Learning from class-imbalanced data: overlap-driven resampling for imbalanced data classification.

VUTTIPITTAYAMONGKOL, P.

2020

Learning from Class-Imbalanced Data: Overlap-driven resampling for imbalanced data classification

Pattaramon Vuttipittayamongkol

A report submitted as part of the requirements for the degree
of Doctor of Philosophy
at the School of Computing
Robert Gordon University
Aberdeen, Scotland

October 2020

Supervisor Dr. Eyad Elyan

Abstract

Classification of imbalanced datasets has attracted substantial research interest over the past years. This is because imbalanced datasets are common in several domains such as health, finance and security, but learning algorithms are generally not designed to handle them. Many existing solutions focus mainly on the class distribution problem. However, a number of reports showed that class overlap had a higher negative impact on the learning process than class imbalance.

This thesis thoroughly explores the impact of class overlap on the learning algorithm and demonstrates how elimination of class overlap can effectively improve the classification of imbalanced datasets. Novel undersampling approaches were developed with the main objective of enhancing the presence of minority class instances in the overlapping region. This is achieved by identifying and removing majority class instances potentially residing in such a region. Seven methods under the two different approaches were designed for the task. Extensive experiments were carried out to evaluate the methods on simulated and well-known real-world datasets. Results showed that substantial improvement in the classification accuracy of the minority class was obtained with favourable trade-offs with the majority class accuracy. Moreover, successful application of the methods in predictive diagnostics of diseases with imbalanced records is presented.

These novel overlap-based approaches have several advantages over other common resampling methods. First, the undersampling amount is independent of class imbalance and proportional to the degree of overlap. This could effectively address the problem of class overlap while reducing the effect of class imbalance. Second, information loss is minimised as instance elimination is contained within the problematic region. Third, adaptive parameters enable the methods to be generalised across different problems. It is also worth pointing out that these methods provide different trade-offs, which offer more alternatives to real-world users in selecting the best fit solution to the problem.

Keywords: class imbalance, class overlap, undersampling, classification, machine learning, medical application

Acknowledgements

I wish to express my deepest appreciation to my principal supervisor, Dr. Eyad Elyan, for his persistent support, guidance, patience and motivation since day one of this PhD journey. Without his substantial help, this thesis would not have been written but he would have enjoyed his weekends more. I would like to also thank Dr. Andrei Petrovski for his mentorship and support whenever needed. Special thanks go to Prof. Chrisina Jayne, who considered giving me the PhD funding and introducing me to the best supervisor I could ever ask for, and also to the School of Computing Science and Digital Media, RGU for providing this funding otherwise I would be stuck somewhere else during the COVID-19 outbreak.

I would like to also acknowledge the continuous support and great love of my family – my mum who stopped me from getting diabetes by snacking too much during studying, my dad who greatly motivated me to this success by always staying positive and taking good care of himself to hit the unwelcome cells hard, my middle sister for paving an easier way for me to the PhD by passing me highlighted textbooks way back in school, and my big sister and her family for having such adorable and smart twin babies to brighten up my days and get me confused. My appreciation goes to my patient boyfriend, Aaron, for being a supportive PhD buddy who let me win in most fights.

More thanks go to all of my research and non-research buddies – Adamu, my nearest neighbour, for bearing with a lot of my questions and my noisy snacking in the office, Pam for helping me in writing and sharing with me many nice stories, Joice for bringing me so much joys and accompanying me everywhere, from my sofa to somewhere we got lost overseas, Dr. Jitkomut for sharing her knowledge on data science and letting me bug her with plenty of questions, Nan for cooking me nice food and sweets, and so so many other of you whose names could not be listed here otherwise this acknowledgement note would be longer than my work. Last but not least, big thanks to every single RGU staff member who assisted me during these years.

Abbreviations & Variables

μ_{th}	elimination threshold
AdaOBU	Adaptive-threshold OBU (method)
BA	balanced accuracy
BLSMOTE	Borderline-SMOTE (method)
BoostOBU	Boosted OBU (method)
EA	evolutionary algorithm
f	number of features
FCM	fuzzy c-means clustering algorithm
FN	false negative
FP	false positive
FPR	false positive rate
GAN	Generative Adversarial Net
imb	class imbalance degree
kmUnder	k-means undersampling (method)
kNN	k-nearest neighbour rule
n	number of majority class instances
N	number of instances
NB-based	Neighbourhood-based (method)
NB-Basic	Basic Neighbourhood Search (method)
NB-Comm	Common Nearest Neighbours Search (method)
NB-Rec	Recursive Search (method)
NB-Tomek	Modified Tomek Link Search (method)
OBU	Overlap-Based Undersampling (method)
p	number of minority class instances
RF	random forest
SMOTE	Synthetic Minority Over-sampling Technique
T	training set
TN	true negative
TP	true positive
TPR	true positive rate

Publications

Journals

- Vuttipittayamongkol, Pattaramon, and Eyad Elyan. "Neighbourhood-based undersampling approach for handling imbalanced and overlapped data." Information Sciences, 509, 2020: 47-70. https://doi.org/10.1016/j.ins.2019.08.062

- Vuttipittayamongkol, Pattaramon, and Eyad Elyan. "Improved Overlap-based Undersampling for Imbalanced Dataset Classification with Application to Epilepsy and Parkinson's Disease." International Journal of Neural Systems, 30(8), 2020. (To appear in August 2020)

Conferences

- Vuttipittayamongkol, Pattaramon, et al. "Overlap-Based Undersampling for Improving Imbalanced Data Classification." International Conference on Intelligent Data Engineering and Automated Learning. Springer, Cham, 2018. https://doi.org/10.1007/978-3-030-03493-1_72

- Vuttipittayamongkol, Pattaramon, and Eyad Elyan. "Overlap-based Undersampling Method for Classification of Imbalanced Medical Datasets." 16th International Conference on Artificial Intelligence Applications and Innovations. Springer, 2020. https://doi.org/10.1007/978-3-030-49186-4_30

Code

- The author's github is available at https://github.com/fonkafon.

Declaration

I confirm that the work contained in this PhD project report has been composed solely by myself and has not been accepted in any previous application for a degree. All sources of information have been specifically acknowledged and all verbatim extracts are distinguished by quotation marks.

Signed .... Date 1 October 2020

Pattaramon Vuttipittayamongkol

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

Supervised machine learning is employed for classification tasks across a wide range of real-world problems. These include anomaly detection [2], medical diagnosis [3], object recognition [4] and business analysis [5]. In these domains, it is common that the class of interest is under-represented, and misclassifying an instance of such a class often comes at a high cost. Thus, in this work, where binary-class problems are considered, we refer to the minority class and the majority class as the positive class and the negative class, respectively, unless otherwise stated. Standard learning algorithms are not generally designed to handle datasets with skewed class distributions [6]. They build classification models based upon maximising the overall accuracy. Thus, without appropriate adjustments, the minority class tends to be overlooked, and hence misclassified. This is the well-known class imbalance problem.

In addition to class imbalance, real-world datasets also often suffer from class overlap. The problem of class overlap occurs when examples from different classes share similar characteristics. This exists near the class borderlines, which makes it difficult for the learning algorithm to define a clear decision boundary. In this research, we aim to address the problem of class overlap in classification of imbalanced datasets, evaluate its level of impact against that of class imbalance on learning algorithms' performance, and finally examine whether elimination of class overlap can be established as a potential approach for improving imbalanced learning.

1.1 Class Imbalance and Class Overlap

An imbalanced dataset is a dataset with an unequal distribution of classes. This is depicted in Figure 1.1, where majority and minority class instances are represented by circles and triangles, respectively.

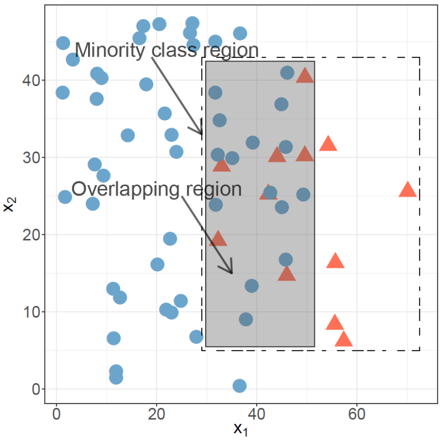


Figure 1.1: An illustration of an imbalanced and overlapped dataset [1]

There are many indicators of the degree of class imbalance. A common scale is the imbalance ratio (imb) shown in Eq. 1.1, where n and p are the numbers of majority and minority class instances, respectively. Class imbalance can also be measured as the percentage of the minority class with respect to the majority class ($\%minority$) as expressed in Eq. 1.2. The percentage of the majority class with respect to the total number of instances ($\%majority$) (Eq. 1.3) can provide an idea of how much the dataset is occupied with majority class instances.

$$imb = \frac{n}{p}, \tag{1.1}$$

$$\%minority = \frac{p}{n} * 100. \tag{1.2}$$

$$\%majority = \frac{n}{n+p} * 100. \tag{1.3}$$

Class overlap occurs when instances of more than one class share a common region of the data space. These instances have similarities in feature values although they belong to different classes, and such a complication is a substantial obstacle in classification tasks. The class overlap problem becomes more serious when class imbalance is also present in the data, and vice versa [7]. In an imbalanced and overlapped dataset, the negative class is normally dominant in the overlapping region. As a result, negative instances are more frequently and clearly visible to the learner than positive instances in such a region. This means that the decision boundary tends to shift towards the negative class leading to misclassification of positive instances near the class boundary, which is undesired in real-world problems.

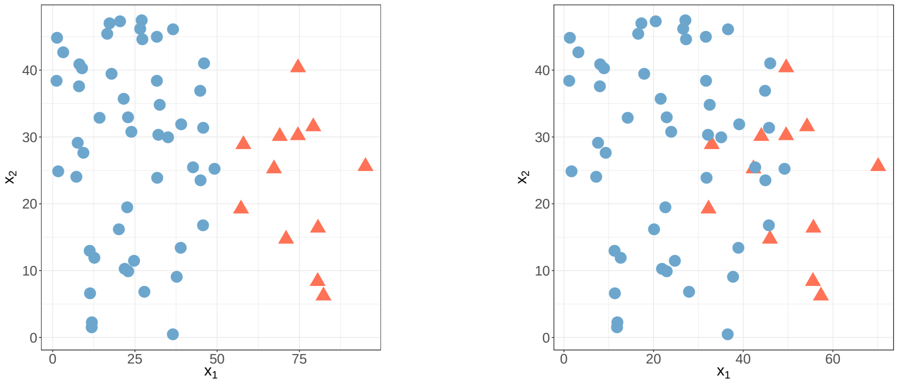
Since class overlap has not been mathematically well characterised, a standard measurement of the overlap degree is not yet defined [8]. Several approaches have been formulated to estimate the overlap degree, however with some constraints [7–10]. For experimental purposes, this research uses the measure of class overlap as expressed in Eq. 1.4, where the area approximations are illustrated in Figure 1.1. This is adopted from the measure proposed by Garcia et al. [7] with some adjustment. The modification is made such that the overlap degree is calculated with respect to the minority class area instead of the total data space. This is to also realise a possible bias in the measure of class overlap caused by the effect of class imbalance.

$$overlap(\%) = \frac{overlapping\ area}{minority\ class\ area} * 100 \tag{1.4}$$

1.2 Motivation

The problem of imbalanced datasets is common in real-world scenarios. For example, in detecting fraudulent transactions, there are considerably more records of legitimate transactions. Similarly, in medical diagnoses, it is not easy to find a large number of patients with a targeted life-threatening disease, e.g. cancer, heart disease. Also, complete information needed to collect from the patients is not always available. As a result, cases with the illness are under-represented in the dataset. When employing a typical learning algorithm to do such tasks, misclassification of minority class instances is easily neglected. This is because the algorithm aims to maximise the overall accuracy and only a small percentage of the minority class accuracy contributes to that. In this matter, high average accuracy does not guarantee that results are preferable. To develop a better understanding of this bias, consider the following example. In diagnoses of one hundred cases with tumors, where one of them is cancerous, the predictive model suggests that all of the tumors are benign. This gives 99% accuracy. However, this is misleading as the model has completely missed the most important case, which is the cancerous tumor. Even though 100% of the benign tumors are correctly identified, misclassification of the cancerous case results in 0% accuracy on the class of interest. This considers a failure in the classification task.

A substantial number of algorithms have been proposed to improve the classification of imbalanced datasets over the past decades [11]. Many of these mainly focused on rebalancing class distribution by means of data resampling. This rebalancing approach proved to be effective [12–14]. However, consider Figure 1.2a, where the imbalanced dataset has no overlapping region, the classification task will be simple. In fact, a linearly separable dataset can be simply classified by a typical classification algorithm no matter how skewed the class distribution is [15]. This implies that no data resampling is needed and attempting to rebalance the class distribution may not be beneficial. On the other hand, when both class imbalance and class overlap are present (Figure 1.2b), the task becomes complicated. This suggests that the impact of class imbalance depends on the presence of class overlap. Likewise, when class imbalance is higher, it is often the case that the imbalance situation in the overlapping region is also higher. There is expected to be fewer minority class instances in the overlapping region, which leads to a higher bias in classification towards the majority class and higher classification errors in the minority class. Thus, it can also be said that the impact of class overlap on the learning algorithm depends on class imbalance to some extent.



(a)

(b)

Figure 1.2: Two imbalanced datasets with the same class distributions (a) without class overlap and (b) with class overlap [1]

Interestingly, several literatures reported that class overlap had a higher negative impact on the performance of learning algorithms than class imbalance [7,13,16,17]. It was shown that imbalanced datasets with no presence of class overlap could be perfectly classified [16]. Moreover, when the class overlap degree was low, class imbalance had no significant effects on the classification results [17]. However, their experimental results were based on limited variations of class imbalance and class overlap degrees. Further investigation needs to be carried out to reinforce this finding. If classification of imbalanced data is less affected by a skewed class distribution than by the overlap problem, minimising class overlap may be a more effective approach to improve classification than rebalancing data.

1.3 Research Objectives

The overall aim of this research is to create and evaluate methods that improve classification of imbalanced datasets by addressing the class overlap problem. The detailed objectives are as follows.

- To investigate and critically review literature on imbalanced dataset classification and solutions.
- To assess and objectively evaluate the impact of class imbalance and class overlap on a learning algorithm's performance. An experimental framework to assess the scale of impact will be created. This includes developing a method to measure the degree of class overlap and designing an experiment to compare the two factors.
- To create novel methods to improve the classification of imbalanced datasets. The methods will aim at minimising the presence of class overlap while at the same time maximising the visibility of the minority class. To achieve this, techniques to identify and remove majority class instances in the overlapping region will be designed and developed.
- To evaluate the developed methods across extensive class imbalance and class overlap degrees using a wide range of data including simulated, real-world and large datasets. The evaluation will also be carried out against well-established and state-of-the-art techniques.

1.4 Contributions

The main contributions of this thesis are outlined as follows.

- A critical discussion of literature including well-established and recent methods. Existing methods were reviewed in the perspective of class overlap-based approaches, class distribution-based approaches and other recent emerging techniques. An overview of commonly-used benchmarking methods in the literature is also provided. Moreover, an extensive experiment illustrating the levels of impact of class overlap and class imbalance was carried out at the full scale of class overlap and an extreme range of class imbalance. Results showed clearly that the performance of the learning algorithm deteriorated across varying degrees of class overlap whereas class imbalance did not always have an effect. This emphasises the need for further research towards handling class overlap in imbalanced datasets.

- A novel overlap-based undersampling method to improve the visibility of minority class instances was created. This involves designing a technique based on soft clustering to identify majority class instances in the overlapping region. An evaluation on public real-world datasets is provided, demonstrating significant improvement in classification, especially on sensitivity, and better performance over a state-of-the-art method. *This work was presented at 19th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL) in Madrid, Spain [18].*

- A new adaptive hybrid method to address the class overlap problem in classification of imbalanced datasets was developed. The method extended the above overlap-based undersampling algorithm with a self-adjusting threshold and an integration with an oversampling technique. The threshold was developed to be adaptive to the degree of class overlap enabling generalisation of this new method across different datasets in real life. An oversampling algorithm was incorporated in the method to improve the detection of negative instances in the overlapping region. A thorough evaluation of this new method using simulated datasets at the full scale of class overlap and extreme degrees of class imbalance, real-world datasets, and large datasets was carried out. Results showed statistically significant improvement over the overlap-based undersampling method suggesting more accurate elimination and less information loss. The method also showed competitive results with other well-established and state-of-the-art methods. *This work was invited to submit to the International Journal of Neural Systems (under review).*

- A new neighbourhood-based undersampling approach for handling imbalanced and overlapped data was created. This contains four methods employing a neighbourhood searching algorithm and different criteria to carefully identify potential overlapped instances. An extensive experiments using simulated and real-world datasets showed comparable performance with state-of-the-art methods with exceptional and statistically significant improvement in sensitivity. *This work was published in the journal of Information Sciences [1].*

- A successful application of the new methods in the medical domain, where identification of rare but significant events is often needed. Results showed high sensitivity on predictive diagnostics of diseases with imbalanced records. Good trade-offs between sensitivity and specificity were also achieved. *The results were submitted to 16th International Conference on Artificial Intelligence Applications and Innovations (under review) and the International Journal of Neural Systems (under review).*

The resulting papers and publications are listed in the preface section Publications.

1.5 Thesis Structure

The remainder of this thesis is organised as follows.

Chapter 2 provides an introduction to the problem in imbalanced dataset classification, which includes the issues of class imbalance and class overlap. An overview of supervised learning algorithms and a discussion on the use of evaluation metrics in imbalanced learning are provided. Existing solutions are critically reviewed.

Chapter 3 introduces a novel overlap-based approach that employs a soft clustering algorithm to identify potential overlapped majority class instances for elimination.

Chapter 4 presents two new undersampling methods that extend and improve the performance of the overlap-based methods presented in Chapter 3. The extensions are developed to improve the identification of potential overlapped instances and introduce an adaptive parameter to enable generalisation of the method.

Chapter 5 details a new overlap-based approach employing a neighbourhood searching algorithm to accurately remove overlapped majority class instances. Four variations with different criteria to identify such instances are presented.

Chapter 6 shows successful application of the overlap-based methods in the medical domain.

Chapter 7 concludes the thesis and discusses findings, summary, limitations and future directions of this work.

Chapter 2

Research Background

This chapter presents an introduction to the problem of imbalanced data classification, which includes the issues of class imbalance and class overlap. Firstly, an overview of learning algorithms is presented. This is followed by a discussion on the usage of common evaluation metrics for imbalanced datasets. Finally, an in-depth review of literature and existing solutions is provided.

2.1 Problem Statement

Classification is the process of classifying samples into the given set of categories based on past observations. As depicted in Figure 2.1, a machine learning algorithm learns from a set of labelled training samples to build a predictive model that maps input variables to the output variable (class). The model is then used to predict the class of new unlabelled instances.

Since traditional learning algorithms are designed to maximise the overall accuracy, the

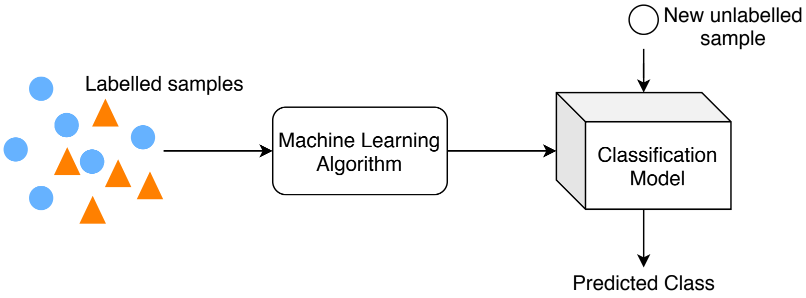


Figure 2.1: An overview of the classification procedure.

classification decision will be biased towards the over-represented class. This could result in high misclassification errors in the minority class, which is not desired in real-world problems. Likewise, some evaluation measures are sensitive to skewed class distributions. Standard metrics used for assessing a classification model are such as overall accuracy, true positive rate (i.e. sensitivity, recall), true negative rate (i.e. specificity), precision, F1-score, and G-mean. Metrics that consider the per-class accuracy or equally weight both classes with respect to their size, e.g, sensitivity and G-mean, are not affected by class distribution. On the other hand, metrics that disregard the difference in class size can be misleading with biases towards the majority class. For example, on a highly imbalanced dataset, the overall accuracy is primarily affected by the accuracy on the majority class whereas a high classification error on the minority class may not be reflected. This is because the result on the minority class makes an insignificant contribution to the average accuracy in such a scenario. Thus, when dealing with imbalanced datasets, evaluation metrics have to be carefully selected and discussed.

The learning task of imbalanced datasets becomes problematic when any two instances of different classes have the same or similar characteristics. This is known as the problem of class overlap, which obstructs the learning algorithm in defining the class boundary. Even though there have been many proposed methods to estimate the degree of class overlap, there are no clear agreements [8]. In [7], the overlap degree of a dataset was determined from the overlapping area with respect to the total data space. However, in an imbalanced dataset, the majority class is generally less overwhelmed by the class overlap. By considering the total data space, which is mostly occupied by the majority class, class overlap will be underestimated. The authors of [1] suggested that this issue should be of concern, especially at a higher class imbalance degree. They followed the class overlap approximation of [7] with some modification. The degree of class imbalance was also taken into account to reduce a possible bias in the measurement. The overlap degree was instead calculated as the overlapping area with respect to the total area of the positive class. However, both approaches are only applicable to synthetic datasets.

Another common approach is using the classification error as the estimated overlap degree, e.g., the percentage of instances misclassified by the k-nearest neighbour rule [19] (kNN) with respect to the number of total instances [9, 20]. However, in [8], the authors showed that such an approach was inaccurate and instead introduced another technique. They proposed a use of the ridge curves of the probabilistic density function to quantify class overlap. The computation was based on the ratio of the saddle point and a smaller peak of the ridge curves of the two classes. This method is one of a few existing methods that measure overlap from the actual contour of data and can be extended to handle multi-class datasets. The main drawback of this approach is that it is only applicable to datasets with normal distributions of both data and features, which is impracticable to real-world datasets.

Prati et al. [21] defined the overlap degree as the distance between the class centroids. Due to arbitrary shapes and non-uniformity of data in nature, this method is likely to be inaccurate. Another method proposed in [10] was based on support vectors. They used Support Vector Data Description (SVDD) [22] to locate approximated boundaries of each class in a binary-class dataset. The overlapping region was then estimated in reference to the amount of common instances found within both boundaries. This method share a similar drawback to the approach of [21]. That is it tends to introduce high errors in overlap approximations since SVDD only discovers a spherical shaped boundary, which is not ideal for real-world datasets.

