391892	0.352941	0.405405	0.405405	0.424242	0.306122	0.245902
0	120	1	0.625	0.625	0.625	0.517241	0.833333	0.75	1	0.405405	1
10	120	1	0.6	0.6	0.6	0.483871	0.833333	0.576923	0.9375	0.384615	1
20	120	1	0.535714	0.517241	0.555556	0.441176	0.6	0.535714	1	0.357143	0.254237
30	120	0.866667	0.517241	0.517241	0.517241	0.441176	0.555556	0.483871	0.866667	0.416667	0.25
40	120	0.866667	0.454545	0.483871	0.483871	0.441176	0.483871	0.46875	0.736842	0.405405	0.241935
50	120	0.866667	0.441176	0.441176	0.428571	0.428571	0.46875	0.46875	0.866667	0.405405	0.223881
60	120	0.857143	0.441176	0.428571	0.428571	0.428571	0.428571	0.441176	0.857143	0.220588	0.220588
70	120	0.722222	0.416667	0.428571	0.416667	0.384615	0.454545	0.428571	0.705882	0.208333	0.217391
80	120	1	0.405405	0.394737	0.405405	0.357143	0.428571	0.416667	0.590909	0.197368	0.205479
90	120	0	0.375	0.394737	0.394737	0.333333	0.416667	0.416667	0.5	0.185185	0.202703
100	120	0	0.340909	0.357143	0.405405	0.319149	0.428571	0.375	0	0.182927	0.194805

F1-score

overlap	imb	Baseline	NB-Basic	NB-Tomek	NB-Comm	NB-Rec	SMOTE	BLSMOTE	ENN	kmUnder	OBU
0	1.5	0.998749	0.961538	0.963468	0.963082	0.929152	0.997504	0.992961	0.998333	0.997506	0.888148
10	1.5	0.940883	0.907372	0.91359	0.915332	0.888231	0.943948	0.933126	0.942928	0.942938	0.84033
20	1.5	0.884679	0.861759	0.868936	0.870827	0.847158	0.88966	0.888889	0.887206	0.888631	0.804355
30	1.5	0.823479	0.823328	0.831889	0.833333	0.803213	0.840015	0.852601	0.839258	0.841518	0.761418
40	1.5	0.765454	0.787919	0.791035	0.794439	0.771704	0.80217	0.810886	0.79176	0.801877	0.740088
50	1.5	0.70159	0.759253	0.764088	0.765795	0.745342	0.768126	0.774992	0.754487	0.766747	0.720923
60	1.5	0.678053	0.731484	0.739144	0.740056	0.718348	0.736597	0.748042	0.723757	0.738032	0.693865
70	1.5	0.613112	0.70526	0.711744	0.713224	0.693241	0.710824	0.719822	0.691933	0.710637	0.639587
80	1.5	0.571039	0.683177	0.688666	0.690846	0.671141	0.682821	0.691176	0.651741	0.683639	0.63124
90	1.5	0.475912	0.664636	0.669456	0.669643	0.648649	0.656345	0.657169	0.616462	0.656735	0.59271
100	1.5	0.401808	0.638808	0.649175	0.649175	0.631413	0.635469	0.634306	0.576038	0.629741	0.576025
0	3	0.995008	0.953895	0.954654	0.955414	0.926641	0.993377	0.990841	0.995837	0.992556	0.817996
10	3	0.929839	0.911854	0.914634	0.918836	0.884304	0.938053	0.939702	0.932318	0.937198	0.771225
20	3	0.872322	0.860215	0.867052	0.872727	0.842697	0.895613	0.884304	0.874788	0.898396	0.739158
30	3	0.823529	0.816327	0.824742	0.827016	0.794702	0.845152	0.847242	0.821369	0.851186	0.725697
40	3	0.764407	0.77821	0.782269	0.791035	0.759494	0.802495	0.807848	0.773878	0.806319	0.706813
50	3	0.703412	0.745342	0.756144	0.756144	0.72904	0.768512	0.773817	0.726104	0.766578	0.697281
60	3	0.654867	0.720288	0.729927	0.730816	0.705467	0.739936	0.744879	0.667808	0.739464	0.674542
70	3	0.585366	0.695652	0.7034	0.706714	0.679117	0.714462	0.724305	0.626566	0.717351	0.649828
80	3	0.502483	0.673023	0.679887	0.680658	0.658256	0.690107	0.692082	0.572856	0.692988	0.626749
90	3	0.443137	0.652174	0.659703	0.660066	0.634585	0.668205	0.673154	0.554779	0.666282	0.603609
100	3	0.469869	0.631912	0.637281	0.638638	0.61665	0.650643	0.648468	0.487805	0.646102	0.594505
0	12	0.989967	0.925466	0.934169	0.925466	0.887574	1	0.986755	0.989967	0.943396	0.680272
10	12	0.946309	0.889552	0.892216	0.892216	0.862069	0.929936	0.906344	0.943144	0.90303	0.652174
20	12	0.888889	0.849003	0.863768	0.871345	0.810811	0.878338	0.866279	0.883721	0.866279	0.625
30	12	0.798587	0.807588	0.807588	0.834734	0.78534	0.850575	0.835655	0.805556	0.805405	0.598802