The following section provides an overview of standard learning algorithms focusing on the key models used in this research. This is followed by a discussion on the usage of common evaluation metrics in the imbalanced context. Misleading results and biases of metrics will be pointed out. Then, a critical review of existing methods for handling the bias in classification tasks of imbalanced datasets. To address the two key issues in classification that are class imbalance and class overlap, we categorise the methods into class distribution-based and class overlap-based solutions. The former group aims at diminishing the problem of class imbalance while the latter mainly considers the problem of class overlap. As discussed above that there is no clear definition of how class overlap is measured, the main challenge of most class overlap-based solutions is thus to identify overlapped or borderline instances. The review of methods will focus more on data resampling to serve the objective of this research.

2.2 Classification Algorithms

A classification algorithm learns from the training data and produces a function that maps new instances to specific classes. Standard classification algorithms are such as decision tree (DT), random forest (RF), support vector machine (SVM), kNN, logistic regression and neural network. These algorithms are generally designed to maximise the overall classification accuracy; thus, their performance will be affected by imbalanced class distributions. An overview of key learning algorithms used in the experiments of this research namely DT, RF, SVM and kNN is provided below.

Decision tree creates a consequence of rules used to classify instances in the form of a tree-like structure. It is composed of nodes, leaves and branches. At each node, a condition for a specific feature to separate the classes is determined. A leaf or an end node shows the final outcome of a decision path. DT decides the best split at a node based on a measure of impurity such as information gain (entropy) or Gini index [23]. In the calculation of these measures, the distribution of classes is not taken into account. This potentially makes the decision given by DT more biased towards the majority class. A key disadvantage of DT is overfitting, which can be addressed by proper tree pruning [24]. DT is one of frequently-used classification algorithms for imbalanced datasets [11] as well as for any classification tasks due to its advantages over other algorithms including the ability to handle missing values and both numerical and categorical variables [24, 25].

Random Forest [26] is an ensemble learning algorithm evolving from DT. It uses Bagging [27] to train a number of DTs and provides the final classification results based on majority voting. By doing so, diverse weak learners (trees) are created, which enables RF to tackle the overfitting issue in DT and achieve better accuracy [28]. Similar to DT, RF is sensitive to class imbalance. However, as the most powerful classification algorithm among standard learning algorithms [29], RF is also widely-used as a base classifier for imbalanced problems [11].

Support vector machine [30] is a binary-class classifier. It builds a separating hyperplane that maximises the margin between the two classes. SVM has an advantage of having many variations of kernel functions. Modeling any datasets is attainable when an appropriate kernel is selected [2]. A kernel function maps non-linear data into higher dimensional space in which they become separable. Among several kernel functions available, radial basis function (RBF) is one of the most-used choices because it is relatively easy to calibrate [31]. A main drawbacks of SVM is being computationally expensive since its complexity grows quadratically with the size of samples [2]. This makes SVM not suitable for large datasets. In the survey of Haixiang et al. [11], SVM was shown as the top selection among the learning algorithms used in imbalanced data classification.

Unlike other classification algorithms, the k-nearest neighbour rule [19] does not construct an internal model. It determines the output class of the testing instance based on its nearest training data points. The class belonging to the majority of the nearest neighbours is the predicted class. A proximity measure is used to search for the k nearest neighbours. The literature showed that different measures gave varied classification results across different datasets [32]. Euclidean distance was proposed in the original work of kNN [19] and is the most commonly-used measure [33]. Other variations e.g., Manhattan, Mahalanobis and Chi square are available. kNN is simple to implement; however, its key disadvantage is a high computational cost for testing as the distances to all instances needs to be computed [33]. This drawback of kNN may prevent its

Table 2.1: Confusion matrix

		Actual Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

application in time-sensitive problems where immediate prediction on testing cases are needed. For imbalanced datasets, kNN was shown to be as frequently-used as RF [11].

2.3 Evaluation Metrics

Some evaluation metrics for classification are not affected by skewed class distributions while others can be misleading with biases towards the majority class. Common metrics for classification of imbalanced datasets such as sensitivity, specificity, balanced accuracy, G-mean, AUC and F1-score will be discussed in detail. For other assessment measures, the reader may refer to [34–37].

A confusion matrix contains the performance information of a classification model. It provides the summary of the predicted results with respect to the actual classes as shown in Table 2.1, where TP, FP, FN and TN denote true positive, false positive, false negative and true negative, respectively. This information is needed in the calculation of other standard evaluation metrics. In imbalanced datasets, accurately detecting the minority class is crucial. This is usually measured in terms of sensitivity, which is also known as recall or the true positive rate (TPR). The metric is formulated as in Eq. 2.1. As sensitivity only reflects the performance over one class, it is often reported in conjunction with another metric, such as specificity (i.e. true negative rate) as expressed in Eq. 2.2, balanced accuracy (BA), G-mean and AUC, to also provide the overall performance or the trade-off between the accuracy of the positive and negative classes [18, 38, 39].

$$sensitivity = \frac{TP}{TP + FN} \tag{2.1}$$

$$specificity = \frac{TN}{TN + FP} \tag{2.2}$$

Balanced accuracy is the arithmetic mean of the accuracy over each class (Eq. 2.3) [40, 41]. It is also referred to as balanced mean accuracy [42], average accuracy [14, 43, 44], macro-accuracy [45], etc. The traditional accuracy (Eq. 2.4 or simply $(TP + TN)/2$) can be misleading when class imbalance is high. In such a case, TN can be highly dominant and give a false idea of high total accuracy regardless of TP. For instance, a perfectly classified negative class of 1000 instances with an entirely misclassified positive class of 10 instances results in over 99% accuracy. This near-perfect accuracy is achieved even though all positive test cases have been completely missed. On the other hand, this same case yields 50% BA, which better reflects the actual performance of the model. Thus, BA often replaces the traditional accuracy, and it is among the most common measures used for imbalanced problems [46].

$$balanced\ accuracy = \frac{sensitivity + specificity}{2} \tag{2.3}$$

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{2.4}$$

Another metric for evaluating the overall performance is G-mean [47]. It is the geometric mean of sensitivity and specificity (Eq. 2.5). Since both G-mean and balanced accuracy give the average values of sensitivity and specificity, they are often used interchangeably. Based on the literature reviewed in this research, G-mean was more frequently used. This could be attributed to the fact that G-mean is also a widely-known metric for problems with non-skewed class distributions whereas balanced accuracy simply reduces to the traditional overall accuracy in a balanced class scenario.

$$G\text{-}mean = \sqrt{specificity * sensitivity} \tag{2.5}$$

$$AM - G\text{-}mean\ inequality : \quad \frac{x + y}{2} \geq \sqrt{xy} \tag{2.6}$$

$$balanced\ accuracy \geq G\text{-}mean \tag{2.7}$$

To gain a deeper understanding of the relationship between BA and G-mean and to determine whether they can be used interchangeably in any scenario, we conduct a further investigation. According to the Arithmetic Mean-Geometric Mean Inequality theory (AM–G-mean inequality) (Eq. 2.6), it can then be said that balanced accuracy is always greater than or equal to G-mean (Eq. 2.7). The equality holds when sensitivity and specificity are equal. For further analysis, consider Figure 2.2, which presents values of G-mean and BA across varying scenarios in terms of the difference between sensitivity and specificity. At each x value, all possible combinations of sensitivity and specificity at a step of 10% are presented. For example, at $x = 0$, the possible combination of (sensitivity,specificity) are (0,0), (10,10), ..., (100,100). It can be seen that the difference between G-mean and BA becomes greater when the difference between sensitivity and specificity increases. This is due to the fact that the geometric mean is more affected by the lower value.

For instance, at $specificity = 90\%$ and $sensitivity = 60\%$, G-mean is 73.48% and BA is 75%. The difference between G-mean and BA is not significant. In an extreme case where $specificity = 100\%$ and $sensitivity = 10\%$, the resulting G-mean is 31.62% while BA is 55%. It is clearly seen that G-mean is affected more by sensitivity. For another extreme case when there is zero accuracy of any class, $G\text{-}mean = 0$. This suggests that G-mean is able to reflect these unfavourable scenarios where balanced accuracy only provides the average value. Thus, to determine a more suitable metric between G-mean and BA, the user will need to carefully make a selection based on the application domain and the main objective of the classification task.

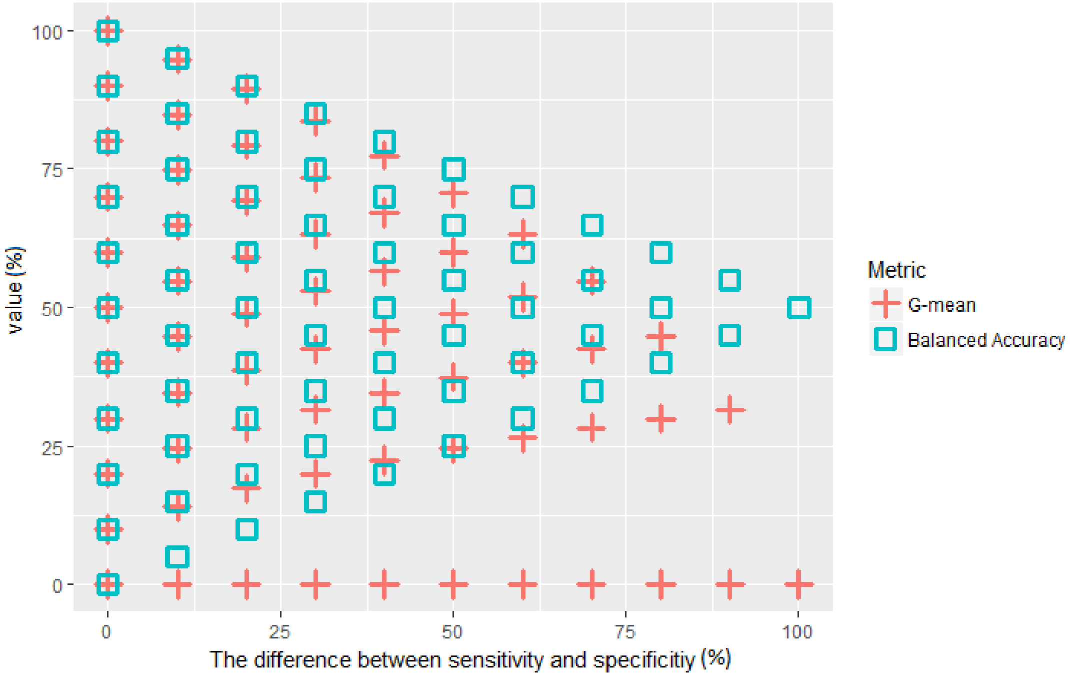


Figure 2.2: The values of G-mean and BA in different scenarios

Another common evaluation metric is F1-score, which is the harmonic mean of sensitivity and precision as expressed in Eq.2.8. It is also widely used for imbalanced problems [9, 45, 48]. However, unlike G-mean and balanced accuracy, F1-score takes into account precision instead of specificity. As shown in Eq. 2.9, precision is calculated using FP and TP. Since FP and TP are not normalised with respect to the class size, FP can be excessively higher than TP in an extremely imbalanced case. This high FP can be deceptive when in fact the false positive rate (FPR) is insignificant. In such case, precision is strongly influenced by FP and does not reflect the actual performance on the positive class. As a consequence, F1-score will also be misleading.

$$F1\text{-}score = \frac{2}{\frac{1}{sensitivity} + \frac{1}{precision}}, \tag{2.8}$$

$$precision = \frac{TP}{TP + FP} \tag{2.9}$$

To demonstrate such an issue, consider an example of a dataset with 10:1000 positive to negative class instances and the classification result of $TP = 10$ and $FP = 10$. This indicates 100% sensitivity and 1% FPR, which is generally highly desirable. Yet, the precision turns out to be 50% leading to 67% F1-score, which very much underestimates and deviates from the actual performance.

It is also worth pointing out that using F1-score alone may not be sufficient to compare between models. In other words, any two models that yield similar FP, TP and sensitivity, will have similar F1-score regardless of their difference in FPRs. Consider an example of two models predicting on datasets with 10:100 and 10:10000 positive to negative class instances. The models achieve 10% and 0.1% of FPR, respectively, and thus both have $FP = 10$. Given the same sensitivity gained of 100%, the models have same value of F1-score accordingly, which is 67%. In fact, the former case with 10% FPR is less favourable than the latter case with 0.1% FPR, but F1-score does not convey that. This is evidence that the use of F1-score alone may not be sufficient to indicate the quality of a classification model on imbalanced datasets. Nonetheless, it can be meaningful when considered along with other measures.

Another commonly-used metric is the area under the receiver operating characteristic curve (AUC). A receiver operating characteristic curve (ROC) visualises the values of TPR against FPR at varying probability thresholds. AUC gives a summary of the ROC curve as a single value. AUC is often used for comparing the performance among classifiers; however, there have been some arguments raised against its usage [49]. Firstly, ROC curves are useful when misclassification costs and class distributions are not specified [37]; so is AUC [50]. This suggests that ROC and AUC can be used for inspecting and summarising the general performance of a classifier. However, in real-life application, the error costs are known and a model can be fine-tuned for the optimal results, which eventually falls onto a single point on the RUC curve. Thus, a classifier with a higher AUC does not necessarily give a better result. This leads to the second argument that visual inspection of ROC curves should be carried out instead of considering only AUC values [49]. However, often there is no clear winning between the two ROC curves making it difficult to compare [50]. Last but foremost, AUC weights the positive and negative class errors equally while in many application domains, misclassification costs are unequal. In this case, summarising over all possible threshold values is unconvincing [51].

In summary, it is recommended that for evaluation of imbalanced dataset classification, individual class accuracy, especially sensitivity, is considered along with an overall performance measure such as balanced accuracy or G-mean. F1-score and AUC can also be assessed; however, they should be carefully discussed due to the constraints discussed above.

2.4 Handling Imbalanced Dataset Classification

Existing literature often discussed solutions to classification of imbalanced datasets as data-level and algorithm-level methods [52–54]. Oversampling and undersampling are common data-level techniques. At the algorithm level, new learning algorithms and modifications of standard learning algorithms are developed. Algorithm-level methods have an advantage of incorporating user's requirements into the model [55]. However, as opposed to the resampling approach, they do not allow flexible choices of learning algorithms. The combinations of data-level and algorithm-level methods, i.e., ensemble-based methods, have also been used. Ensemble-based methods have advantages of both data and algorithm levels, and are less likely to suffer from overfitting than data resampling [56].

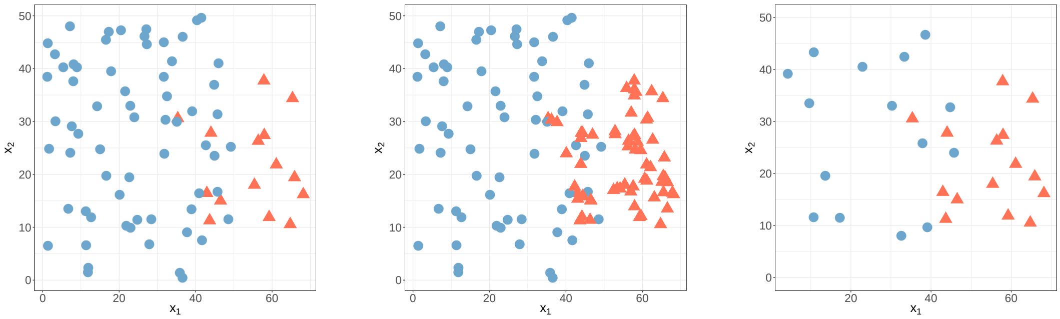
In this thesis, we broadly categorised methods into class distribution-based and class overlap-based focuses. Class distribution-based solutions mainly aim at suppressing the effect of imbalanced class distributions whereas class overlap-based methods deal with instances in the overlapping region to ease the classification task. Additional recent methods using emerging techniques are also discussed here. The overview of the reviewed methods is provided in Table 2.2.

2.4.1 Class distribution-based methods

We categorised methods that were designed to reduce the bias in class distribution as class distribution-based methods. Figure 2.3b and 2.3c illustrates examples of common methods in this category that aim at rebalancing the data by means of oversampling and undersampling, respectively.

Table 2.2: Overview of existing methods

category	technique	resulting classs distribution	methods
class distribution-based	random sampling	varied by settings	Random undersampling; Random oversampling
	linear interpolation	varied by settings	SMOTE [12]
	clustering	balanced	DBSMOTE [39]; k-means SMOTE [57]; k-means undersampling [14]; k-means, Sensitivity-based undersampling [53]
	neighbourhood search	varied by settings	SLSMOTE [58]; BLSMOTE [59]
		balanced weighted	Adaptive kNN [60]
		inversed	k-INOS [61]
	neural networks	balanced	GRSOM [62]; SMOTE-CSELM [63]; UBRKWELM [64]; UBKELM [65]
	ensemble	balanced	RUSBagging [47, 66]; RUSBoost [67]; SMOTEBagging [68]; BalancedEnsemble [69]; GRSOM [62]
		varied by settings	SMOTEBoost [70]
		inversed	Inverse-imbalance Bagging [71]
class overlap-based	clustering	not balanced	OBU [18]; DBMUTE [48]
		(suggested) not balanced	MWMOTE [72]
		balanced	A-SUWO [73]
	neighbourhood search	(suggested) balanced	ADASYN [74]
		not balanced	NCL [75]; PNN [76]; NB-based undersampling [1]
	noise removal	not balanced	SMOTE-IPF [77]; Redency-driven Tomek-link undersampling [43]
	SVM	not balanced	VIRTUAL [78]; OSM [9]
		balanced	DCS [79]
	ensemble	not balanced	HardEnsemble [80]; EVINCI [81]
	region splitting	not balanced	Soft-Hybrid [82]
Others	clustering	balanced	Hierachical decomposition [83]
		not balanced	PSS [84]
	ensemble	not balanced	PT-bagging [45]; Random Balance [85]
	Evolutionary algorithm	balanced	EBUS [86]; EUSBoost [87]; EGIS-CHC [88]
		not balanced	EUSCM [86]; EPRENNID [89]
	neural networks	balanced	LMLE-kNN [44]; cGAN oversampling [90]; MFC-GAN [91]
		balanced error weights	DNN-MFE [92]
		not balanced	Attention Aggregation [42]; CoSen [93]; CSDNN [94]



(a) (b) (c)

Figure 2.3: Class distribution-based resampling applied on (a) the original imbalanced and overlapped dataset using (b) SMOTE and (c) k-means undersampling

Random resampling is the simplest and most common approach. It is the process of either randomly eliminating majority class instances (undersampling) or duplicating minority class instances (oversampling) to achieve the balanced class distribution. Despite its advantage of being simple to implement, random undersampling can potentially lead to important information loss whereas random oversampling is prone to overfitting [6]. Moreover, it was shown in [95] that rebalancing class distribution at random did not guarantee better results.

One of the most well-established methods, Synthetic Minority Over-sampling Technique (SMOTE), was designed to create new instances using linear interpolation between minority class neighbouring points [12]. The authors suggested that the method could expand the decision regions of the minority class and as a results caused less overfitting than random oversampling. Due to its simplicity yet decent performance, SMOTE has been widely applied to real-world problems [96–98]. However, its weaknesses has been presented. In [88], it was shown that by applying SMOTE, the classification results were not improved. This could have been because the method does not consider any selection criteria for linear interpolation; as a result, synthesised instances may not be useful unless they are created near the decision boundary. For more detailed discussion on drawbacks of SMOTE, the reader is referred to [99]. Disadvantages of SMOTE have led to many extensions. These include DBSMOTE [39], Borderline-SMOTE [59] (BLSMOTE), Safe-Level-SMOTE [58] (SLSMOTE) and many others [57, 68, 70].

DBSMOTE [39] is an oversampling method employing DBSCAN [100], a clustering algorithm that can discover arbitrary-shaped clusters, to locate instances in different areas. Another oversampling method, SLSMOTE [58], is based on neighbourhood searching. The common objective of both methods is to synthesise more minority class instances in the non-overlapping region and minimise the synthesis in the overlapping and borderline areas. Although both DBSMOTE and SLSMOTE often achieved improvement over SMOTE, other methods showed superior performance. DBMUTE [48] and BLSMOTE [59], in particular, which also utilise DBSCAN and neighbourhood searching, were shown to outperformed DBSMOTE and SLSMOTE, respectively. It is worth pointing out that these methods take a different approach by emphasising the existence of minority class instances near the borderline regions. Detailed discussion of DBMUTE, BLSMOTE and other class overlap-based extensions of SMOTE is provided in the following subsection.

In [57], the authors proposed a method to account for possible amplification of noise created by SMOTE. They applied k-means clustering to discover clusters dominated by the positive class. This was followed by oversampling in such clusters with the oversampling amount inversely proportional to the number of positive instances. A similar approach was presented in [61]. Both methods however led to significant decreases in the positive class accuracy. This could have been caused by the exclusion of sparse positive instances near the borderline as well as rare cases when performing oversampling.

Koziarski et al. [101] employed radial basis functions in identifying overlapping and non-overlapping regions. This was to avoid synthesising new minority class instances in the overlapping region and maximise the synthesis in the non-overlapping region. However, by doing so, the density of the minority class instances in the overlapping region became relatively sparser. As a consequence, they had a higher tendency to appear as noise to the learning algorithm. Results showed that the method improved specificity but led to lower sensitivity, which is undesired in imbalanced problems. This was consistent with the results obtained using DBSMOTE [39] and SLSMOTE [58] discussed above, and underlines the need of improving the visibility of the minority class instances in the overlapping region.

To address possible information loss in undersampling, clustering is among the common techniques employed during undersampling to ensure the diversity of the remaining instances. In [53] and [14], the authors applied k-means clustering on the majority class and selected representative instances from each cluster. This resulted in a reduced training set with diversified samples. However, since the balanced class distribution was aimed, when applying this method to a highly imbalanced dataset, it nonetheless resulted a significant loss of information.

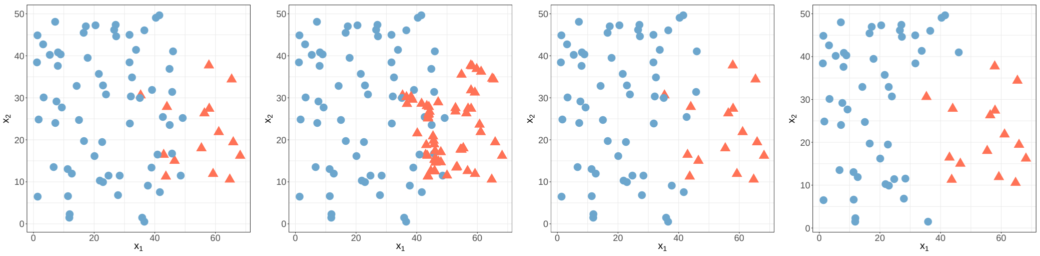
Several rebalancing solutions based on neural networks have been recently proposed [62, 63, 65]. GRSOM, which employs self-organising map and growing ring technique in resampling instances, was introduced in [62]. The deep learning techniques were use in instance generation to preserve the topology of the original data. Unlike other typical data generation methods, GRSOM involves synthesising instances of both majority and minority classes. When majority class undersampling is needed, new majority class instances are created to entirely replace the original minority class instances. Many variants of GRSOM were designed. These included GRSOMO, GRSOMU, which are oversampling and undersampling algorithms, and CnGRSOMO and CnGRSOMU, which are boostrap aggregating variants.

Raghuwanshi and Shukla have recently proposed many variants of methods based on extreme learning machine (ELM) [63–65, 102, 103]. ELM is a single-layer feed-forward neural network that uses a random approach to generate the hidden layer weights. This enables its training speed to be faster than other gradient-based algorithms [64]. The authors exploited this benefit of ELM, and since the traditional ELM was not designed for imbalanced data, they proposed to use many techniques to rebalance the data such as class-specific regularization parameters computed based on the class distribution [63, 102], SMOTE [63] and UnderBagging [64, 65].

Ensemble-based classifiers, which are known to often outperform single classifiers [11], have been extensively adopted to handle imbalanced datasets. In [69], the authors proposed an approach to preserve all available information in building an ensemble-based classifier. This was achieved by subsetting the majority class and combining each subset with the minority class instances to obtain several balanced subsets. Other than preventing information loss, this method has an advantage of having every base classifier trained with no bias in class distribution.

Several widely-known ensemble-based methods are the integrations of ensemble algorithms, such as Bagging (i.e. Bootstrap aggregating) [27] or Boosting [104], and common class distribution-based methods. These are, for example, the combinations of random undersampling and Bagging [47, 66], random undersampling and Boosting (RUSBoost) [67], SMOTE and Boosting [70], and SMOTE and Bagging [68]. These methods were shown to provide promising results. However, with high storage space and computational time required, this type of methods may not be suitable for large datasets.

Unlike typical class distribution-based methods, which attempt to rebalance the class distribution, an inversion of class imbalance was proposed in [71]. This was done by randomly undersampling the negative class until the positive class became over-represented. As a result, higher positive class accuracy was obtained. At the same time, this caused lower negative class accuracy. The authors addressed this issue by combining the approach with Bagging. Results showed that by employing the ensemble-based technique, the trade-off between TPR and FPR was improved.



(a) (b) (c) (d)

Figure 2.4: Class overlap-based resampling applied on (a) the original imbalanced and overlapped dataset using (b) Borderline-SMOTE (c) borderline-based undersampling and (d) overlap-based undersampling

2.4.2 Class overlap-based methods

Class overlap-based methods mainly address the class overlap problem in classification of imbalanced datasets. These methods deal with either overlapped instances near the borderline or instances in the entire overlapping region. Folllowing [82], we define borderline instances as those along the borderline area between the two classes whereas overlapped instances may reside further from the borderline. Therefore, we can say that borderline instances are a subset of overlapped instances. The common objective of overlap-based approaches is to emphasise the presence of the minority class in the overlapping region. This is depicted in Figure 2.4, which shows the resulting datasets after applying class overlap-based resampling methods. Figure 2.4b, 2.4c and 2.4d show the results of oversampling of borderline minority class instances, undersampling of borderline majority class instances, and removing majority class instances from the overlapping region, respectively. As can be seen from these examples, it is worth pointing out that class overlap-based methods may not necessarily produce a balanced class distribution. Due to the risk of potential information loss, most existing overlap-based methods focused specifically on borderline instances, whereas few dealt with the entire overlapping region [1].

DBMUTE [48] is among very few resampling methods that consider the entire overlapping region. The method was designed to eliminate any majority class instances near minority class sub-clusters, which were discovered using DBSCAN. Although both DBMUTE and DBSMOTE [39] employ the same clustering technique to find sub-clusters, they proceed differently. DBMUTE aims at maximising the visibility of minority class instances in the overlapping region by removing majority class instances. On the contrary, DBSMOTE oversamples to rebalance the dataset but avoids creating new minority class instances near the borderline. It was shown that DBMUTE significantly outperformed DBSMOTE [48]. This suggests a higher need to address the overlapping problem for further improvement.

Adaptive Synthetic sampling (ADASYN) was introduced to enhance the presence of the minority class by selectively oversampling in the overlapping region [74]. Instance generation was based on the neighbouring condition. That is, the amount of new instances generated from each minority class instance was proportional to the number of its majority class nearest neighbours. Consequently, more instances were created in the overlapping region while unnecessary syntheses outside such a region were avoided.

HardEnsemble is an ensemble-based method incorporating both oversampling and undersampling to address overlapped instances of both classes [80]. Undersampling was performed based on the contribution to the classification accuracy of instances, which potentially facilitated removal of majority class instances in the overlapping region. Under the same criterion, oversampling was done particularly on minority class instances that were likely to be in the overlapping area. These two resampling processes were carried out in parallel and the resulting datasets were used to train RUSBoost [67]. Although HardEnsemble showed comparable performance with other solutions, it has a benefit over them of no parameter tuning required.

Another method based on ensemble and an evolutionary algorithm (EA), EVINCI, was proposed in [81]. An EA was employed so that negative instances were selectively removed from the overlapping region and minority class instances were more visible. The method was shown to be applicable to multi-class imbalanced problems and outperform other state-of-the-art ensemble-based methods. However, by utilising both EA and ensemble techniques, EVINCI requires high computation complexity. Training time of EVINCI was shown to be extremely higher than other ensemble-based approaches. Thus, the method may not be applicable to large datasets.

In [82], the authors proposed to use different learning algorithms for classifying in different regions of a dataset. Non-overlapping, overlapping, and borderline regions were identified using information based on the data characteristics such as the maximum Fisher's discriminant ratio, probability distributions of the two classes, and the distance between the centers of the two classes. This was followed by using different learning algorithms in the different regions. DBSCAN was selected to learn the borderline region due to its ability in discovering arbitrary-shaped clusters. At the same time, Radial Basis Function Network (RBFN) was used to classified instances in the other regions. This approach showed improvement in classification results. However, it is only applicable to datasets with Gaussian distribution, which is not ideal for handling real-world problems.

To lower the risk of information loss, several methods only focus on overlapped instances that reside near the decision boundary, which we realise as borderline instances. An early and well-known method, Neighbourhood Cleaning Rule (NCL) [75], was adapted from the Edited Nearest Neighbour algorithm (ENN) [105]. NCL is based on removing negative instances that are either misclassified or cause misclassification of positive instances using the *3-NN* classifier. Since NCL only considers three nearest neighbours, it is likely that many negative instances would still remain in the overlapping region, especially in a highly imbalanced and overlapped case.

In [43], aiming at minimising information loss, only negative instances with high similarities and low contribution to classification were removed. However, no thresholds were defined as a stopping criterion for undersampling, and instead negative instances were progressively eliminated according to the similarity and contribution factors until a balanced class distribution was obtained. Applying this method on a highly imbalanced datasets could anyway result in excessive elimination of negative instances.

SMOTE-IPF was proposed in an attempt to remove noisy instances in the original data as well as those generated by SMOTE [77]. This was done by simply applying a noise filter after SMOTE. The authors suggested that this approach had the following advantages over other methods that remove noise prior to oversampling. Firstly, sparse positive instances near the borderline mistaken as noise would no longer appear as noise after being oversampled and hence would not be filtered out. This would preserve highly important information, e.g. rare cases, as well as expand the decision boundary of the positive class. Secondly, having more positive instances in the overlapping region could result in some negative instances being filtered out, hence enhancing the visibility of the positive class in such a region to the learning algorithm.

In addition to class overlap, the problems of small sub-clusters and within-class imbalance were also addressed in [72] and [73]. Majority Weighted Minority Oversampling Technique (MWMOTE) and Adaptive Semi-Unsupervised Weighted Oversampling (A-SUWO) were proposed, respectively. Both methods take an approach of clustering the minority class and subsequently synthesising new instances only within the same sub-cluster. MWMOTE results in more positive instances synthesised in sparser sub-clusters whereas A-SUWO generates more instances in the sub-clusters with higher misclassification errors. Both methods showed improvement in classification results, however, with many parameters needed to be fine-tuned.

SVM, one of the most frequently-used classifiers with imbalanced problems [11], has also been adapted in several methods for handling imbalanced datasets. This includes the use of support vectors to identify and resample potential borderline instances [78, 79] considering that support vectors are mostly composed of such instances [79]. In [78], an SVM-based active learning algorithm was combined with SMOTE to adaptively synthesise instances between positive support vectors in each active learning. Unlike typical data resampling, this oversampling was repeatedly performed during the training process. Similarly, the authors of [79] resampled instances based on support vectors. They made use of Biased SVM [106], which is a learning algorithm implemented specifically to handle imbalanced datasets, to identify support and non-support vectors in the training data. Oversampling and undersampling were then applied to support and non-support vectors, respectively. By doing so, more informative instances were emphasised and information loss was feasibly minimised.

An algorithmic solution based on SVM, an overlap-sensitive margin classifier (OSM), was proposed in [9]. OSM involves instance weighting and selecting different learning algorithms to learn in different regions. Instances were weighted proportionally to the degrees of class imbalance and class overlap. The fuzzy SVM algorithm [107] was employed to locate highly overlapping and low overlapping regions. In the low overlapping region, the classification was carried out using fuzzy SVM. An extreme local search algorithm, *1-NN*, which had shown better results than other classifiers for highly imbalanced and overlapped data [7], was used in the highly overlapping region. Results showed that OSM outperformed other well-known machine learning SVM-based classifiers while consuming lower training time.

A modification of kNN to improve the classification of imbalanced datasets, Positive-biased Nearest Neighbour (PNN), was presented in [76]. The classification decision was adjusted to be biased towards the positive class, particularly in the regions where positive instances were found under-represented. This benefited the positive class especially in the overlapping region. The method showed superior performance over other neighbourhood-based algorithms with significantly lower computational cost.

2.4.3 Emerging methods

Rather than focusing on the class overlap and class imbalance problems, many recent solutions are found to use alternative approaches in handling classification of imbalanced datasets. These include the use of emerging techniques in data resampling such as deep neural network algorithms to obtain the optimal resampled training data. Unlike traditional solutions, some of these methods have the main objective to preserve the topology of the original data, and in some methods, undersampling is not limited to majority class instances but removal of minority class instances is also allowed.

A hierarchical classification method integrating clustering, outlier detection and feature selection was introduced in [83]. Considering that clustering results on outliers and minority class instances were similar, the authors proposed to use an outlier detection method to detect class instances in each level of the hierarchy. The method was shown to be effective in handling highly imbalanced and highly overlapped datasets.

In [84], data clustering technique was employed to allow parallel sampling in large datasets. All discovered clusters of the majority class were simultaneously undersampled to speed up the learning process. Undersampling was carried out in a way that minimum negative class instances were remained for effective model training. That is, only negative instances near the class boundary were kept in the training set. The method proved a substantial reduction in the computational complexity while comparable results to other existing methods were achieved.

As distinct from typical algorithm-based methods, PT-bagging [45] was designed to calibrate the decision probability at the learner's output aiming at reducing the bias in classification decisions towards the majority class. A threshold-moving technique was used to consider the best threshold for each class instead of the commonly-used cut-off probability of 0.5. The technique was combined with Bagging for improved results. Without changing the natural class distribution of data, this approach showed competitive results with other state-of-the-art ensemble-based methods.

In [85], an ensemble was built upon subsets of the training data with random class distributions. To obtain different class distributions, random undersampling and SMOTE were applied. The variety of the training subsets resulted in diversity of weak classifiers, which is beneficial for constructing a good ensemble-based model [108]. Results showed that this simple method performed better than some other state-of-the-art ensembles.

The application of EAs has been extensively seen in recent solutions to imbalanced problems [86–89]. An undersampling framework based on evolutionary prototype selection algorithms was introduced in [86]. The framework aimed at maximising classification results while minimising the training set size. Many variations of methods under this framework were designed. Both balanced and imbalanced training sets were obtained using the proposed variations, and unlike most undersampling methods, removing minority class instances was allowed. Substantial reductions in size of both positive and negative classes were obtained while comparable results with well-established methods were achieved. An ensemble-based extension of this evolutionary-based undersampling approach, EUSBoost, was presented in [87]. EUSBoost is the integration of Boosting and the evolutionary-based undersampling with a modified fitness function to obtain diversified weak classifiers. The extension showed better performance and outperformed many state-of-the-art ensembles.

EPRENNID is an integratation of ensemble, undersampling and oversampling based on EAs [89]. Evolutionary prototype selection and prototype generation were used as undersampling and oversampling methods, respectively. By employing evolutionary prototype selection on both positive and negative instances, several reduced subsets were obtained. Then, only well-performing subsets were selected for subsequent prototype generation. To avoid overfitting, which may be introduced by prototype generation, combinations of several resampled subsets were used for ensemble-based classification. EPRENNID produced relatively robust results on different densities of the minority class compared to some existing solutions while reducing instances of both classes. However, its training time was shown to be far higher than those methods. This was attributed to the use of EAs together with an ensemble technique, which are both computationally expensive.

Another evolutionary-based method was proposed in [88]. The authors applied an EA for selecting the generalised exemplars, i.e. representative instances, that maximised the classification results, particularly in AUC. Classification decisions of new instances were made based on their distances to these generalised exemplars. Experiments showed that the method achieved higher AUC on imbalanced datasets than other exemplar-based learning algorithms. This method can be adapted to consider optimising the results in other measures as required by the user for different problems.

One of the most recent approaches for handling imbalanced datasets is the use of neural network algorithms. Like other learning algorithms, deep neural networks are used to learn imbalanced datasets, and to improve the performance, they are used in conjunction with data resampling and cost-sensitive learning methods [44, 94, 109]. In [42, 92], new loss functions were formulated to reduce the bias in imbalanced class distribution of data. The authors of [92] proposed to use loss functions that considered the mean error of each class; however, results showed trivial improvement over the mean square error (MSE), which is a commonly-used loss function. In [94] and [93], novel loss functions were introduced for the purpose of neural network training and feature extraction. The use of such loss functions was shown to improve the classification performance.

In [60], two novel adaptive kNN algorithms were proposed. Neural networks were utilised in the first algorithm to find the minimum value of k that correctly classified each instance in the training set. In the second algorithm, the value of k was inversely proportional to the local density. This allowed a relatively smaller k value to be used in the overlapping region, which was suggested to be more effective in classifying overlapped instances than a high value of k [7, 9].

Over the past few years, extensions of a state-of-the-art data augmentation algorithm, Generative Adversarial Net (GAN) [110], have been widely used as oversampling methods for imbalanced datasets [90,91]. GAN consists of two models – the generative model, which generates new samples as similar to the original data as possible, and the discriminative model, which attempts to distinguish between the original data and the generated data. The objective of GAN is to simultaneously optimise the two models so that the overall distance between the original and the generated distribution is minimised. This ability of GAN was employed as an oversampling technique in [90] and [91] to synthesise minority class instances. In [90], Conditional GANs (cGAN) [111] was directly applied as an oversampling method. Since GAN was originally designed as an unsupervised learning algorithm, the authors included class labels as an additional learning condition required in cGAN. Results showed that the method outperformed common resampling methods such as BLSMOTE [59] and ADASYN [74]. However, there was inconsistency in the results, which migh have been attributed to insufficient numbers of training data [112,113]. In [91], Multiple Fake Class GAN (MFC-GAN) was proposed specifically as an oversampling technique to rebalance the class distribution. Unlike common GAN extensions, MFC-GAN was designed to create multiple fake classes to improve the classification accuracy of the minority class. This method was evaluated on multi-class image datasets and results showed that it outperformed SMOTE and other GAN extensions [114,115]. Despite promising results achieved using these GAN-based methods, a limitation on the size of training data when applying a deep learning model remains a concern.

2.4.4 Benchmarking methods

An overview of common and well-known methods that were used in the reviewed literature for evaluation and comparison purposes is presented in this subsection. Table 2.3 outlines these benchmarking methods mapped with their compared methods and listed in the order of publishing year. Table 2.4, 2.5 and 2.6 provide further details based on category of the benchmarking methods, namely class distribution-based category, class overlap-based category and emerging techniques, respectively. In these tables, data type indicates the type of datasets used in experiments – real-world (real) or simulated (sim). The range of class imbalance of the datasets used is shown by the minimum and maximum imbalance levels denoted by min imb and max imb, respectively. We defined the levels based on the gaps in imbalance degrees of all datasets used in the literature we reviewed, which are as follows: *balanced = 1-1.5, slightly imbalanced = 1.7-3.4, moderately imbalanced = 8-16.4, highly imbalanced = 21.9-46.6, very highly imbalanced = 51-87, and extremely imbalanced = 115-229.8.* Finally, the right most column of the tables contains the reviewed methods that were shown to be competitive with the benchmarks along with the learning algorithms used.

Table 2.3: Overview of benchmarking methods

benchmark		compared methods
data level	CNN(1968) [116]	[43]; [86]
	Tomek-link(1976) [117]	[43]; [86]; [48]
	NCL(2001) [75]	[86]; [101]
	SMOTE(2002) [12]	[73]; [71]; [61]; [79]; [74]; [58]; [78]; [76]; [45]; [93]; [101]; [48]; [62]; [72]; [69]; [77]; [82]; [9]; [90]; [57]; [1]; [59]; [83]
	BLSMOTE(2005) [59]	[73]; [90]; [57]; [61]; [48]; [89]; [39]; [101]; [77]; [1]
	ADASYN(2008) [74]	[61]; [72]; [101]; [90]
	SLSMOTE(2009) [58]	[73]; [61]; [48]; [39]; [77]
	MWMOTE(2014) [72]	[73]; [61]
	k-means undersampling(2017) [14]	[18]; [1]
algorithm level	1-NN(2008) [7]	[88]; [86]
	PANDA(2014) [118]; FACENET(2015) [119]; Anet(2015) [120]	[44]
	Fast R-CNN(2015) [121]; GoogleNet(2015) [122]; ResNet(2016) [123]	[42]
ensemble	SMOTEBoost(2003) [70]	[67]; [85]; [69]; [87]
	BalanceCascade(2009) [124]	[53]
	SMOTEBagging(2009) [68]	[81]; [85]; [14]; [87]
	EasyEnsemble(2009) [124]	[71]; [53]; [69]; [87]
	UnderBagging(2009) [68]	[14]; [69]; [87]
	RUSBoost(2010) [67]	[85]; [14]; [80]; [69]; [87]; [84]
	Random Balance(2015) [85]	[45]

The information provided in Table 2.3 - 2.6 suggests common and reliable methods that can be considered as good standards for evaluating purposes. However, it is worth pointing out that some of these methods such as SMOTE and BLSMOTE are long-established and have been outperformed by a number of more recent methods. This suggests that there is a need for benchmarking against recent and state-of-the-art methods for more convincing and reliable evaluation.

2.5 Conclusions

In this chapter, related literature, existing solutions and the usage of evaluation metrics in classification of imbalanced datasets were discussed. Solutions were categorised into class distribution-based focus, class overlap-based focus and other emerging techniques. Class distribution-based methods mainly aimed at minimising the problem of class imbalance. The review showed that this was mostly handled by rebalancing the class distribution. On the other hand, many class overlap-based methods dealt with overlapped instances regardless of class distribution. It was evidenced that without having to rebalance the class distribution, class overlap-based methods could provide better results than class distribution-based ones. This finding emphasises that more research effort should be put into development of class overlap-based algorithms.

Moreover, the discussion of evaluation metrics showed how some common metrics can be biased and misleading in an imbalanced scenario. The overview of benchmarking methods presented frequently-used benchmarks for evaluation and comparison purposes.