40	12	0.732601	0.776042	0.790451	0.790451	0.742574	0.809783	0.794667	0.744681	0.780105	0.571429
50	12	0.676471	0.737624	0.75443	0.75063	0.712919	0.766067	0.752525	0.688406	0.746867	0.556586
60	12	0.613636	0.714628	0.72155	0.723301	0.680365	0.73399	0.729927	0.649635	0.707838	0.536673
70	12	0.537549	0.678815	0.685057	0.693023	0.647826	0.704492	0.699531	0.621622	0.686636	0.521739
80	12	0.43038	0.653509	0.663697	0.665179	0.627368	0.685057	0.683486	0.58147	0.669663	0.498339
90	12	0.524138	0.632696	0.64086	0.638116	0.603239	0.653509	0.647826	0.552381	0.634043	0.490196
100	12	0.415094	0.603239	0.61191	0.613169	0.580271	0.624738	0.618257	0.47651	0.599598	0.473186
0	30	0.991597	0.930233	0.930233	0.930233	0.882353	0.97561	0.991597	0.983333	0.916031	0.655738
10	30	0.966667	0.902256	0.902256	0.909091	0.875912	0.944	0.9375	0.958678	0.888889	0.628272
20	30	0.942149	0.888889	0.888889	0.888889	0.84507	0.899225	0.902256	0.942149	0.839161	0.603015
30	30	0.905983	0.851064	0.857143	0.863309	0.784314	0.859259	0.875912	0.916667	0.810811	0.582524
40	30	0.733945	0.794702	0.810811	0.839161	0.759494	0.857143	0.827586	0.728972	0.779221	0.55814
50	30	0.698113	0.754717	0.764331	0.764331	0.731707	0.776316	0.780822	0.732143	0.759494	0.535714
60	30	0.647619	0.722892	0.725	0.736196	0.697674	0.759494	0.75	0.641509	0.714286	0.515021
70	30	0.609524	0.705882	0.698795	0.707317	0.697674	0.731707	0.727273	0.622642	0.681818	0.495868
80	30	0.510204	0.689655	0.690058	0.686047	0.670391	0.701754	0.697674	0.554455	0.689655	0.470588
90	30	0.528302	0.662921	0.662857	0.67052	0.655738	0.693642	0.678363	0.453608	0.655738	0.458015
100	30	0.422018	0.652174	0.659218	0.666667	0.618557	0.681818	0.686047	0.537313	0.628272	0.43956
0	60	1	0.84507	0.84507	0.84507	0.75	0.909091	1	1	0.705882	0.526316
10	60	0.90625	0.8	0.810811	0.821918	0.722892	0.869565	0.869565	0.920635	0.740741	0.508475
20	60	0.857143	0.75	0.759494	0.779221	0.674157	0.810811	0.810811	0.857143	0.705882	0.487805
30	60	0.83871	0.75	0.75	0.75	0.638298	0.789474	0.779221	0.819672	0.540541	0.487805
40	60	0.786885	0.666667	0.722892	0.722892	0.631579	0.759494	0.731707	0.774194	0.512821	0.46875
50	60	0.705882	0.659341	0.666667	0.674157	0.606061	0.714286	0.731707	0.705882	0.495868	0.458015
60	60	0.676923	0.638298	0.638298	0.631579	0.588235	0.674157	0.659341	0.686567	0.487805	0.447761
70	60	0.65625	0.625	0.631579	0.631579	0.566038	0.631579	0.652174	0.666667	0.487805	0.441176
80	60	0.59375	0.588235	0.6	0.585859	0.555556	0.625	0.618557	0.617647	0.48	0.431655
90	60	0.611111	0.576923	0.568627	0.582524	0.540541	0.6	0.5	0.47619	0.472441	0.416667
100	60	0	0.560748	0.563107	0.557692	0.521739	0.576923	0.576923	0.444444	0.46875	0.394737
0	120	1	0.769231	0.769231	0.769231	0.681818	0.909091	0.857143	1	0.576923	1
10	120	1	0.75	0.75	0.75	0.652174	0.909091	0.731707	0.967742	0.555556	1
20	120	0.928571	0.697674	0.681818	0.714286	0.612245	0.75	0.697674	0.928571	0.526316	0.405405
30	120	0.866667	0.681818	0.681818	0.681818	0.612245	0.714286	0.652174	0.866667	0.588235	0.4
40	120	0.866667	0.625	0.652174	0.652174	0.612245	0.652174	0.638298	0.823529	0.576923	0.38961
50	120	0.866667	0.612245	0.612245	0.6	0.6	0.638298	0.638298	0.866667	0.576923	0.365854
60	120	0.827586	0.612245	0.6	0.6	0.6	0.6	0.612245	0.827586	0.361446	0.361446
70	120	0.787879	0.588235	0.6	0.588235	0.555556	0.625	0.6	0.75	0.344828	0.357143
80	120	0.235294	0.576923	0.566038	0.576923	0.526316	0.6	0.588235	0.702703	0.32967	0.340909
90	120	0	0.545455	0.566038	0.566038	0.5	0.588235	0.588235	0.347826	0.3125	0.337079
100	120	0	0.508475	0.526316	0.576923	0.483871	0.6	0.545455	0	0.309278	0.326087