Table 2.4: Benchmarks for class distribution-based methods

benchmark		data type	min imb	max imb	compared methods
data level	NCL(2001) [75]	real	slightly	very highly	multi(DT,kNN,SVM,NB): Radial-based oversampling [101]
	SMOTE(2002) [12]	real	balanced	highly	DT: Inverse undersampling [71]; multi(DT,kNN,GBM,SVM,RF): k-INOS [61]
			balanced	extremely	Inverse-imbalance Bagging [71]
			slightly	moderately	DT: SLSMOTE [58]
			slightly	highly	multi(BPN, SVM): GRSOM [62]
			slightly	very highly	multi(DT,kNN,SVM,NB): Radial-based oversampling [101]
			moderately	highly	multi(NB,DT,RF): BalancedEnsemble [69]
		real+sim	balanced	extremely	multi(LR, kNN, Gradient tree boosting): k-means SMOTE [57]
			slightly	moderately	DT: BLSMOTE [59]
	BLSMOTE(2005) [59]	real	balanced	extremely	multi(LR, kNN, Gradient tree boosting): k-means SMOTE [57]
			balanced	highly	multi(DT,kNN,GBM,SVM,RF): k-INOS [61]
			slightly	highly	multi(DT,MLP,NB,kNN,SVM,LR,RF): DBMUTE [48]
			slightly	very highly	multi(DT,kNN,SVM,NB): Radial-based oversampling [101]
	ADASYN(2008) [74]	real	balanced	highly	multi(DT,kNN,GBM,SVM,RF): k-INOS [61]
			slightly	very highly	multi(DT,kNN,SVM,NB): Radial-based oversampling [101]
	SLSMOTE(2009) [58]	real	balanced	highly	multi(DT,kNN,GBM,SVM,RF): k-INOS [61]
			slightly	very highly	multi: DBSMOTE [39]
	MWMOTE(2014) [72]	real	balanced	highly	multi(DT,kNN,GBM,SVM,RF): k-INOS [61]
ensemble	SMOTEBoost(2003) [70]	real	slightly	very highly	multi(DT,NB): RUSBoost [67]
			moderately	highly	multi(NB,DT,RF): BalancedEnsemble [69]
	BalanceCascade(2009) [124]	real	slightly	moderately	RBFNN: Sensitivity-based undersampling [53]
	SMOTEBagging(2009) [68]	real	slightly	extremely	multi(DT, MLP): k-means undersampling [14]
	EasyEnsemble(2009) [124]	real	balanced	highly	DT: Inverse undersampling [71]
			balanced	extremely	Inverse-imbalance Bagging [71]
			slightly	moderately	RBFNN: Sensitivity-based undersampling [53]
			moderately	highly	multi(NB,DT,RF): BalancedEnsemble [69]
	UnderBagging(2009) [68]	real	slightly	extremely	multi(DT, MLP): k-means undersampling [14]
			moderately	highly	multi(NB,DT,RF): BalancedEnsemble [69]
	RUSBoost(2010) [67]	real	slightly	extremely	multi(DT, MLP): k-means undersampling [14]
			moderately	highly	multi(NB,DT,RF): BalancedEnsemble [69]

Table 2.5: Benchmarks for class-overlap based methods

benchmark		data type	min imb	max imb	compared methods
data level	CNN (1968) [116]	real	balanced	slightly	multi(BPN,kNN,SVM,NB): Redency-driven Tomek-link undersampling [43]
	Tomek-link(1976) [117]	real	balanced	slightly	multi(BPN,kNN,SVM,NB): Redency-driven Tomek-link undersampling [43]
	SMOTE (2002) [12]	real	balanced	moderately	multi(SVM, kNN, LR, A-SUWO [73]
			balanced	very highly	SVM: DCS [79]
			slightly	moderately	DT: ADASYN [74]
			slightly	highly	SVM-AL: VIRTUAL [78]; multi(DT,kNN): PNN [76]; multi(DT,MLP,NB,kNN,SVM,LR,RF): DBMUTE [48]; multi(kNN, DT): MWMOTE [72]
		real+sim	balanced	moderately	DT: SMOTE-IPF [77]; multi(SVM, RBFN): Soft-Hybrid [82]
			balanced	very highly	SVM:OSM [9]
			balanced	extremely	DT: NB-based undersampling [1]
	BLSMOTE (2005) [59]	real-world	balanced	moderately	multi(SVM, kNN, LR, A-SUWO [73]
			slightly	highly	multi(DT,MLP,NB,kNN,SVM,LR,RF): DBMUTE [48]
		real+sim	balanced	moderately	DT: SMOTE-IPF [77]
			balanced	extremely	DT: NB-based undersampling [1]
	ADASYN (2008) [74]	real	slightly	highly	multi(kNN, DT): MWMOTE [72]
	SLSMOTE (2009) [58]	real	balanced	moderately	multi(SVM, kNN, LR, A-SUWO [73]
			slightly	highly	multi(DT,MLP,NB,kNN,SVM,LR,RF): DBMUTE [48]
			slightly	very highly	multi: DBSMOTE [39]
		real+sim	balanced	moderately	DT: SMOTE-IPF [77]
	MWMOTE (2014) [72]	real	balanced	moderately	multi(SVM, kNN, LR, A-SUWO [73]
	k-means undersampling (2017) [14]	real	slightly	extremely	RF: OBU [18]
		real+sim	balanced	extremely	DT: NB-based undersampling [1]
ensemble	SMOTEBagging (2009) [68]	real	balanced	highly	DT: EVINCI [81]
	RUSBoost (2010) [67]	real	slightly	extremely	RUSBoost: HardEnsemble [80]

Table 2.6: Benchmarks for other emerging methods

benchmark		data type	min imb	max imb	compared methods
data level	CNN(1968) [116]	real	slightly	extremely	kNN:EA undersampling [86]
	Tomek-link(1976) [117]	real	slightly	extremely	kNN:EA undersampling [86]
	NCL(2001) [75]	real	slightly	extremely	kNN:EA undersampling [86]
	SMOTE(2002) [12]	real	slightly	very highly	DNN: CoSen [93]
			slightly	highly	ensembles(DT,kNN): PT-bagging [45]
		real+sim	balanced	extremely	multi(LR,SVM,kNN,DT, Gradient tree boosting): cGAN oversampling [90]
			slightly	highly	proposed(SMOTE+ kNN,SVM,DT): Hierachical decomposition [83]
	BLSMOTE(2005) [59]	real	balanced	extremely	multi(LR,SVM,kNN,DT, Gradient tree boosting): cGAN oversampling [90]
			slightly	highly	kNN: EPRENNID [89]
	ADASYN(2008) [74]	real+sim	balanced	extremely	multi(LR,SVM,kNN,DT, Gradient tree boosting): cGAN oversampling [90]
algorithm level	1-NN(2008) [7]	real	slightly	extremely	EGIS-CHC [88]; kNN:EA undersampling [86]
	PANDA(2014) [118]	real	balanced	highly	LMLE-kNN [44]
	FACENET(2015) [119]	real	balanced	highly	LMLE-kNN [44]
	Anet(2015) [120]	real	balanced	highly	LMLE-kNN [44]
	Fast R-CNN(2015) [121]	real	balanced	highly	Attention Aggregation [42]
	GoogleNet(2015) [122]	real	balanced	highly	Attention Aggregation [42]
	ResNet(2016) [123]	real	balanced	highly	Attention Aggregation [42]
ensemble	SMOTEBoost(2003) [70]	real	slightly	extremely	DT: RB-Boost [85]
			moderately	extremely	DT: EUSBoost [87]
	SMOTEBagging(2009) [68]	real	moderately	extremely	DT: EUSBoost [87]
	EasyEnsemble(2009) [124]	real	moderately	extremely	DT: EUSBoost [87]
	UnderBagging(2009) [68]	real	moderately	extremely	DT: EUSBoost [87]
	RUSBoost(2010) [67]	real	moderately	extremely	DT: EUSBoost [87]
		real+sim	moderately	extremely	SVM: PSS [84]
	Random Balance(2015) [85]	real	slightly	highly	ensembles(DT,kNN): PT-bagging [45]

They can be seen as good standards for future work. At the same time, some of these methods are long-established and have been constantly outperformed. This suggests the need for further comparison against recent and state-of-the-art methods for more convincing and reliable assessments.

Chapter 3

Overlap-Based Undersampling

In this chapter, we first present an objective evaluation on the impact of class imbalance and class overlap. This is followed by an introduction of a novel overlap-based undersampling method. The objective of the method is to eliminate majority class instances from the overlapping region in order to improve the visibility of minority instances. An extensive experiment using 36 public datasets showed statistically significant improvement in sensitivity. *Part of this work was presented at 19th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL) in Madrid, Spain [18].*

3.1 Overview

Traditional learning algorithms are often designed to maximise the overall classification accuracy. As a result, they tend to be biased towards the over-represented class in imbalanced scenarios. Oversampling and undersampling data to obtain better class distributions are commonly used to address this issue. Oversampling has an advantage of no information losses; however, it may significantly increase computational costs on big data. As opposed, undersampling can lead to elimination of important data, but it can be useful in reducing the complexity of training data when instances are carefully removed.

Data resampling methods are widely used due to their simplicity and flexibility. Most existing resampling techniques aim at rebalancing class distribution. However, class imbalance is not the only factor that impacts the performance of the learning algorithm. Literature and results in the previous chapter have shown that class overlap is a key hindrance to classification of imbalanced datasets. Furthermore, class overlap often showed a higher negative impact than class imbalance [7, 13, 16, 17].

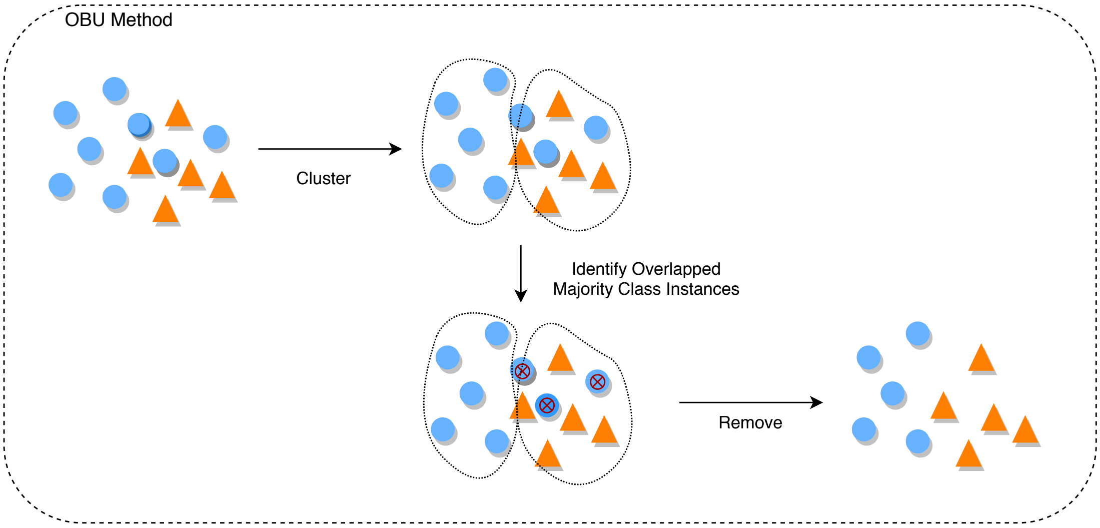


Figure 3.1: The overview of the OBU method.

In this chapter, to develop deeper understanding on the scale of impact of class imbalance and class overlap on the learning algorithm's performance, an experiment was carried out. Unlike in other reports, the full spectrum of class overlap and an extreme range of class imbalance were considered. The results emphasise the need to handle the problem of class overlap in imbalanced data classification. We thus propose a new undersampling method to address the class overlap problem in imbalanced datasets. The method will reduce the dominance of the majority class instances by removing them from the overlapping region. For convenience, we refer to it as Overlap-Based Undersampling method (OBU). As shown in Figure 3.1, OBU incorporates a soft clustering algorithm to determine overlapped instances. The soft clustering algorithm will assign membership degrees to each instance. We hypothesise that an instance with uncertain membership degrees is likely to be in the overlapping region. If such an instance belong to the majority class, it will be removed. By doing so, the visibility of the minority class to the learner will be improved leading to better classification without the need of data rebalancing.

3.2 Impacts of Class Overlap vs Class Imbalance

Previous literature suggested that class overlap had a higher negative effect on the learner's performance than class imbalance [7,16,17]. Their experimental results showed that imbalanced datasets with no presence of class overlap could be perfectly classified. Moreover, when the class overlap degree was low, class imbalance had no significant effect on the classification results. However, it has to be pointed out that these observations were based on limited experiments. In [17], the experiment was carried out with only a few variations of class imbalance and class overlap. Although a wide range of class imbalance degrees was used in [16], the maximum overlap degree experimented was 64% (see [16] for their measurement of the overlap degree). In [7], some datasets were simulated such that the positive class became dominant in the overlapping region. This caused the inconsistency in the overall imbalance degree and the imbalance situation in the overlapping region, which led to inconclusive results. To establish these results at the full scale of class overlap with extreme cases of class imbalance, we have carried out a thorough experiment as follows.

3.2.1 Experiment

Datasets

We synthesised 1,010 uniformly distributed datasets from all possible combinations of 101 class overlap degrees and 10 class imbalance degrees. The overlap degrees (%overlap) as shown in Eq. 1.4 of 0%-100% with a step of 1 were used. The imbalance percentages (%minority), as defined in Eq. 1.2, ranged from 10%-100% with a step of 10. In each dataset, there were 1,000 negative instances and the number of positive instances was based on the imbalance degree.

Setup

Random Forests (RF), one of the mostly-used classifiers for imbalanced datasets [11], was chosen as the learning algorithm. The default parameter settings of RF in *caret*[1] package in *R* were used. That is, the number of trees (*mtree*) was set to 500. The datasets were partitioned into training and testing sets at the ratio of 80 to 20, where the training set was used for model training and the testing set was only used to evaluate the model for the result report. During model training, 10-fold cross-validation was applied for automatic tuning of *mtry*, the number of features determined at each split, in RF. The models were evaluated using sensitivity, specificity, balanced accuracy, and AUC, which are common metrics as discussed in Chapter 2. These will provide the accuracy of each class as well as the overall performance of the models.

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

3.2.2 Results & Discussions

Classification results are shown in Figure 3.2 and 3.3. In Figure 3.2, each point represents the average result of 10 datasets with adjacent overlap degrees for the ease of viewing. Each of the points in Figure 3.3 represents the result on each dataset; however, it should be noted that there were cases that multiple models shared the same results and appear as a single point.

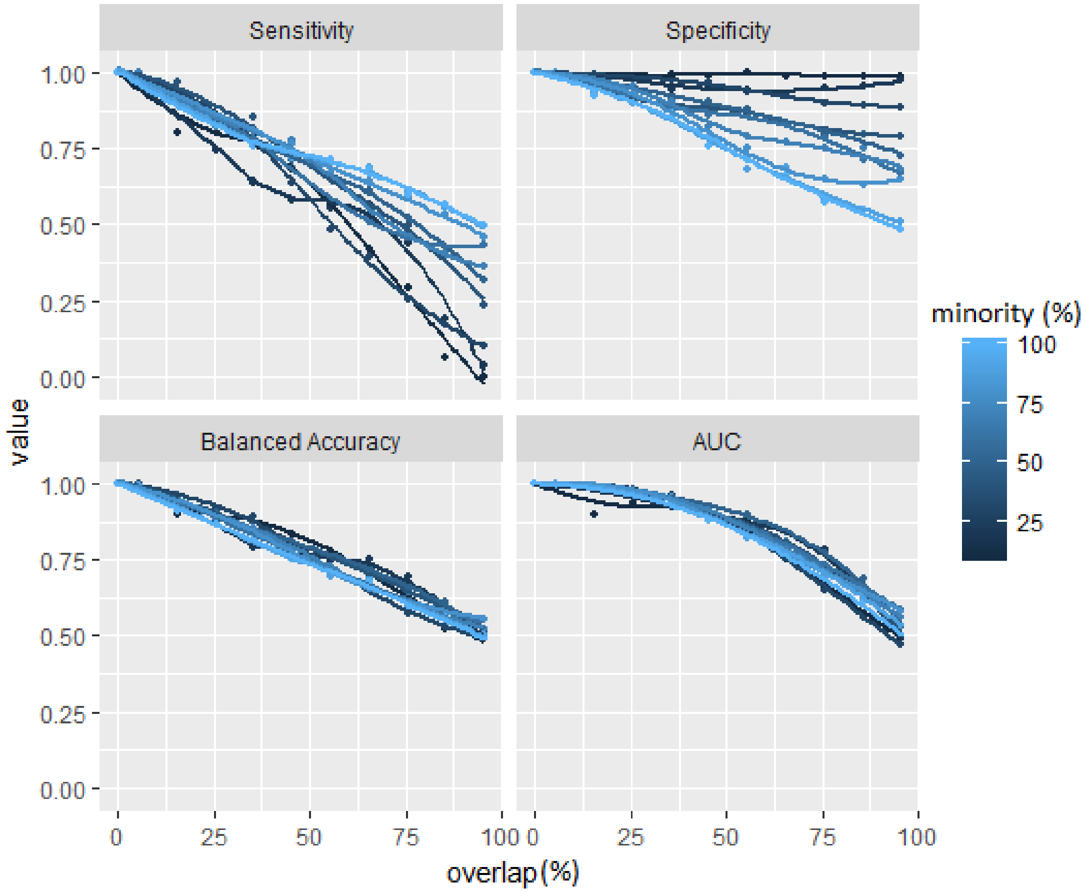


Figure 3.2: Classification results corresponding to various degrees of class imbalance and class overlap with the color scale indicating different imbalance degrees

As can be seen in Figure 3.2 and Figure 3.3, both class imbalance and class overlap cause degradation in sensitivity. However, Figure 3.2 shows that at low overlap degrees, class imbalance has a small effect on the learner's performance whereas in Figure 3.3, class overlap highly degrades sensitivity at any degree of class imbalance. This suggests that class overlap negatively affects sensitivity more than class imbalance.

Figure 3.2 shows that specificity increases as class imbalance increases. This is expected as the dominance of the majority class is increased. On the other hand, class overlap

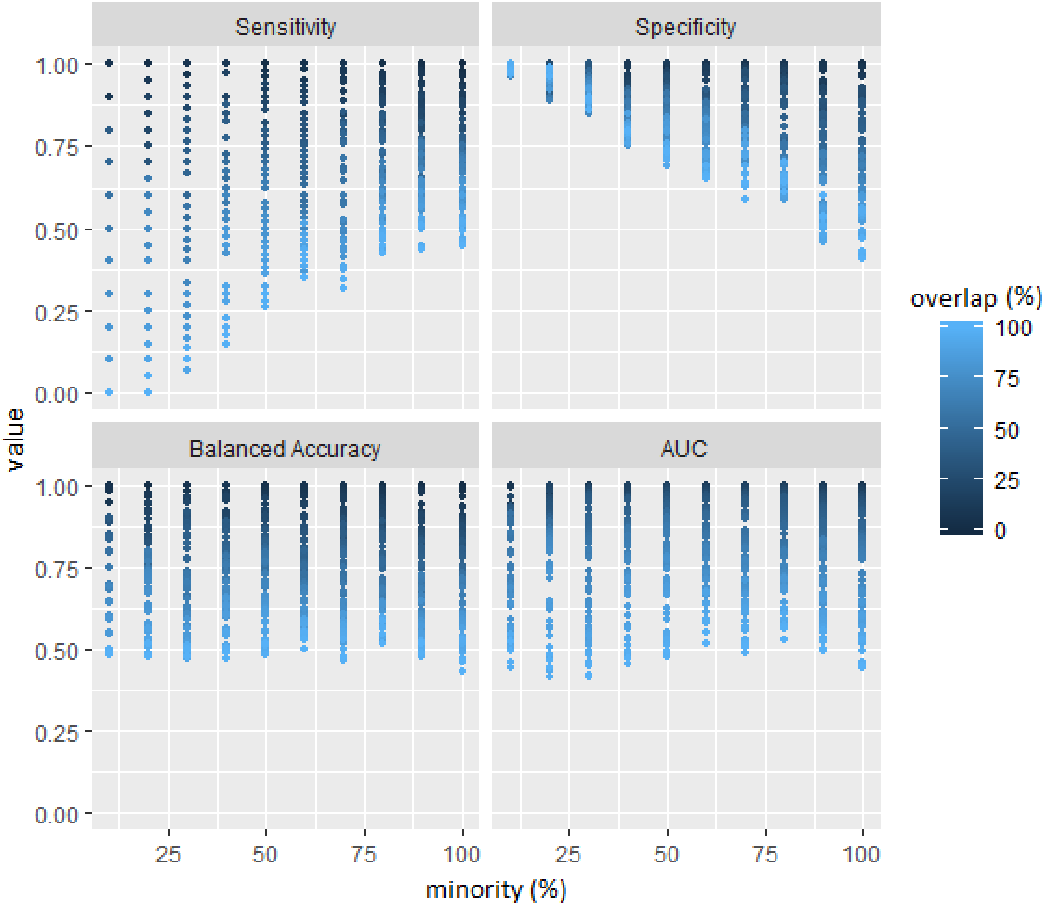


Figure 3.3: Classification results corresponding to various degrees of class imbalance and class overlap with the color scale indicating different overlap degrees

reduces the visibility of instances, hence degrading specificity. It can be observed in Figure 3.3 that class overlap had a higher impact on sensitivity than on specificity. This is attributed to the fact that in our experiment, class overlap was measured with respect to the minority class data space. In an extreme case, the overlapping region occurs in the entire minority class and only occupies part of the majority class.

As a result of the trade-off between the decreases in sensitivity and the increases in specificity, Figure 3.2 shows that class imbalance seems to have no apparent impact on BA and AUC. In contrast, it is clearly seen that BA and AUC decreased as class overlap was higher. The changes of BA and AUC over different degrees of class overlap appear to be linear and non-linear, respectively. These correspond to the relationships between sensitivity and specificity in the calculation of the metrics.

Finally, Figure 3.3 proves that when there is no class overlap, data can be perfectly classified. More importantly, this holds true at any degree of class imbalance.

3.2.3 Summary

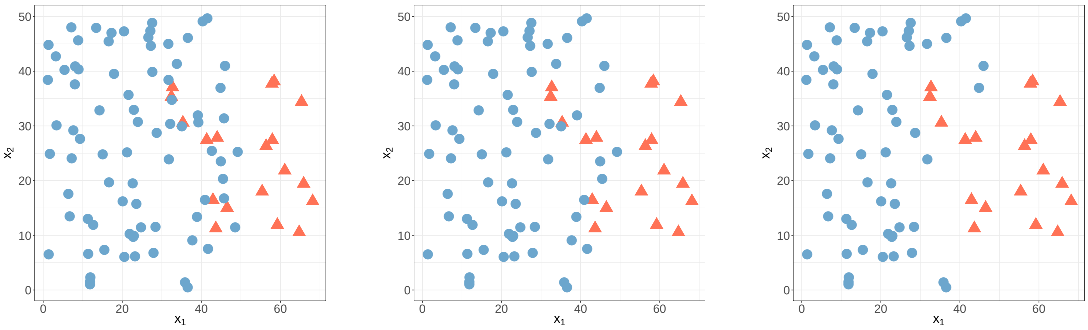
Our experiment clearly shows that class overlap hurt the learner's performance more than class imbalance. While class overlap always degraded the results, class imbalance had an impact only in the presence of class overlap. Moreover, the scale of impact of class imbalance highly depended on the degree of class overlap. That is, class imbalance was more impactful when class overlap was high; on the other hand, it seemed insignificant when class overlap was low.

3.3 The Overlap-Based Undersampling Method

In Chapter 2, several methods under the class overlap-based category were discussed. Some of them focused on borderline instances while some dealt with the entire overlapping region. Based on the experimental results in the previous section, we were motivated to develop a solution that would remove not only majority class instances near the borderline but also those in the overlapping region. To introduce OBU, this section will first briefly discuss a general idea of borderline-based undersampling and overlap-based undersampling to point out their differences. This is followed by a description of a related algorithm used in OBU that is Fuzzy C-means. Finally, the method is presented, evaluated and discussed.

3.3.1 Borderline vs Overlap

Figure 3.4 illustrates examples of a borderline-based undersampling (Figure 3.4b) and an overlap-based undersampling (Figure 3.4c). In Figure 3.4b, some borderline instances have been removed from the original dataset (Figure 3.4a). This was carried out by removing majority class instances that most of their three nearest neighbours are of the minority class. This is likely to lead to better classification results compared to the original dataset. However, high classification errors in the minority class in the complex region may yet occur as the minority class is still under-represented. In Figure 3.4c, we further removed the remaining majority class instances that were overlapped with minority class ones. This was achieved by eliminating any majority class instances that had at least one minority class instance in its three nearest neighbours. This significantly improved the visibility of the minority class, and as a result, potentially maximised its boundary.



(a) (b) (c)

Figure 3.4: Undersampling solutions to (a) an imbalanced and overlapped dataset with (b) borderline instances removed and (c) overlapped instances removed [1]

3.3.2 The Fuzzy C-Means Algorithm

Fuzzy C-means [125] is one of the most commonly-used soft clustering algorithms. Unlike hard clustering, a soft clustering algorithm allows each instance to be a member of many clusters. The likelihood of belonging to a cluster is expressed as a membership degree, whose value is between 0 and 1. The membership degrees of an instance sum up to 1. FCM follows a similar procedure to k-means, a well-known hard clustering algorithm. It begins with randomly initialising cluster centroids. Then, the within-cluster variance is calculated from the fractional distances of all instances to each centroid. This variance is the objective function described as in Eq. 3.1, where m is a real number, μ_{ij} is the membership degree of x_i in the cluster j, x_i is the i_{th} instance of the dataset, and c_j is the centroid of the cluster. Subsequently, the new centroids are recalculated. These steps are iterated until the objective function is minimised.

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{C}\mu_{ij}^{l}||x_i - c_j||^2 \quad , \quad 1 \leq m \leq \infty \tag{3.1}$$

Due to more work during variance calculations, FCM has higher time complexity than k-means. However, it has the benefit of providing membership degrees instead of assigning an instance into one cluster. This is favourable when some specific understandings of datasets are needed such as class overlap, data patterns, mixed information, noise or outliers, *etc.*

3.3.3 The Overlap-Based Undersampling Algorithm

OBU employs a soft clustering algorithm to facilitate the detection and elimination of negative instances from the overlapping region. In this work, we presume that each class possesses its own uniqueness. Thus, for a binary-class dataset, the classes could roughly be represented by two distinct clusters. Then, if any instances have high similarity to that main characteristics of the other class, they are considered as fuzzy instances and are likely to be in the overlapping region. In OBU, such instances are discovered using membership degrees assigned by the soft clustering algorithm. If there is uncertainty in the membership degree, the instance is identified as an overlapped instance. Here we demonstrate and evaluate the OBU method with FCM; however, any soft clustering algorithms can be applied.

Algorithm 1: OBU Algorithm

```
input  : traning set $T = T_{neg} \cup T_{pos}$,
         elimination threshold $\mu_{th}$
output : resampled training set
begin
    $T \leftarrow FCM(T, cluster = 2)$
    $T_{neg\_new} \leftarrow subset(T_{neg}, x_i | \mu_{ineg} \geq \mu_{th})$
    $T_{OBU} \leftarrow T_{neg\_new} \cup T_{pos}$
end
```

Alg. 1 describes the process of OBU. First, FCM is applied to the training set T to determine the representative clusters and assign membership degrees to each sample. The two membership degrees of each sample indicates its likelihood of belonging to the two discovered clusters. It is expected that the two clusters will represent the main characteristics of the negative and positive classes. Then, negative instances that have high membership degrees in the positive cluster and hence low membership degrees in the negative cluster (μ_{neg}) are considered potentially overlap with positive instances, and thus are eliminated from the training set. To determine the cut-off membership degree for potential overlapped instances, the elimination threshold (μ_{th}) is used. Note that μ_{th} is a free parameter and needs to be empirically set across different datasets for the optimal results.

3.3.4 Selection Process

In OBU, when two clusters are created, they may not be readily matched with the two prior class labels. For linearly separable problems, this can be resolved by simply finding the dominant class of the cluster. However, in a complex dataset where both

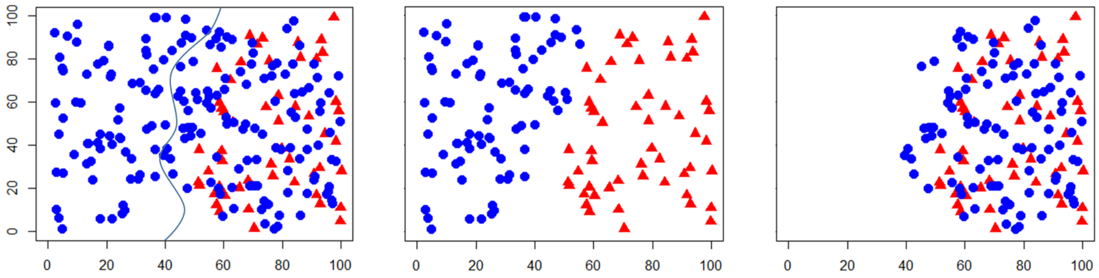


Figure 3.5: Original data with the cluster boundary obtained using FCM clustering (left), correctly undersampled data (middle), incorrectly undersampled data (right)

imbalance and overlap exist, an alternative and principled approach to perform this matching process is needed.

Figure 3.5 illustrates a complex scenario where the data is both imbalanced (minority:majority = 3:10) and highly overlapped as an example. Negative and positive instances are presented with blue circles and red triangles, respectively. Performing FCM clustering on the data results in two clusters showed in the left diagram. Note that in this example, it is assumed that an instance belongs to the cluster where it shows higher membership degree. The between-class border is shown by the grey line. There are 80 and 100 negative instances in the left and the right clusters, respectively. With OBU, the 100 negative instances in right cluster are supposed to be eliminated even though these are the majority of the negative class. Thus, a criterion to eliminate a smaller number of negative instances cannot be applied as a selection process of OBU. It is also worth pointing out that judging from the size of the positive class is not valid for all cases either.

In imbalanced and overlapping datasets, besides this example, there are various problematic cases that prevent the clustering labels to be matched correctly with the actual labels. Therefore, OBU has been adapted to handle such ambiguous scenarios. This is shown in Figure 3.6. Two classification models are built using negative instances in the two clusters (Batch 1 and Batch 2). Then, since the positive class should be more visible in the overlapping region after applying OBU, the model obtained from the correctly undersampled case is expected to yield higher positive class accuracy. The selection is performed at this stage and the other model is discarded.

3.3.5 Time Complexity Analysis

The time complexity of OBU is $O(N)$, where N is the number of instances in the dataset. This is because the main cost of the method is the FCM algorithm, whose time

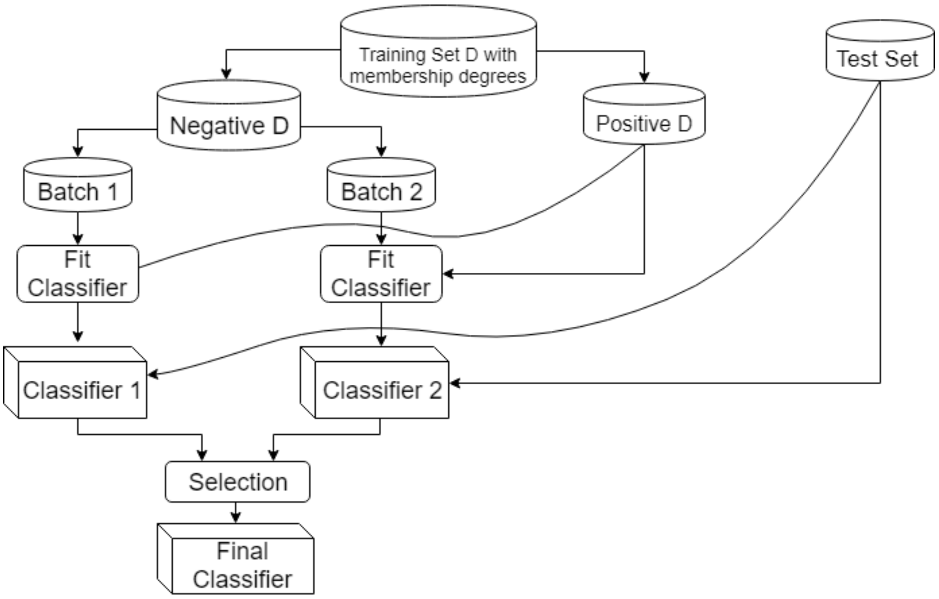


Figure 3.6: Overlap-based undersampling method

complexity is $O(N)$ [126]. Thus, the running time of OBU is comparable to k-means undersampling [14], whose time complexity is also linear in the size of the training set [127]. Moreover, OBU will be faster than DBMUTE and other methods in the SMOTE family including SMOTE, BLSMOTE, SLSMOTE and DBSMOTE, which have the time complexities of $O(N^2)$ [39, 48].

3.4 Experiment

To evaluate the performance of OBU, we carried out an experiment using 36 real-world datasets covering slight to extreme degrees of class imbalance. Results will be compared with the baseline and a state-of-the-art undersampling method.

3.4.1 Setup

Three different classification models were built upon the same datasets with different preprocessing methods as shown in Figure 3.7. The first classifier was trained with the data undersampled using OBU. The second classifier was a result of undersampling using a k-means based approach [14] (kmUnder). Lastly, the baseline classifier was trained using the original training data with no resampling.

Random Forest was chosen as the baseline as it proved to be amongst the top performing traditional machine learning algorithms [29, 128] and a commonly-used classifier for

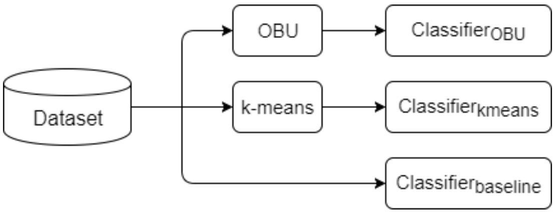


Figure 3.7: Classification models built with different preprocessing methods for evaluation of OBU

imbalanced datasets [11]. Also, RF is similar to the base classifier used in the original work of the k-means based approach [14], which was DT combined with AdaBoost. However, RF is more common in the literature. This will allow comparison of our results with the benchmarking method as well as a wide range of methods across the literature.

The 10-fold cross-validation technique was applied during training for parameter tuning of RF. Only *mtry* was tuned to achieve the best model for each dataset based on the overall accuaracy. For other parameters of RF, the default settings in *caret* package in *R* including $mtree = 500$ were used. Each dataset was partitioned into 80:20 for training and testing. The testing data was only used during model evaluation for the result report. The performance of classification models were measured using common evaluation metrics for imbalanced problems as discussed in Chapter 2. These included sensitivity, which is the accuracy of the class of interest, and BA, which shows the overall performace and was reported in the benchmarking literature [14]. Providing results in terms of these common metrics will also allow comparison of our results with methods across the literature.

For OBU, $\mu_{th} = 0.45$ was used based on empirical results over the values of 0.3, 0.36, 0.42, 0.45, and 0.5. The full code for reproducing the experiment is available on *GitHub*[2].

3.4.2 Datasets

We selected 36 frequently-used datasets in class-imbalance classification. These datasets were obtained from UCI [129] and KEEL repositories [130]. As can be seen in Table 3.1, these datasets vary in terms of size (129 to 5472 instances), imbalance ratio (1.87 to 129.44), and number of features (3 to 19). These variations allowed the method to be tested for its robustness under different situations.

[2] https://github.com/fonkafon/Overlap_based_Undersampling

3.4.3 Results

OBU significantly improved classification results and achieved the most favourable results among the three methods. Results of OBU, k-means based undersampling and the baseline are presented in Table 3.1. The results highlighted in **bold** indicate that OBU achieved the highest result among the methods, where some of these are presented in ***italic*** indicating that they also tied with another method.

As can be seen in Table 3.1, OBU achieved the highest sensitivity and BA on 26 and 19 datasets, respectively. These include the wins in sensitivity on 13 datasets and 13 ties with kmUnder, and the wins in BA on 16 datasets and 3 ties with kmUnder. Most of these ties occurred with the sensitivity value of *100%*. This means that these datasets were linearly separable and applying resampling might not be necessary. It is worth noting that OBU provided the highest results in both metrics on 14 datasets, which far outnumbered kmUnder. Results also show that OBU improved the classification in terms of sensitivity and G-mean on most of the datasets. At the same time, it was unlikely to hurt the classification performance on a linearly separable dataset. This is because OBU undersamples based on class overlap and instance elimination is potentially minimised on a linear separable dataset.

In Table 3.1, results were presented in four groups based on the results of OBU compared to the other methods. In the first group, OBU achieved the highest results in both metrics. This suggests that OBU could improve sensitivity with favourable trade-offs with lower specificity. The second group of results showed wining in sensitivity but not in BA. This occurred due to higher trade-offs between better visibility of minority class instances and information loss in the majority class. In the third group, OBU produced the best results in BA, but not sensitivity. This implies that the elimination of majority class instances by OBU was compromised on these datasets. In other words, more majority class instances could have been eliminated. For the last group, OBU outperformed the baseline but not the k-means based method. The variation in these results might have been due to the inherent data characteristics. Also, it should be noted that these results are based on a global empirical setting of the μ_{th} value. Fine-tuning this value for individual datasets could potentially improve the results further.

To further assess the significance of the improvement using OBU, one-tailed Wilcoxon signed rank tests were carried out. The resulting p-values for OBU paired with the baseline and k-means undersampling on sensitivity were 1.16×10^{-6} and 0.473, and on BA were 0.108 and 0.271, respectively. These statistical results suggest that at the significance level of 0.05, OBU gained statistically significant improvement over the baseline in sensitivity. The improvement in results of OBU over the baseline in BA and

Table 3.1: Experimental results

Dataset	Instances	Imb	Features	OBU		kmUnder		Baseline	
				Sensitivity	BA	Sensitivity	BA	Sensitivity	BA
Abalone09-18	731	16.40	8	**75.00**	**71.81**	50.00	61.86	37.50	68.39
Ecoli1	336	3.36	7	**100.00**	**94.12**	80.00	89.02	80.00	87.06
Ecoli2	336	5.46	7	**90.00**	**92.32**	80.00	90.00	80.00	90.00
Glass016vs2	192	10.29	9	**100.00**	**55.71**	33.33	39.52	0.00	50.00
Glass4	214	15.47	9	**100.00**	**82.50**	50.00	68.75	50.00	73.75
Haberman	306	2.78	3	**75.00**	**63.06**	62.50	61.25	31.25	53.40
Ecoli0137vs26	281	39.14	7	***100.00***	**99.07**	100.00	61.11	100.00	98.15
Ecoli4	336	15.80	7	***100.00***	**100.00**	100.00	98.41	50.00	75.00
New-thyroid1	215	5.14	5	***100.00***	**97.22**	100.00	95.83	85.71	92.86
Vowel0	988	9.98	13	***100.00***	**98.60**	100.00	90.78	94.44	97.22
Yeast5	1484	32.73	8	***100.00***	**96.88**	100.00	94.10	50.00	74.83
Iris0	150	2.00	4	***100.00***	***100.00***	100.00	100.00	100.00	100.00
Page-blocks13vs2	472	15.86	10	***100.00***	***100.00***	100.00	97.73	100.00	100.00
Shuttle2vs4	129	20.50	9	***100.00***	***100.00***	100.00	100.00	100.00	100.00
Glass0	214	2.06	9	**100.00**	64.29	71.43	83.93	57.14	78.57
Glass0123vs456	214	3.20	9	**100.00**	51.56	90.00	91.88	80.00	86.88
Glass1	214	1.82	9	**100.00**	51.85	66.67	74.07	66.67	81.48
Glass6	214	6.38	9	**100.00**	63.51	80.00	88.65	60.00	80.00
Pima	768	1.87	8	**90.57**	50.28	77.36	75.68	64.15	73.08
Vehicle2	846	2.88	18	**100.00**	77.20	97.67	96.44	95.35	97.67
Yeast1	1484	2.46	8	**88.24**	70.42	85.88	74.22	56.47	72.55
Vehicle1	846	2.90	18	***83.72***	53.06	83.72	81.06	58.14	73.07
Ecoli3	336	8.60	7	***85.71***	79.52	85.71	82.86	28.57	63.45
Glass016vs5	184	19.44	9	***100.00***	51.43	100.00	90.00	0.00	50.00
Glass5	214	22.78	9	***100.00***	82.93	100.00	89.02	0.00	50.00
Segmemt0	2308	6.02	19	***100.00***	98.99	100.00	99.37	98.46	99.23
Yeast05679vs4	528	9.35	8	80.00	**85.26**	100.00	75.26	50.00	74.47
Yeast1289vs7	693	22.10	8	33.33	**66.39**	100.00	50.27	16.67	58.06
Yeast1458vs7	459	14.30	8	16.67	**54.55**	50.00	42.80	0.00	50.00
Yeast4	1484	28.10	8	80.00	**84.93**	100.00	50.70	30.00	65.00
Yeast6	1484	41.40	8	71.43	**81.22**	100.00	51.73	42.86	71.26
Abalone19	4174	129.44	8	50.00	57.07	83.33	68.48	0.00	50.00
Glass2	214	11.59	9	66.67	50.00	100.00	70.51	0.00	50.00
Vehicle3	846	2.99	18	78.57	73.81	85.71	80.95	35.71	63.49
Yeast2vs4	514	9.08	8	80.00	89.46	100.00	94.02	50.00	75.00
Yeast3	1484	8.10	8	78.13	84.14	100.00	90.15	62.50	80.68

Table 3.2: Comparative results with evolutionary and deep learning-based methods

Dataset	OBU				EVINCI	CnGRSOMO*	CnGRSOMU*		
	Sensitivity	BA	F1-score	G-mean	G-mean	F1-score	Sensitivity	BA	F1-score
Ecoli2	-	-	-	**92.29**	86.20	-	-	-	-
Yeast5	-	-	-	**96.83**	95.86	-	-	-	-
Ecoli1	-	-	**84.21**	-	-	78.10	-	-	-
Ecoli3	-	-	41.38	-	-	**64.50**	-	-	-
Abalone0918	75.00	71.81	21.05	-	-	-	**83.50**	**84.49**	**39.00**
Yeast4	**80.00**	**84.93**	**34.04**	-	-	-	75.00	80.10	25.00

*Estimated results from graphs.

over the k-means based undersampling may not be significant.

3.5 Performance Comparison with Evolutionary and Deep Learning-Based Methods

In this section, the performance of OBU is compared with evolutionary and deep learning-based methods. This allows further evaluation of OBU in comparison with emerging techniques that are able to handle more complex problems. EVINCI [81], CnGRSOMO [62], CnGRSOMU [62] and SMOTE-CSELM [63] are used as the compared methods. EVINCI is an undersampling method based on EA and ensemble, which mainly deals with overlapped instances. CnGRSOMO and CnGRSOMU are ensemble methods employing deep learning algorithms to rebalance the class distribution by oversampling and undersampling, respectively. SMOTE-CSELM involves class-specific regularization parameter setting in ELM and SMOTE to rebalance data. Detailed discussion of the methods are provided in Chapter 2. These methods were chosen because they were evaluated using common datasets with OBU in the literature.

Table 3.2 and Table 3.3 present the comparative results between OBU and each of the compared methods. The higher value between the methods is highlighted in bold. It should be noted that because these results are based on the availability in the literature, not all measures can be obtained.

Table 3.2 shows that OBU is comparable to EVINCI, CnGRSOMO and CnGRSOMU on the given datasets and metrics. As can be seen in Table 3.3, OBU achieved the highest sensitivity on 14 out of 27 datasets and the highest G-mean on 12 out of 35 datasets. OBU did not perform as well as SMOTE-CSELM, which provided the highest sensitivity and G-mean on 20 and 23 datasets. However, it is worth noting that the results from OBU was obtained using significantly lower time complexity than SMOTE-CSELM, which requires $O(N^3)$ [63]. Compared to OBU with time complexity of $O(N)$, this will

Table 3.3: Comparative results with a deep learning-based method

Dataset	OBU		SMOTE-CSELM	
	Sensitivity	G-mean	Sensitivity	G-mean
Glass1	75.00	71.35	**95.21**	**78.66**
Wisconsin	**100.00**	**100.00**	98.74	97.99
Ecoli01vs5	-	70.71	-	**95.55**
Pima	**90.00**	**100.00**	80.96	76.65
Glass0	**100.00**	71.71	**100.00**	**82.01**
Haberman	**100.00**	**100.00**	77.72	65.92
Vehicle2	**100.00**	73.19	**100.00**	**99.29**
Vehicle1	0.00	**96.82**	**98.00**	86.17
Glass0123vs456	**100.00**	0.00	96.00	**96.02**
Vehicle0	**100.00**	50.92	**100.00**	**99.46**
Newthyroid1	0.00	77.46	**100.00**	**99.16**
Newthyroid2	-	0.00	-	**99.16**
Ecoli2	75.00	89.44	**100.00**	**93.64**
Segment0	**100.00**	63.25	**100.00**	**99.67**
Glass6	**100.00**	**100.00**	**100.00**	95.79
Yeast3	**100.00**	**94.28**	**100.00**	93.54
Ecoli3	87.39	**94.28**	**94.29**	91.52
Page-blocks0	-	92.57	-	**93.97**
Yeast2vs4	88.68	**98.86**	**100.00**	94.73
glass-0-1-5_vs_2	-	32.79	-	**84.75**
Yeast05679vs4	**100.00**	35.24	86.00	**83.18**
Vowel0	-	80.19	-	**100.00**
Glass2	**100.00**	**100.00**	**100.00**	86.87
Shuttle0vs4	100.00	98.04	-	**100.00**
Yeast1vs7	95.35	45.38	**100.00**	**79.58**
Glass4	**100.00**	61.32	**100.00**	**98.22**
Ecoli4	85.71	43.64	**100.00**	**98.40**
Abalone0918	**100.00**	0.00	93.06	**90.61**
Shuttle2vs4	-	68.44	-	**100.00**
Yeast1458vs7	-	**74.10**	-	68.80
Yeast2vs8	33.33	52.70	**100.00**	**80.12**
Yeast1289vs7	-	0.00	-	**74.57**
Ecoli0137vs26	80.00	**82.28**	**100.00**	79.06
Yeast6	75.00	84.47	**100.00**	**89.54**
Abalone19	85.71	**90.48**	**96.00**	79.51

make a substantial difference as the number of samples grows larger.

3.6 Conclusions

In this chapter, an extensive experiment on the impact of class overlap in classification of imbalanced datasets was presented The experiment was carried out at the full scale of class overlap and a wide range of class imbalance degrees including extreme cases. Results showed that classification errors increased with the degree of class overlap regardless of the imbalance degree. On the contrary, the effect of class imbalance highly depended on the presence of class overlap.

Also in this chapter, a new overlap-based undersampling method was proposed. By identifying and removing negative instances from the overlapping region, where misclassification often occurs, positive instances were more visible to the learning algorithm. As a result, statistically significant improvement in sensitivity with relatively small trade-offs with specificity was achieved. OBU proved to enhance the classification of well-known imbalanced datasets and outperformed the state-of-the-art k-means based undersampling in most cases.

These results can be attributed to some advantages of OBU as follows. First, the amount of undersampling is proportional to the overlap degree. Second, the method is unlikely to eliminate instances outside the overlapping region, which lessens information loss. However, OBU has some limitations that need to be addressed for further improvement. There were variations in the experimental results. Some results suggested insufficient elimination; on the contrary, some implied excessive elimination of negative class instances. This may have been partly due to the global setting of the elimination threshold. Thus, a threshold that is adaptive to class overlap and class imbalance may be a good solution to the issue. Also, enhancing the clustering algorithm's performance for more accurate identification of overlapped instances may also reduce excessive elimination. These limitations and possible development of the method led to an extension of OBU with significant improvement presented in the next chapter.

Chapter 4

Adaptive & Boosted OBU

Following the work discussed in Chapter 3, in this chapter, two new methods that extended and improved the performance of OBU are proposed. The two methods were developed to achieve more accurate identification and removal of overlapped negative instances. Thorough evaluations using simulated and real-world datasets covering an extensive range of imbalance and overlap scenarios including extreme cases were carried out. Results showed statistically significant improvements over OBU, which were competitive with well-established and state-of-the-art methods. *The report of this work is to be published in the International Journal of Neural Systems [131].*

4.1 Overview

In Chapter 3, the Overlap-Based Undersampling method was introduced and shown to perform well on several real-world imbalanced datasets. The method aimed at maximising the visibility of positive class instances by eliminating negative instances in the overlapping region. With the presumption that each class possessed its own uniqueness, identifying overlapped negative instances was based on soft clustering results. FCM was employed to discover two distinct clusters. Then, any negative instances with high similarity to the positive cluster' properties, were considered to be in the overlapping region and hence removed. OBU showed improvement over the baseline, especially in sensitivity. However, some limitations of the method, such as an empirical setting of the elimination threshold and excessive elimination of negative instances need to be addressed for further improvement.

In this chapter, new methods, Boosted OBU (BoostOBU) and Adaptive-threshold OBU (AdaOBU), which extended OBU with some improvements, are presented. The main objective of both methods is to provide more accurate identification and elimination of overlapped negative instances. AdaOBU extends OBU by incorporating an adaptive elimination threshold that is based on the overlap degree. This replaces the fine-tuning process and enables better generalisation across different scenarios. BoostOBU is a hybrid approach that integrates OBU and an oversampling method to emphasise the presence of borderline minority class instances. By doing so, we hypothesise that more accurate clustering and hence more precise identification of overlapped negative instances will be achieved.

4.2 The Adaptive & Boosted OBU Methods

This section presents and discusses AdaOBU and BoostOBU in detail. A brief overview of a BLSMOTE [59], which is a related algorithm used in BoostOBU is also provided.

4.2.1 BLSMOTE: Borderline-SMOTE

BLSMOTE is an improvement of SMOTE [12] by oversampling only borderline samples [59]. The rationale of this approach is that samples far from the borderline are less likely to be misclassified and hence contribute less to the classification. In BLSMOTE, minority class samples are identified as "*danger*", "*safe*" and "*noise*" based on the number of majority class samples in their k nearest neighbours. If none of the nearest neighbours belongs to the minority class, the sample is considered as noise. The sample is *safe* if its nearest neighbours consist of fewer majority class samples than minority class samples. Otherwise, the sample is marked as *danger*, which indicates that it is likely to be in the borderline region. Only *danger* samples are then used for oversampling by the same linear interpolation technique used in SMOTE.

BLSMOTE has two models – BLSMOTE1 and BLSMOTE2. BLSMOTE1 only generates new instances from the *danger* samples and their minority class nearest neighbours whereas in BLSMOTE2 all nearest neighbours are considered regardless of class.

4.2.2 AdaOBU: Adaptive-threshold OBU

AdaOBU incorporates an adaptive elimination threshold in OBU allowing the method to be more generalised across datasets with varying degrees of class overlap. The adaptive threshold is self-adjusting to the fuzziness of the dataset, which is measured by the overall similarity of instances to their own class' properties. By this definition, it can be

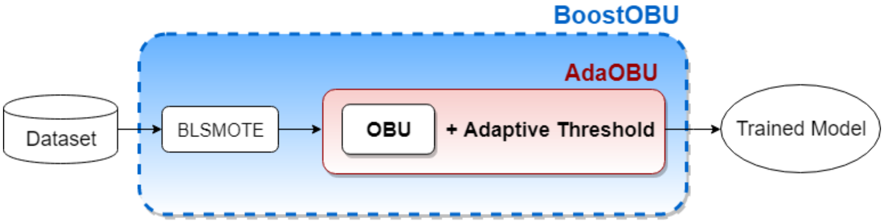


Figure 4.1: A diagram showing the extensions of OBU by AdaOBU and BoostOBU

said that a dataset is fuzzier than another one if larger percentage of its instances have indistinct membership degrees. This also implies higher class overlap in the dataset.

The algorithm of AdaOBU is shown in Alg. 2. Line $3-5$ express how the adaptive threshold is computed. First, the average membership degrees of all negative instances belonging to the negative cluster ($\bar{\mu}_{neg}$) and the positive cluster ($\bar{\mu}_{pos}$) are calculated. Then, the minimum between $\bar{\mu}_{neg}$ and $\bar{\mu}_{pos}$ is used as the elimination threshold (μ_{th}). The rationale behind this is as follows. The difference between the two means ($|\bar{\mu}_{neg} - \bar{\mu}_{pos}|$) indicates the fuzziness of the dataset. Note that according to FCM, the membership degree ranges between 0 and 1; thus, $|\bar{\mu}_{neg} - \bar{\mu}_{pos}|$ is also within the range of 0 and 1. In an extreme case when $|\bar{\mu}_{neg} - \bar{\mu}_{pos}| = 0$, where both means are 0.5, none of the clusters shows distinct nature of the negative class suggesting possibility of very high overlapping between the two classes. And the opposite applies in the other extreme case of $|\bar{\mu}_{neg} - \bar{\mu}_{pos}| = 1$, where one mean is 0 and the other is 1. Accordingly, we can say that the overlapping degree and hence elimination amount are to be proportional to the smaller value between $\bar{\mu}_{neg}$ and $\bar{\mu}_{pos}$. Finally, the elimination process is followed.

Algorithm 2: AdaOBU Algorithm

```
input  : traning set $T = T_{neg} \cup T_{pos}$
output : resampled training set
1 begin
2     $T \leftarrow FCM(T, cluster = 2)$
3     $\bar{\mu}_{neg} \leftarrow mean(\mu_{ineg} | x_i \in T_{neg})$
4     $\bar{\mu}_{pos} \leftarrow mean(\mu_{ipos} | x_i \in T_{neg})$
5     $\mu_{th} \leftarrow min(\bar{\mu}_{neg}, \bar{\mu}_{pos})$
6     $T_{neg\_new} \leftarrow subset(T_{neg}, x_i | \mu_{ineg} \geq \mu_{th})$
7     $T_{AdaOBU} \leftarrow T_{neg\_new} \cup T_{pos}$
8 end
```

4.2.3 BoostOBU: Boosted OBU

BoostOBU is presented to improve the detection of negative instances in the overlapping region, hence reducing excessive elimination. We hypothesised that erroneous elimination of OBU could have been partly due to low visibility of positive instances within the overlapping region, which caused poor performance of the clustering algorithm. To address this issue, BoostOBU was developed to improve the presence of the positive class, especially along the borderline, before applying clustering. Moreover, the adaptive elimination threshold proposed in 4.2.2 is also used in BoostOBU.

To serve the purpose of emphasising the border of the minority class, BLSMOTE1 was selected. The choice of using this method can be justified as follows. Firstly, BLSMOTE proved to successfully improve the visibility of minority class borders to the learning algorithm [59]. This was evidenced by higher TPR achieved over SMOTE and random oversampling. Secondly, since noisy samples are not considered for oversampling, the effect of noise would not be enlarged. Thirdly, BLSMOTE1 only synthesises based on minority class samples, thus it is ensured that the minority class' border is highlighted rather than being expanded.

```
**Algorithm 3:** BoostOBU Algorithm
input  : traning set $T = T_{neg} \cup T_{pos}$
output : resampled training set
begin
    $(T_{BS} = T_{neg} \cup T_{pos\_new}) \leftarrow BLSMOTE(T)$
    $T \leftarrow FCM(T_{BS}, cluster = 2)$
    $\bar{\mu}_{neg} \leftarrow mean(\mu_{ineg} | x_i \in T_{neg})$
    $\bar{\mu}_{pos} \leftarrow mean(\mu_{ipos} | x_i \in T_{neg})$
    $\mu_{th} \leftarrow min(\bar{\mu}_{neg}, \bar{\mu}_{pos})$
    $T_{neg\_new} \leftarrow subset(T_{neg}, x_i | \mu_{ineg} \geq \mu_{th})$
    $T_{BoostOBU} \leftarrow T_{neg\_new} \cup T_{pos\_new}$
end
```

Alg. 3 outlines BoostOBU algorithm, which integrates both oversampling and undersampling techniques. BLSMOTE is first applied and followed by overlap-based undersampling with the adaptive elimination threshold. As illustrated in Figure 4.1, AdaOBU is incorporated into the design of BoostOBU.

4.2.4 Time Complexity Analysis

The time complexity of AdaOBU is $O(N)$ because OBU and the calculation of the adaptive threshold each requires $O(N)$. BoostOBU has the time of complexity of $O(N^2)$, which is the running time required by BLSMOTE [39]. Thus, AdaOBU is comparable to k-means undersampling [14] in terms of time complexity and faster than SMOTE-base extensions, whose time complexities will be at least quadratic [39, 48].

4.3 Experiments

Extensive experiments covering a wide range of imbalanced and overlapped datasets were carried out. This includes 66 synthetic datasets and 68 public real-world datasets, 2 of which are large and high-dimensional. Results were compared against well-established methods and state-of-the-art methods. The Friedman test and 1xN post-hoc Wilcoxon signed rank tests with Holm correction were carried out to assess the significance of the result improvement. For reproducibility, the code of AdaOBU and BoostOBU as well as the simulated datasets used is available on *GitHub*[1].

4.3.1 Setup

Three sets of experiments were carried out. Simulated datasets and small to medium-sized real-world datasets were used in Experiment I and Experiment II, respectively. Experiment III was carried out on larger and more complex real-world datasets. SVM, one of the most common learning algorithms for imbalanced datasets [11], was chosen as the learning algorithm. Sensitivity, specificity, G-mean, and F1-score were used to assess the methods. Results were compared against SMOTE [12], BLSMOTE [59], kmUnder [14] and OBU [18]. In Experiment II, two additional experiments were carried out to further evaluate the methods – 1) comparisons against more robust methods, namely SMOTE-ENN [132], SMOTEBagging [68] (SMTBagging) and RUSBoost [67], and 2) comparisons using different learning algorithms that are Decision Tree (J48), kNN and RF. This selection of various classification algorithms and evaluation metrics, which are commonly used in the literature, will also allow the reader to compare our results with other methods.

4.3.2 Datasets

In Experiment I, we used 66 simulated binary-class datasets, which cover a wide range of class overlap and class imbalance degrees. To evaluate the performance of our methods in relation to the changes in class imbalance and class overlap, the datasets were uniformly

[1] https://github.com/fonkafon/BoostedOBU

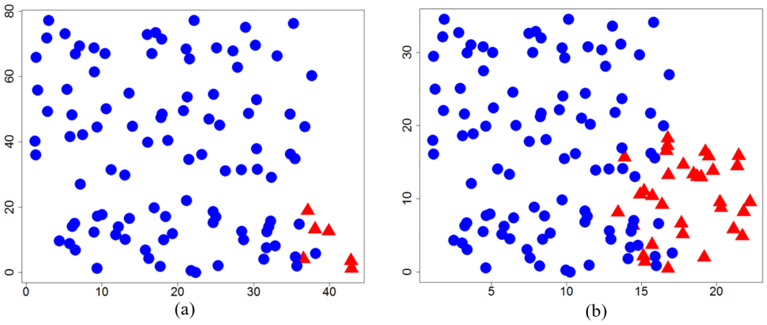


Figure 4.2: Examples of two synthetic datasets with 50% class overlap and (a) $imb = 15$ and (b) $imb = 3$.

distributed in two-dimensional space (*i.e.* data densities of the positive and negative classes were equal within a dataset).

All datasets were generated with a fixed number of negative samples and fixed data space of positive samples. In each dataset, the number of positive samples was based on the imbalance degree, and the density of the negative class was made equal to that of the positive class. This enabled us to obtain many variations of datasets. Figure 4.2 illustrates two examples of simulated datasets. Both datasets have 100 negative samples. There are 6 positive samples in Figure 4.2(a) and 33 positive samples in Figure 4.2(b) making $imb = 15$ and $imb = 3$, respectively. Note that the axes of the two plots are of different scales. The density of data in Figure 4.2(a) is lower than that in Figure 4.2(b).

For Experiment I, we simulated datasets with the imbalance degrees of $1.5, 3, 12, 30, 60$ and 120. At each imbalance degree, the overlap degree ranged from $0\% - 100\%$ in a step of 10. The number of negative instances generated in each dataset was $6,000$ while the number of positive instances was varied between $50 - 4,000$ based on the imbalance degree.

In Experiment II, 66 datasets from *UCI Repository* [129] and *KEEL Repository* [130] were used. As shown in Table 4.1, the datasets vary in imbalance degrees (*1.82-129.44*), number of features (*3-34*), and number of instances (*92-5,472*).

Experiment III was carried out on large and high-dimensional datasets. These were the breast cancer dataset from *KDD Cup 2008*[2] and the handwritten digits dataset from the *MNIST* database [133]. The breast cancer dataset is binary-class with 117 features and 102,294 samples. It contains 101,671 negative and 623 positive samples, which makes $imb = 163.20$. The handwritten digits dataset is 10-class with 784 features and 60,000 samples. As AdaOBu and BoostOBU are designed for binary-class datasets,

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

Table 4.1: Datasets used in Experiment II

Dataset	Instances	Imb	f
Glass1	214	1.82	9
Ecoli0vs1	220	1.86	7
Wisconsin	683	1.86	9
Pima	768	1.87	8
Iris0	150	2.00	4
Glass0	214	2.06	9
Yeast1	1484	2.46	8
Haberman	306	2.78	3
Vehicle2	846	2.88	18
Vehicle1	846	2.90	18
Vehicle3	846	2.99	18
Glass0123vs456	214	3.20	9
Vehicle0	846	3.25	18
Ecoli1	336	3.36	7
Newthyroid1	215	5.14	5
Newthyroid2	215	5.14	5
Ecoli2	336	5.46	7
Segment0	2308	6.02	19
Glass6	214	6.38	9
Yeast3	1484	8.10	8
Ecoli3	336	8.60	7
Pageblocks0	5472	8.79	10
Yeast2vs4	514	9.08	8
Ecoli067vs35	222	9.09	7
Glass015vs2	172	9.12	9
Yeast02579vs368	1004	9.14	8
Ecoli046vs5	203	9.15	6
Ecoli0267vs35	224	9.18	7
Glass04vs5	92	9.22	9
Ecoli0346vs5	205	9.25	7
Yeast05679vs4	528	9.35	8
Vowel0	988	9.98	13
Ecoli067vs5	220	10.00	6
Ecoli0147vs2356	336	10.59	7
Led7digit02456789vs1	443	10.97	7
Ecoli01vs5	240	11.00	6
Glass0146vs2	205	11.06	9
Glass2	214	11.59	9
Cleveland0vs4	177	12.62	13
Ecoli0146vs5	280	13.00	6
Shuttle0vs4	1829	13.87	9
Yeast1vs7	459	14.30	7
Glass4	214	15.46	9
Ecoli4	336	15.80	7
Pageblocks13vs2	472	15.86	10
Abalone0918	731	16.40	8
Dermatology6	358	16.90	34
Glass016vs5	184	19.44	9
Shuttle2vs4	129	20.50	9
Yeast1458vs7	693	22.10	8
Glass5	214	22.78	9
Yeast2vs8	482	23.10	8
Yeast4	1484	28.10	8
Winequalityred4	1599	29.17	11
Yeast1289vs7	947	30.57	8
Winequalityred8vs6	656	35.44	11
Ecoli0137vs26	281	39.14	7
Abalone21vs8	581	40.50	8
Yeast6	1484	41.40	8
Winequalitywhite3vs7	900	44.00	11
Winequalityred8vs67	855	46.50	11
Abalone19vs10111213	1622	49.69	8
Winequalitywhite39vs5	1482	58.28	11
Shuttle2vs5	3316	66.67	9
Winequalityred3vs5	691	68.10	11
Abalone19	4174	129.44	8

we treated the handwritten digits dataset as a binary-class problem using the one-vs-all scheme. Two binary-class datasets were made of class3-vs-all and class5-vs-all. Class 3 and class 5 were chosen as the minority class in the two datasets since they were ones of hard-to-classify numbers and even the most challenging classes for a state-of-the-art deep learning-based method [91]. In each dataset, the selected minority class was undersampled in order to have a higher imbalance degree. In the first dataset, MNIST_3, class 3 was undersampled such that $imb = 43.90$, which is made up of 53,869 negative and 1,227 positive instances. In the second dataset,MNIST_5, class 5 was undersampled such that $imb = 20.13$, which consists of 53,869 negative and 2,711 positive instances.

For all datasets, the partitioning of 80:20 of training to testing sets was used. To diminish the effect of noisy instances, the training data was normalised using standard scores. The holdout testing data was only used during model evaluation for the result report. In Experiment I and II, 10-fold cross-validation was employed in the training phase for the purpose of automatic parameter tuning of the classification model. Follow the methods available in the *caret* package in R, $cost\,(C)$ of SVM, $mtry$ of DT and RF, and k of kNN were tuned to obtain the best models based on the overall accuracy. No cross-validation was applied in Experiment III as the datasets are sufficiently large.

4.3.3 Parameter Settings

To provide a fair comparison, no parameter tuning was performed for the resampling methods. AdaOBU has no free parameters. In BoostOBU, the k value in BLSMOTE was set to 5, and no other parameter settings were required.

For SMOTE [12], BLSMOTE [59], OBU [18], and SMOTE-ENN [132] the same parameter settings as reported in the original work were used. These were $k = 5$ in SMOTE and BLSMOTE, and $\mu_{th} = 0.45$ in OBU. In SMOTE-ENN, $k = 5$ and $k = 3$ were set for SMOTE and ENN, respectively. As for SMOTEBagging [68], 40 weak learners were used as suggested by [134].

The Radial Basis Function kernel was used for SVMs with the default setting in *caret* package in R of $\gamma = \frac{1}{f}$, where f is the number of features in the dataset. For RF, the number of trees ($mtree$) was 500. Lastly, $k = 5$ was used for kNN.

4.4 Results & Discussions

The experimental results are discussed in the following order – Experiment I: Simulated datasets, Experiment II: Small to medium-sized real-world datasets, Experiment III:

Table 4.2: Average results and statistical test results from Experiment I

	Baseline	SMOTE	BLSMOTE	kmUnder	OBU	AdaOBU	BoostOBU
sensitivity	67.75*†	98.11†	98.56†	98.35†	97.46†	97.50†	**99.50**
specificity	96.20*†	**90.35***†	89.87*†	89.63*	84.38*†	85.41†	87.47
G-mean	77.86*†	**93.87***	93.78*	91.71†	90.43*†	91.00†	92.41
F1-score	69.88*†	76.41*	75.59*	68.54*†	57.57*†	62.04†	**77.73**

*The difference in results of the method and of AdaOBU is statistically significant.
†The difference in results of the method and of BoostOBU is statistically significant.

Large high-dimensional real-world datasets.

4.4.1 Experiment I: Simulated datasets

Experimental results on 66 simulated datasets with $imb = 1.5$ to 120 and overlap degrees from 0% to 100% are shown in Fig. 4.3. The performance of OBU and the proposed extensions are marked with dashed lines, and the results of the other methods are marked with solid lines. The shaded areas are the areas under the performance curves of the baseline (SVM with no resampling applied).

BoostOBU achieved the top performance across all metrics in most imbalance and overlap scenarios. AdaOBU showed competitive performance with OBU across all metrics and provided comparable results with well-established and state-of-the-art methods, especially at higher imbalance degrees.

In Fig. 4.3, AdaOBU showed clear improvement over the baseline in sensitivity and G-mean across most imbalance and overlap degrees. This is also confirmed by its average performance across 66 scenarios given in Table 4.2, where the top result in each metric is highlighted in bold. The symbols next to each value indicate the results of the significance tests at 95% confidence level comparing the results cross 66 datasets. An asterisk (*) denotes a statistically significant difference between the method and AdaOBU, and a dagger (†) denotes a statistically significant difference between the method and BoostOBU.

Table 4.2 shows that AdaOBU improved sensitivity from 67.75% to 97.5% and G-mean from 77.86% to 91% on average. As can be seen in Fig. 4.3, AdaOBU was competitive in sensitivity, specificity and G-mean with SMOTE, BLSMOTE and kmUnder at higher imbalance degrees. However, AdaOBU suffered from high FP, especially when the imbalance and overlap degrees were high. This must have been caused by excessive elimination as a result of inaccurate identification and removal of negative instances. More excessive elimination was likely to occur at higher imbalance and overlap degrees,

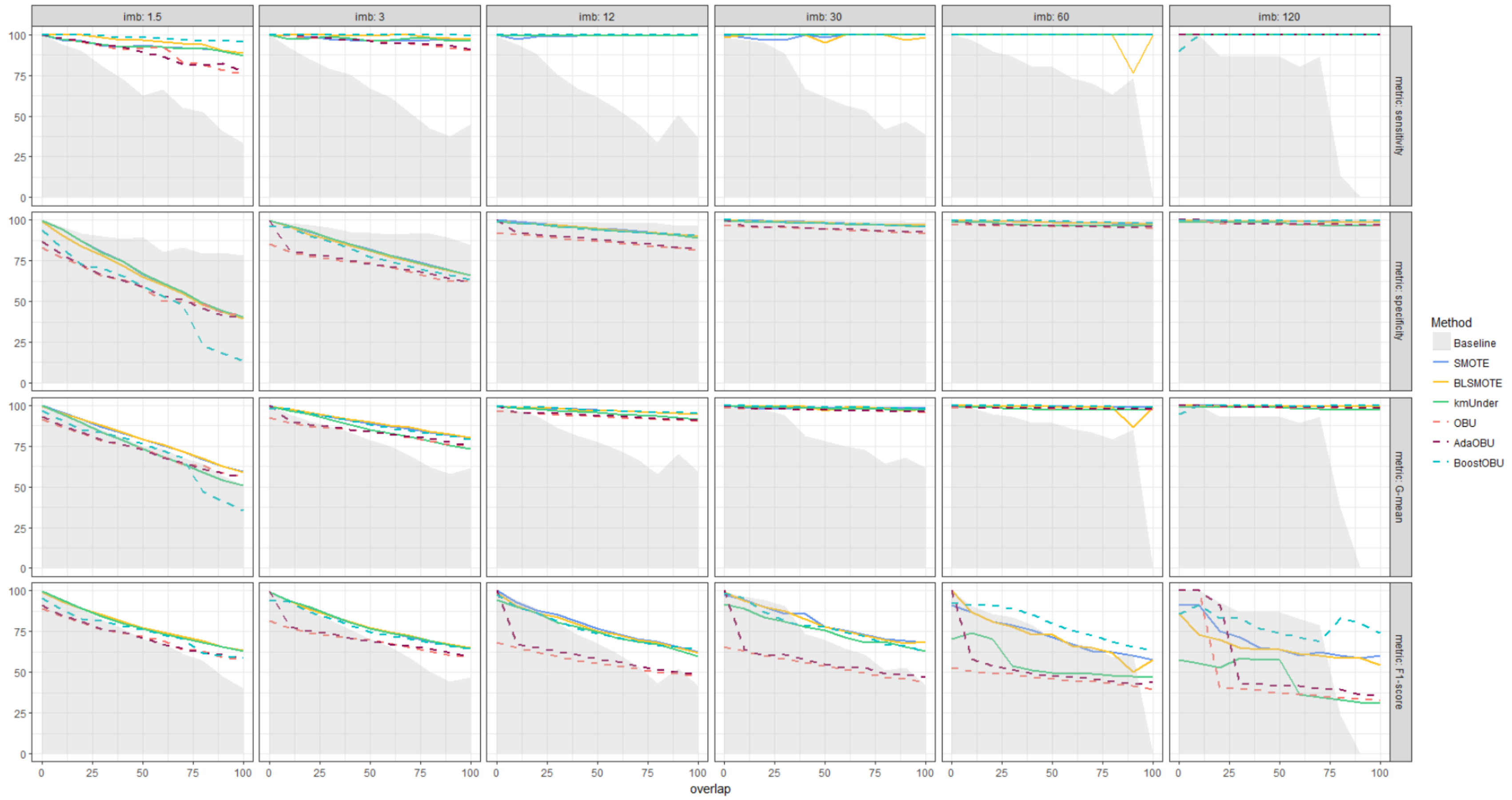


Figure 4.3: Performance of the methods in terms of sensitivity, specificity, G-mean and F1-score across various imbalance and overlap degrees in Experiment I, where each column of the subplots shows the results on a specific imbalance degree of data with the full range of class overlap.

where the visibility of the minority class to the clustering algorithm was lower, resulting in poorer performance of the clustering algorithm. Only in a few cases with no overlap or slight overlap that AdaOBU showed the smallest FP compared to the other methods since a smaller number of negative instances was removed. As shown in Table 4.2, AdaOBU achieved higher F1-score and had competitive sensitivity, specificity and G-mean with OBU on average indicating less excessive elimination. Therefore, the proposed adaptive threshold has shown to be able to effectively replace the free parameter in OBU.

From Table 4.2, BoostOBU achieved the highest average sensitivity (99.5%) and F1-score (77.73%). Even though the average specificity of BoostOBU was not as high as SMOTE, BLSMOTE and kmUnder, Fig. 4.3 shows that BoostOBU, in fact, provided competitive specificity with those methods across most imbalance degrees. The exception occurred at very low imbalance degrees, especially at $imb = 1.5$, where BoostOBU outperformed the other methods in sensitivity but suffered from low specificity. Similarly, at all imbalance and overlap levels, except at $imb = 1.5$, BoostOBU often achieved the highest G-mean among all methods. This indicates a good trade-off between the accuracy of the positive and the negative classes achieved by BoostOBU. In terms of F1-score, BoostOBU performed competitively with SMOTE, BLSMOTE and kmUnder. However, Fig. 4.3 shows that BoostOBU clearly outperformed these methods at very high to extreme imbalance degrees, *i.e.* $imb = 60$ and 120. These results indicate that BoostOBU not only could provide the highest sensitivity but also significantly reduced the number of FP from OBU.

In conclusion, the competitive and higher results of AdaOBU compared to OBU across a wide range of overlap and imbalance scenarios proved that the proposed adaptive threshold could potentially replace parameter tuning in OBU. The significantly better performance of BoostOBU over OBU and AdaOBU across all metrics (Table 4.2) suggests that emphasizing the presence of borderline positive instances helped improve the detection of overlapped negative instances. Moreover, BoostOBU outperformed all of the well-established and state-of-the-art methods in sensitivity and F1-score while achieving competitive performance in specificity and G-mean. These results show that BoostOBU provided the most optimized solution among the methods.

Adaptive threshold analysis

We had collected the threshold values for analyzing its relation to imbalance and overlap degrees. Results verified that the adaptive threshold was successfully proportional to the amount of overlapped samples.

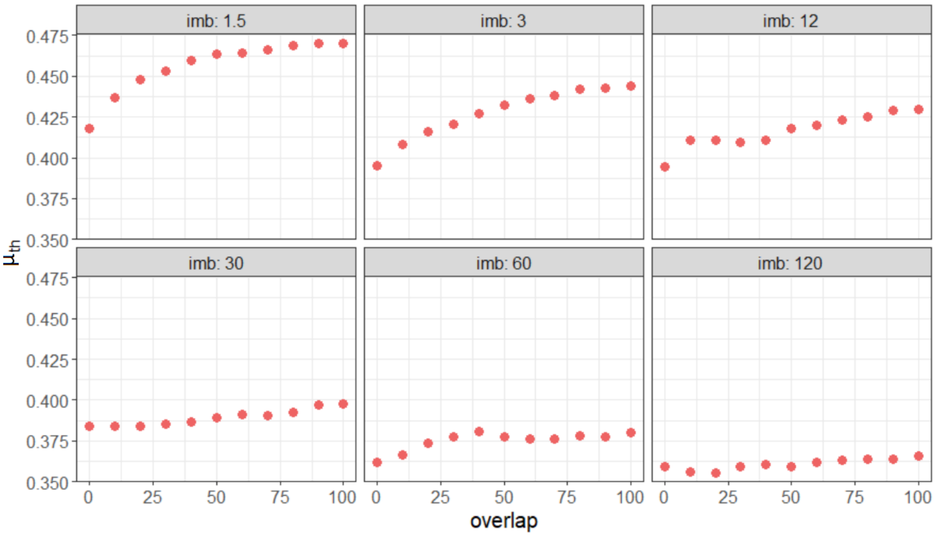


Figure 4.4: The adaptibility of the elimination threshold to imbalance and overlap degrees

Figure 4.4 presents the plots of the adaptive threshold (μ_{th}) in different scenarios of imbalance and overlap degrees. Across all imbalance degrees, the plots show a clear trend of μ_{th} increasing with the degree of class overlap, which was as hypothesised. From no overlap to complete overlap, the changes in μ_{th} were 12.41% at $imb = 1.5$, 12.34% at $imb = 3$, 8.88% at $imb = 12$, 3.59% at $imb = 30$, 5.07% at $imb = 60$ and 1.82% at $imb = 120$. This shows that the adaptive threshold was able to adapt to a change in the overlap degree making the elimination amount directly proportional to the degree of class overlap.

It can also be observed in Figure 4.4 that μ_{th} is inversely proportional to the imbalance degree. As discussed earlier in Section 4.3.2 and shown in Figure 4.2, at a higher imbalance degree, there were fewer negative instances in the overlapping region. Consequently, fewer negative instances were removed. This is another evidence that μ_{th} was able to self-adjust to different overlap scenarios.

4.4.2 Experiment II: Real-world datasets

The performance of AdaOBU and BoostOBU on 66 real-world datasets was consistent with that in Experiment I, apart from slight variations in the ranks, which was partly due to more comparison methods added in this experiment. AdaOBU and BoostOBU were among the methods that provided highest sensitivity. Their G-mean and F1-score were also comparable with others. For the ease of discussion, Table 4.3-4.6 show the results of 24 representative datasets sorted from low to high imbalance degrees as examples. These 24 examples were selected to cover all ranges of imbalance ratios and all performance behaviors of 66 datasets. However, the discussion will be based on the results of the 66 datasets, whose detailed results are available in Appendix A.

In Table 4.3-4.6, ranks based on the performance compared across all methods are also provided next to the metric values. Rank 1 indicates the best performance across all methods on the dataset, and so on. The average rank of each method and significance test results across all 66 datasets are provided.

As seen in Table 4.3, AdaOBU achieved the highest sensitivity rank followed by OBU, kmUnder and BoostOBU. AdaOBU provided the highest sensitivity on 41 datasets and BoostOBU on 31 datasets. More importantly, both methods outperformed the ensemble-based methods, namely SMTBagging and RUSBoost. In particular, AdaOBU was significantly better than SMTBagging as well as the baseline, SMOTE, BLSMOTE and SMOTE-ENN in sensitivity. The imbalance degree did not seem to affect the performance of AdaOBU and BoostOBU, which was consistent with the results in Experiment I.

Non-winning cases in sensitivity of AdaOBU may have been due to other variations such as data density that we have not considered in this work. In most cases that AdaOBU improved the sensitivity over OBU, highest sensitivity was achieved. There were few exceptions, for example, on Shuttle2vs4, where BoostOBU improved the performance further from AdaOBU and had the highest sensitivity. The decreases in sensitivity on Vehicle1, Vehicle3, Yeast1vs7 and Yeast2vs8 from OBU evidence unsuccessful cases of the adaptive threshold. Since the adaptive threshold is solely distance-based, other factors such as local data density may have caused the inaccuracy during the clustering process. Similarly, the results on Cleveland0vs4, Yeast4, Winequalityred8vs6 and some others where none of the OBU-based methods won suggested that considering only the distance factor may not be sufficient. Many non-winning cases of BoostOBU over AdaOBU such as Vehicle3, Yeast1289vs7 and Abalone19vs10111213 were highly likely affected by the poor performance of BLSMOTE as can be seen in Table 4.3.

Table 4.4 shows that all methods commonly led to decreases in specificity. These were due to the trade-offs for higher sensitivity, except for SMOTE-ENN, which had poorer performance than the baseline in both sensitivity and specificity. AdaOBU, which achieved the highest average rank in sensitivity, had the lowest specificity on average. This indicates that the trade-offs of AdaOBU were high, which may not be suitable for some application domains.

Comparing BoostOBU with the other methods, its winning over BLSMOTE and

Table 4.3: Sensitivity results from Experiment II

Dataset	Sensitivity Value/Rank																			
	Baseline		SMOTE		BLSMOTE		kmUnder		SMOTE-ENN		SMTBagging		RUSBoost		OBU		AdaOBU		BoostOBU	
Glass1	25.00	9	37.50	7	62.50	4	62.50	4	12.50	10	37.50	7	62.50	4	75.00	2	**87.50**	**1**	75.00	2
Glass0	80.00	8	90.00	3	60.00	10	90.00	3	80.00	8	90.00	3	90.00	3	90.00	3	**100.00**	**1**	**100.00**	**1**
Vehicle1	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	90.00	8	**100.00**	**1**	**100.00**	**1**	90.00	8	90.00	8
Vehicle3	0.00	7	0.00	7	33.33	3	33.33	6	33.33	3	33.33	3	0.00	7	**100.00**	**1**	66.67	2	0.00	7
Vehicle0	66.67	8	60.00	9	80.00	5	73.33	7	60.00	9	80.00	5	86.67	4	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**
Ecoli2	80.00	4	**100.00**	**1**	80.00	4	60.00	9	60.00	9	80.00	4	**100.00**	**1**	80.00	4	**100.00**	**1**	80.00	4
Segmemt0	18.75	10	50.00	6	43.75	8	56.25	4	31.25	9	56.25	4	50.00	6	75.00	2	68.75	3	**81.25**	**1**
Pageblocks0	81.98	10	92.79	5	95.50	3	91.89	6	93.69	4	91.89	7	90.09	8	87.39	9	**100.00**	**1**	**100.00**	**1**
Glass015vs2	0.00	9	33.33	4	33.33	4	**66.67**	**1**	0.00	9	33.33	4	33.33	4	**66.67**	**1**	**66.67**	**1**	33.33	4
Vowel0	89.23	9	96.92	6	89.23	9	**100.00**	**1**	98.46	5	95.38	8	95.38	7	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**
Cleveland0vs4	50.00	2	0.00	6	0.00	6	**100.00**	**1**	0.00	6	0.00	6	50.00	2	50.00	2	50.00	2	0.00	6
Yeast1vs7	41.86	9	88.37	5	81.40	8	95.35	2	25.58	10	88.37	5	95.35	2	**95.35**	**1**	93.02	4	88.37	5
Ecoli4	38.10	9	71.43	7	76.19	6	85.71	3	21.43	10	69.05	8	78.57	5	85.71	2	**90.48**	**1**	83.33	4
Shuttle2vs4	41.18	10	74.12	6	78.82	4	71.76	7	51.76	9	71.76	8	75.29	5	82.35	3	85.88	2	**87.06**	**1**
Yeast2vs8	0.00	9	**33.33**	**1**	**33.33**	**1**	33.33	6	0.00	9	**33.33**	**1**	16.67	7	**33.33**	**1**	16.67	7	**33.33**	**1**
Yeast4	70.00	9	90.00	2	**100.00**	**1**	90.00	2	40.00	10	80.00	7	90.00	2	80.00	7	90.00	2	90.00	2
Yeast1289vs7	75.00	3	75.00	3	0.00	8	50.00	6	0.00	8	75.00	3	**100.00**	**1**	0.00	8	**100.00**	**1**	25.00	7
Winequalityred8vs6	0.00	10	33.33	8	**100.00**	**1**	**100.00**	**1**	66.67	7	33.33	8	**100.00**	**1**	66.67	4	66.67	4	66.67	4
Yeast6	62.50	9	**100.00**	**1**	87.50	6	75.00	7	50.00	10	**100.00**	**1**	**100.00**	**1**	75.00	7	**100.00**	**1**	**100.00**	**1**
Winequalityred8vs67	0.00	6	0.00	6	0.00	6	**100.00**	**1**	0.00	6	0.00	6	33.33	5	**100.00**	**1**	**100.00**	**1**	66.67	4
Abalone19vs10111213	0.00	10	16.67	4	16.67	4	33.33	3	16.67	8	16.67	4	16.67	8	**50.00**	**1**	**50.00**	**1**	16.67	4
Winequalitywhite39vs5	0.00	7	0.00	7	20.00	6	**100.00**	**1**	0.00	7	0.00	7	60.00	2	40.00	3	40.00	3	40.00	3
Winequalityred3vs5	0.00	10	50.00	2	50.00	2	**100.00**	**1**	50.00	2	50.00	2	50.00	2	50.00	2	50.00	2	50.00	2
Abalone19	71.43	8	85.71	3	85.71	3	71.43	9	42.86	10	85.71	3	**100.00**	**1**	85.71	3	**100.00**	**1**	85.71	3
Average‡		5.64*†		3.73*		4.82*†		2.77		5.88*†		3.91*		3.17		2.12		**2.09**†		2.91

‡The average and the significance test results are based on 66 datasets.

Table 4.4: Specificity results from Experiment II

Dataset	Specificity Value/Rank																			
	Baseline		SMOTE		BLSMOTE		kmUnder		SMOTE-ENN		SMTBagging		RUSBoost		OBU		AdaOBU		BoostOBU	
Glass1	**99.27**	**1**	91.24	4	86.13	6	84.67	7	98.54	2	91.97	3	81.75	8	67.88	9	61.31	10	87.59	5
Glass0	**100.00**	**1**	**100.00**	**1**	92.86	7	98.21	5	**100.00**	**1**	**100.00**	**1**	96.43	6	57.14	9	55.36	10	67.86	8
Vehicle1	**100.00**	**1**	96.88	6	**100.00**	**1**	93.75	7	90.63	9	**100.00**	**1**	**100.00**	**1**	93.75	7	**100.00**	**1**	59.38	10
Vehicle3	**100.00**	**1**	94.29	5	94.29	5	57.14	8	97.14	3	94.29	5	**100.00**	**1**	31.43	9	28.57	10	97.14	3
Vehicle0	**88.89**	**1**	81.48	2	70.37	5	66.67	6	77.78	3	77.78	3	59.26	7	25.93	10	51.85	8	51.85	9
Ecoli2	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	94.59	10	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**
Segmemt0	**100.00**	**1**	57.78	6	55.56	8	66.67	3	71.11	2	62.22	4	55.56	7	53.33	9	57.78	5	33.33	10
Pageblocks0	**98.17**	**1**	94.20	7	91.65	8	95.21	4	96.54	3	94.60	5	94.40	6	98.07	2	23.01	10	36.56	9
Glass015vs2	**100.00**	**1**	96.77	4	80.65	6	54.84	8	**100.00**	**1**	**100.00**	**1**	90.32	5	16.13	10	29.03	9	64.52	7
Vowel0	96.96	5	99.75	2	96.96	5	99.24	4	**100.00**	**1**	99.49	3	94.94	7	64.30	9	61.77	10	81.52	8
Cleveland0vs4	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	65.63	10	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**
Yeast1vs7	**93.60**	**1**	72.00	6	70.40	7	76.00	4	91.20	2	78.40	3	68.00	8	21.60	10	24.00	9	72.80	5
Ecoli4	**94.44**	**1**	76.19	5	76.19	5	74.60	7	**94.44**	**1**	79.37	3	76.98	4	22.22	10	27.78	9	32.54	8
Shuttle2vs4	**93.36**	**1**	72.99	5	61.61	7	70.62	6	85.31	2	74.88	3	72.99	4	56.87	8	56.87	9	51.18	10
Yeast2vs8	**100.00**	**1**	80.30	7	80.30	7	75.76	10	98.48	2	83.33	5	90.91	4	83.33	5	78.79	9	93.18	3
Yeast4	**98.91**	**1**	95.65	4	95.65	4	93.48	7	**98.91**	**1**	95.65	4	92.39	8	88.04	9	86.96	10	97.83	3
Yeast1289vs7	**100.00**	**1**	91.30	8	94.57	7	72.83	9	97.83	5	95.65	6	**100.00**	**1**	**100.00**	**1**	71.74	10	98.91	4
Winequalityred8vs6	**100.00**	**1**	92.13	3	94.49	2	38.58	10	91.34	5	92.13	3	79.53	7	70.87	9	78.74	8	89.76	6
Yeast6	99.31	2	98.61	4	97.92	5	94.10	9	99.31	2	97.92	5	94.79	8	95.14	7	90.97	10	**99.65**	**1**
Winequalityred8vs67	**100.00**	**1**	93.41	5	94.61	4	38.92	10	93.41	6	96.41	3	78.44	7	73.05	8	67.07	9	97.01	2
Abalone19vs10111213	**100.00**	**1**	89.94	3	83.33	7	66.04	10	92.14	2	87.42	5	89.62	4	66.35	8	66.35	8	84.91	6
Winequalitywhite39vs5	**99.31**	**1**	92.78	5	90.72	6	11.68	10	93.47	3	97.59	2	89.00	7	81.44	8	80.76	9	93.13	4
Winequalityred3vs5	**100.00**	**1**	96.32	5	97.79	3	50.00	10	96.32	4	96.32	5	88.24	7	63.97	8	58.82	9	**100.00**	**1**
Abalone19	**100.00**	**1**	94.46	5	88.24	8	94.12	6	**100.00**	**1**	91.35	7	84.43	10	95.50	4	86.51	9	95.85	3
Average		**1.51***†		3.52*†		4.80*		6.29		2.23*†		3.09*†		5.74*		6.38†		6.69†		4.37

Table 4.5: G-mean results from Experiment II

Dataset	G-mean Value/Rank																			
	Baseline		SMOTE		BLSMOTE		kmUnder		SMOTE-ENN		SMTBagging		RUSBoost		OBU		AdaOBU		BoostOBU	
Glass1	49.82	9	58.49	8	73.37	2	72.75	4	35.10	10	58.73	7	71.48	5	71.35	6	73.25	3	**81.05**	**1**
Glass0	89.44	5	**94.87**	**1**	74.64	8	94.02	3	89.44	5	**94.87**	**1**	93.16	4	71.71	10	74.40	9	82.38	7
Vehicle1	**100.00**	**1**	98.43	4	**100.00**	**1**	96.82	5	95.20	7	94.87	8	**100.00**	**1**	96.82	5	94.87	8	73.10	10
Vehicle3	0.00	7	0.00	7	56.06	2	43.64	6	**56.90**	**1**	56.06	2	0.00	7	56.06	2	43.64	5	0.00	7
Vehicle0	76.98	2	69.92	8	75.03	3	69.92	7	68.31	9	**78.88**	**1**	71.66	6	50.92	10	72.01	4	72.01	5
Ecoli2	89.44	4	**100.00**	**1**	89.44	4	77.46	9	77.46	9	89.44	4	97.26	3	89.44	4	**100.00**	**1**	89.44	4
Segmemt0	43.30	10	53.75	5	49.30	8	61.24	3	47.14	9	59.16	4	52.70	6	**63.25**	**1**	63.03	2	52.04	7
Pageblocks0	89.71	8	93.49	4	93.55	2	93.54	3	**95.11**	**1**	93.24	5	92.22	7	92.57	6	47.97	10	60.46	9
Glass015vs2	0.00	9	56.80	3	51.85	5	**60.46**	**1**	0.00	9	57.74	2	54.87	4	32.79	8	43.99	7	46.37	6
Vowel0	93.02	6	98.32	3	93.02	6	**99.62**	**1**	99.23	2	97.42	4	95.16	5	80.19	9	78.60	10	90.29	8
Cleveland0vs4	70.71	2	0.00	6	0.00	6	**81.01**	**1**	0.00	6	0.00	6	70.71	2	70.71	2	70.71	2	0.00	6
Yeast1vs7	62.60	7	79.77	5	75.70	6	**85.13**	**1**	48.30	8	83.24	2	80.52	3	45.38	10	47.25	9	80.21	4
Ecoli4	59.98	6	73.77	5	76.19	3	**79.97**	**1**	44.99	9	74.03	4	77.77	2	43.64	10	50.13	8	52.07	7
Shuttle2vs4	62.00	10	73.55	2	69.69	6	71.19	4	66.45	9	73.31	3	**74.13**	**1**	68.44	7	69.89	5	66.75	8
Yeast2vs8	0.00	9	51.74	4	51.74	4	50.25	6	0.00	9	52.70	2	38.92	7	52.70	2	36.24	8	**55.73**	**1**
Yeast4	83.21	9	92.78	3	**97.80**	**1**	91.72	4	62.90	10	87.48	7	91.19	5	83.93	8	88.47	6	93.83	2
Yeast1289vs7	86.60	2	82.75	5	0.00	8	60.34	6	0.00	8	84.70	3	**100.00**	**1**	0.00	8	84.70	4	49.73	7
Winequalityred8vs6	0.00	10	55.42	8	**97.21**	**1**	62.11	7	78.03	3	55.42	8	89.18	2	68.73	6	72.45	5	77.36	4
Yeast6	78.78	9	99.30	2	92.56	6	84.01	8	70.46	10	98.95	3	97.36	4	84.47	7	**95.38**	**5**	**99.83**	**1**
Winequalityred8vs67	0.00	6	0.00	6	0.00	6	62.39	4	0.00	6	0.00	6	51.13	5	**85.47**	**1**	81.89	2	80.42	3
Abalone19vs10111213	0.00	10	38.72	5	37.27	9	46.92	3	39.19	4	38.17	7	38.65	6	**57.60**	**1**	**57.60**	**1**	37.62	8
Winequalitywhite39vs5	0.00	7	0.00	7	42.60	5	34.18	6	0.00	7	0.00	7	**73.08**	**1**	57.08	3	56.84	4	61.03	2
Winequalityred3vs5	0.00	10	69.40	5	69.93	3	**70.71**	**1**	69.40	4	69.40	5	66.42	7	56.56	8	54.23	9	**70.71**	**1**
Abalone19	84.52	8	89.98	5	86.97	7	81.99	9	65.47	10	88.49	6	91.89	2	90.48	4	**93.01**	**1**	90.64	3
Average		5.30		4.03		5.48		4.08		5.73		**3.97**		4.26		4.89		4.68		4.48

Table 4.6: F1-score results from Experiment II

Dataset	F1-score Value/Rank																			
	Baseline		SMOTE		BLSMOTE		kmUnder		SMOTE-ENN		SMTBagging		RUSBoost		OBU		AdaOBU		BoostOBU	
Glass1	36.36	2	26.09	7	31.25	3	29.41	4	18.18	10	27.27	5	26.32	6	20.69	8	20.59	9	**38.71**	**1**
Glass0	88.89	4	**94.74**	**1**	60.00	7	90.00	3	88.89	4	**94.74**	**1**	85.71	6	41.86	10	44.44	9	52.63	8
Vehicle1	**100.00**	**1**	95.24	4	**100.00**	**1**	90.91	8	86.96	9	94.74	5	**100.00**	**1**	90.91	7	94.74	6	56.25	10
Vehicle3	0.00	7	0.00	7	33.33	2	10.53	6	**40.00**	**1**	33.33	2	0.00	7	20.00	4	13.33	5	0.00	7
Vehicle0	71.43	2	62.07	8	68.57	5	62.86	7	60.00	9	**72.73**	**1**	66.67	6	60.00	9	69.77	3	69.77	4
Ecoli2	88.89	3	**100.00**	**1**	88.89	3	75.00	9	75.00	9	88.89	3	83.33	8	88.89	3	**100.00**	**1**	88.89	3
Segmemt0	31.58	9	37.21	6	32.56	8	45.00	3	29.41	10	42.86	5	36.36	7	**48.98**	**1**	47.83	2	44.07	4
Pageblocks0	82.73	3	76.01	6	70.90	8	78.46	4	83.53	2	76.69	5	75.19	7	**85.46**	**1**	22.70	10	26.27	9
Glass015vs2	0.00	9	40.00	2	20.00	5	21.05	4	0.00	9	**50.00**	**1**	28.57	3	12.90	8	14.81	6	13.33	7
Vowel0	85.93	5	97.67	3	85.93	5	97.74	2	**99.22**	**1**	96.12	4	84.35	7	47.97	9	46.26	10	64.04	8
Cleveland0vs4	**66.67**	**1**	0.00	6	0.00	6	26.67	5	0.00	6	0.00	6	**66.67**	**1**	**66.67**	**1**	**66.67**	**1**	0.00	6
Yeast1vs7	52.17	7	65.52	5	60.87	6	**71.93**	**1**	33.85	10	70.37	2	66.13	3	45.05	8	44.94	9	66.09	4
Ecoli4	49.23	6	58.82	5	61.54	3	**65.45**	**1**	31.03	10	59.79	4	63.46	2	40.91	9	44.44	7	43.21	8
Shuttle2vs4	52.24	10	61.46	2	57.51	6	58.65	4	55.00	9	61.31	3	**62.14**	**1**	56.91	7	58.63	5	56.49	8
Yeast2vs8	0.00	9	11.76	4	11.76	4	10.00	7	0.00	9	13.33	2	10.53	6	13.33	2	5.71	8	**23.53**	**1**
Yeast4	77.78	4	78.26	3	83.33	2	72.00	6	53.33	10	72.73	5	69.23	7	55.17	9	58.06	8	**85.71**	**1**
Yeast1289vs7	85.71	2	40.00	4	0.00	8	12.90	7	0.00	8	54.55	3	**100.00**	**1**	0.00	8	23.53	6	33.33	5
Winequalityred8vs6	0.00	10	14.29	5	**46.15**	**1**	7.14	9	25.00	2	14.29	5	18.75	4	9.52	8	12.50	7	22.22	3
Yeast6	66.67	4	80.00	2	66.67	4	38.71	9	57.14	6	72.73	3	51.61	7	42.86	8	38.10	10	**94.12**	**1**
Winequalityred8vs67	0.00	6	0.00	6	0.00	6	5.56	4	0.00	6	0.00	6	5.00	5	11.76	2	9.84	3	**40.00**	**1**
Abalone19vs10111213	0.00	10	5.13	4	3.33	9	3.45	8	**6.25**	**1**	4.26	6	5.00	5	5.17	2	5.17	2	3.64	7
Winequalitywhite39vs5	0.00	7	0.00	7	6.06	5	3.75	6	0.00	7	0.00	7	**15.00**	**1**	6.56	3	6.35	4	14.81	2
Winequalityred3vs5	0.00	10	25.00	3	33.33	2	5.56	7	25.00	3	25.00	3	10.53	6	3.85	8	3.39	9	**66.67**	**1**
Abalone19	**83.33**	**1**	41.38	5	25.53	9	34.48	6	60.00	2	31.58	7	23.73	10	46.15	4	26.42	8	48.00	3
Average		4.00*		3.55*		5.36		4.97		4.98		**3.48***		4.91		5.62†		5.7†		4.17

RUSBoost on both sensitivity and specificity proved that BoostOBU provided a better solution than these approaches. The method also showed higher average specificity than kmUnder while their average sensitivity ranks were comparable. Compared with OBU and AdaOBU, which had higher sensitivity, BoostOBU had significantly higher average specificity. In contrast, its specificity was lower than SMOTE, SMOTE-ENN and SMTBagging, which achieved lower sensitivity. However, BoostOBU won on 24 out of 66 datasets whereas each of SMOTE and SMTBagging only had 21 winning cases. These results only suggest different trade-offs between sensitivity and specificity of BoostOBU and these methods. Their g-mean and F1-score will be discussed for a more conclusive comparison.

Table 4.5 shows that AdaOBU and BoostOBU had higher average G-mean than the baseline, BLSMOTE, SMOTE-ENN and OBU. However, the statistical tests did not indicate significant differences between our methods and the others. Thus, it may be said that AdaOBU and BoostOBU had comparable G-mean to the other methods on average. However, it is worth pointing out that AdaOBU and BoostOBU achieved the highest G-mean on 18 and 20 datasets while SMTBagging and RUSBoost, which had higher average ranks, only won on 16 and 15 datasets.

In Table 4.6, BoostOBU showed significantly higher average rank on F1-score than OBU and AdaOBU. This suggests that BoostOBU provided a better trade-off between the accuracy of the two classes than OBU and AdaOBU. Even though BoostOBU had a lower average rank than SMTBagging and SMOTE, it far outnumbered the two methods in winning cases by 23 to 17 and 18, respectively. Extremely low F1-score can be observed in Table 4.6, especially on large and highly imbalanced datasets. In many cases, e.g. on Yeast6 and Abalone19, low F1-score is seen although high values in the other metrics were achieved. This is because F1-score factors in precision, which considers TP and FP. On a large and highly imbalanced scenario, the calculation of F1-score can be heavily dominated by high FP regardless of specificity. The 23 winning cases of BoostOBU were spread throughout all imbalance degrees. In particular, it handled extremely imbalanced datasets better than the other methods. This is evidence that BoostOBU performed the best among the methods in minimising information loss while maximising sensitivity.

Both AdaOBU and BoostOBU have shown their superior results over other well-established and state-of-the-art methods including ensemble-based methods in many cases. AdaOBU achieved the highest average sensitivity but suffered from high information loss in the negative class. BoostOBU, which often provided high sensitivity and most favourable trade-offs of relatively smaller FP, may be more preferred in many problem domains.

Table 4.7: Results on the large datasets from Experiment III

Dataset	Metric	Baseline	SMOTE	BLSMOTE	kmUnder	OBU	AdaOBU	BoostOBU
Breast Cancer	sensitivity	28.23	70.16	55.65	**94.35**	42.74	58.87	75.00
	specificity	99.96	97.50	97.08	66.34	**99.91**	78.37	78.83
	G-mean	53.12	**82.71**	73.50	79.12	65.35	67.92	76.89
	F1-score	41.92	24.17	17.56	3.30	**54.08**	3.18	4.11
MNIST_3	sensitivity	82.45	87.76	84.08	**95.92**	82.45	87.35	92.24
	specificity	99.80	99.82	**99.89**	36.56	99.80	97.75	99.13
	G-mean	90.71	93.60	91.64	59.22	90.71	92.40	**95.62**
	F1-score	86.14	**89.77**	88.98	6.43	86.14	61.06	80.00
MNIST_5	sensitivity	90.96	93.91	93.91	**95.94**	90.96	93.73	94.10
	specificity	99.61	99.61	**99.69**	91.81	99.61	85.95	95.38
	G-mean	95.18	96.72	**96.76**	93.85	95.18	89.75	94.74
	F1-score	91.47	93.05	**93.82**	53.17	91.47	39.32	65.55

For further evaluation, an additional experiment using J48, kNN and RF was carried out (Detailed results are available on *GitHub*[3]). Statistical analysis suggests that there were no significant differences in the results using SVM compared to J48 and RF. However, AdaOBU with kNN performed poorer than AdaOBU with SVM across all metrics. Our results also showed that BoostOBU with kNN achieved significantly higher sensitivity and lower performance in other metrics compared to SVM. These results are consistent with literature [135], which showed that SVM outperformed other algorithms in sensitivity when there were fewer negative instances in the overlapping region.

4.4.3 Experiment III: Large datasets

Table 4.7 shows the results on the three large and high-dimensional datasets. In all scenarios, AdaOBU obtained higher sensitivity than OBU, and BoostOBU further improved from AdaOBU. Results in other measures varied across datasets.

On the breast cancer dataset, AdaOBU and BoostOBU significantly improved sensitivity from the baseline, BLSMOTE, and OBU. They outperformed kmUnder in specificity, and outperformed the baseline and OBU in G-mean. BoostOBU also achieved higher G-mean than BLSMOTE. AdaOBU and BoostOBU suffered from high FP as can be seen from low F1-score. It is worth pointing out that none of the methods could yield high sensitivity without a high decrease in F1-score. This trade-off was likely caused by the issue of high class overlap. This is evidenced by the results of SMOTE, which showed significant improvement in sensitivity from 28.23% to 70.16% and slightly lower specificity from 99.96% to 97.5% compared to the baseline. However, F1-score of SMOTE was largely reduced from 41.92% to 24.17% due to the bias caused by relatively

[3] https://github.com/fonkafon/BoostedOBU/blob/master/Results.zip

large FP compared to the number of TP.

On MNIST_3, BoostOBU was among the methods that produced the most favorable results. BoostOBU showed good performance across all metrics. It achieved the second-highest sensitivity of 92.24%, high specificity of 99.13%, the highest G-mean of 95.62%, and relatively high F1-score of 80%. This was significantly higher than the overall performance of kmUnder, which produced the highest sensitivity but very low specificity, G-mean and F1-score. AdaOBU showed improvement over OBU in sensitivity and G-mean, however suffered from high FP. BoostOBU improved further from AdaOBU with higher sensitivity and a reduction in FP as reflected by high specificity and F1-score. Note that OBU with the fixed elimination threshold failed to undersample this dataset as well as MNIST_5.

On MNIST_5, AdaOBU and BoostOBU provided competitive sensitivity with SMOTE, BLSMOTE and kmUnder and outperformed the baseline and OBU. AdaOBU did not performed as well as the other methods in terms of specificity, G-mean and F1-score due to excessive elimination. Consequently, BoostOBU showed low F1-score. However, BoostOBU had reasonable specificity and G-mean, and produce higher specificity, G-mean and F1-score than kmUnder.

The proposed AdaOBU and BoostOBU performed relatively well on the large datasets in terms of sensitivity compared to other methods. Competitive results in specificity and G-mean were achieved in some cases. However, they often suffered from high FP partly due to the trade-off nature on a large and highly imbalanced datasets.

4.4.4 Discussion

Results on simulated and real-world datasets showed that our proposed methods often achieved high sensitivity. Compared to other existing methods, BoostOBU in particular provided higher sensitivity with better trade-offs of relatively smaller FP in most cases. The improvement in sensitivity of our methods is attributed to better visibility of the minority class near the borderline, which was obtained after removing majority class instances from the overlapping region. This allowed the learning algorithm to learn the maximum boundary of the minority class without interference of majority class instances. By oversampling borderline minority class instances in BoostOBU to enhance the performance of the clustering algorithm, the presence of the minority class near the boundary was also increased as an additional benefit. Higher sensitivity and better trade-offs of BoostOBU over other methods can be justified as follows. While BoostOBU attempted to maximise the presence of the minority class near the borderline, SMOTE and k-means undersampling only aimed to rebalance the class distribution. The

Table 4.8: Comparative results with evolutionary and deep learning-based methods

Dataset	AdaOBU/BoostOBU				EVINCI	CnGRSOMO*	CnGRSOMU*			
	Sensitivity	Specificity	G-mean	F1-score	G-mean	F1-score	Sensitivity	Specificity	G-mean	F1-score
Ecoli2	-	-	**100.00**	-	86.20	-	-	-	-	-
Winequalitywhite3vs7	-	-	**84.36**	-	66.11	-	-	-	-	-
Ecoli1	-	-	-	20.00	-	**78.10**	-	-	-	-
Ecoli3	-	-	-	**66.67**	-	64.50	-	-	-	-
Abalone0918	**97.87**	**92.05**	**94.91**	**92.00**	-	-	83.50	85.48	84.48	39.00
Yeast4	**90.00**	**97.83**	**93.83**	**85.71**	-	-	75.00	85.19	79.93	25.00

*Estimated results from graphs.

improvement in the presence of the minority class in the overlapping region by SMOTE and k-means undersampling was limited by the imbalance degree. Also, as opposed to k-means undersampling, BoostOBU was unlikely to remove instances outside of the overlapping region causing smaller unnecessary information loss. Lastly, BLSMOTE only dealt with borderline instances whereas BoostOBU addressed the entire overlapping region. These enabled BoostOBU to achieve higher sensitivity and higher F1-score. This higher F1 score can be attributed to a higher increase in TP in relation to a smaller FP. This indicates a good trade-off of the method.

4.5 Performance Comparison with Evolutionary and Deep Learning-Based Methods

In this section, the performance of AdaOBU and BoostOBU with SVM classifiers is compared with evolutionary and deep learning-based methods, namely EVINCI [81], CnGRSOMO [62], CnGRSOMU [62] and SMOTE-CSELM [63]. Comparative results, which are based on availability of the results in the literature, are presented in Table 4.8 and Table 4.9. In the column named AdaOBU/BoostOBU, results from the better performing method between AdaOBU and BoostOBU are displayed. Note that if there are more than one available measure for the dataset, the better performing method was selected primarily based on sensitivity. The higher value between AdaOBU/BoostOBU and the compared method is highlighted in bold.

Table 4.8 shows that AdaOBU/BoostOBU clearly outperformed EVINCI and CnGRSOMU on the reported metrics. It is worth pointing out that since CnGRSOMU was shown to provide higher results than CnGRSOMO on Abalone0918 and Yeast4 [62], it can be said that AdaOBU/BoostOBU also outperformed CnGRSOMO on these datasets. On Ecoli1, AdaOBU/BoostOBU had significantly lower F1-score than CnRSOMO, but they were comparable on Ecoli3. As can be seen in Table 4.9, AdaOBU/BoostOBU provided competitive results with SMOTE-CSELM. Each of the methods achieved the highest sensitivity on 16 out of 26 datasets and the highest G-mean on 19 out of 35

Table 4.9: Comparative results with a deep learning-based method

Dataset	AdaOBU/BoostOBU		SMOTE-CSELM	
	Sensitivity	G-mean	Sensitivity	G-mean
Glass1	87.50	73.25	**95.21**	**78.66**
Wisconsin	**100.00**	**100.00**	98.74	97.99
Ecoli01vs5	-	70.71	-	**95.55**
Pima	**100.00**	**100.00**	80.96	76.65
Glass0	**100.00**	**82.38**	**100.00**	82.01
Haberman	**100.00**	**100.00**	77.72	65.92
Vehicle2	90.00	62.68	**100.00**	**99.29**
Vehicle1	**100.00**	**94.87**	98.00	86.17
Glass0123vs456	**100.00**	**100.00**	96.00	96.02
Vehicle0	**100.00**	72.01	**100.00**	**99.46**
Newthyroid1	100.00	**100.00**	100.00	99.16
Newthyroid2	-	0.00	-	**99.16**
Ecoli2	68.75	**100.00**	**100.00**	93.64
Segment0	**100.00**	63.03	**100.00**	**99.67**
Glass6	**100.00**	**100.00**	**100.00**	95.79
Yeast3	**100.00**	**100.00**	**100.00**	93.54
Ecoli3	**100.00**	89.75	94.29	**91.52**
Page-blocks0	-	60.46	-	**93.97**
Yeast2vs4	84.91	**100.00**	**100.00**	94.73
glass-0-1-5_vs_2	-	46.37	-	**84.75**
Yeast05679vs4	**100.00**	52.12	86.00	**83.18**
Vowel0	-	90.29	-	**100.00**
Glass2	0.00	0.00	**100.00**	**86.87**
Shuttle0vs4	-	**100.00**	-	**100.00**
Yeast1vs7	88.37	**80.21**	**100.00**	79.58
Glass4	**100.00**	65.73	**100.00**	**98.22**
Ecoli4	90.48	50.13	**100.00**	**98.40**
Abalone0918	**97.87**	**94.91**	93.06	90.61
Shuttle2vs4	-	69.89	-	**100.00**
Yeast1458vs7	-	**88.78**	-	68.80
Yeast2vs8	33.33	55.73	**100.00**	**80.12**
Yeast1289vs7	-	**84.70**	-	74.57
Ecoli0137vs26	60.00	71.98	**100.00**	**79.06**
Yeast6	**100.00**	**99.83**	**100.00**	89.54
Abalone19	**100.00**	**93.01**	96.00	79.51

datasets. These results suggest that AdaOBU/BoostOBU provided competitive results with these EA and deep learning-based methods. Moreover, AdaOBU and BoostOBU with running time of $O(N)$ and $O(N^2)$, respectively, have lower time complexities than SMOTE-CSELM, which requires $O(N^3)$ [63]. This will make AdaOBU and BoostOBU more preferable than SMOTE-CSELM, especially on large datasets.

4.6 Conclusions

In this chapter, new overlap-based undersampling methods extended from OBU were presented. By removing negative instances from the overlapping region based on an adaptive threshold, exceptional improvement in the minority class accuracy with a relatively small trade-off of FP was achieved. The methods proved to enhance the classification of well-known imbalanced datasets and showed significant improvements over OBU across different scenarios. Furthermore, they outperformed other existing methods across a wide range of simulated and real-world datasets of varying class imbalance and class overlap degrees. These results can be attributed to several advantages of the methods over other common undersampling techniques. First, the adaptive elimination threshold enables the amount of undersampling to be proportional to the overlap degree. This also results in minimising the excessive elimination of negative instances, which reduces information loss. Second, enhancing the presence of the positive instance class near the borderline areas showed to be beneficial to the overall performance of the method.

Future work will address limitations of the methods. These may include the dependencies on the techniques used such as BLSMOTE and the distance-based algorithms. Results showed that the performance of BoostOBU could be highly dependant on how BLSMOTE performs, thus other oversampling methods that provide better results may be explored. Moreover, the performance of the methods were more consistent on the simulated datasets than on the real-world datasets. This can be partly due to the difference in data uniformity. Thus, another potential future direction is to also factor in other information such as class density and local data density, which could be obtained using the techniques proposed in [136]. The problem of small disjuncts in the minority class could be tackled by an adaptive selection on the number of clusters. Finally, as a distance-based clustering algorithm is used in the proposed methods, the well-known curse of dimensionality could have affected the results. This issue may be addressed by using other improved soft clustering algorithms that showed less dependency on similarity measure [137, 138]. Alternatively, projecting data onto a lower-dimensional space using a technique such as Principle Components Analysis may be considered.

Chapter 5

Neighbourhood-Based Undersampling

This chapter presents an alternative overlap-based approach to handle imbalanced problems. Four new undersampling methods based on neighbourhood searching are proposed. Unlike in the OBU-based methods, where global searching is used, these methods employ a local searching algorithm aiming at more accurate identification and elimination of overlapped majority class instances. Extensive experiments using simulated and real-world datasets were carried out. Results showed higher performance than state-of-the-art methods across different common metrics with exceptional and statistically significant improvements in sensitivity. This work was published in the journal of Information Sciences [1].

5.1 Background

Neighbourhood searching has long been used in class overlap-based methods to discover potential borderline and overlapped instances. Among many neighbourhood-based techniques, kNN is one of the most widely-used algorithms. In [74], Adaptive Synthetic sampling (ADASYN), which employed kNN, was presented. The number of new minority class instances created from each original instance was proportional to the amount of majority class neighbours surrounded. By doing so, more minority class instances were introduced into the overlapping and borderline region. Results showed improvement in 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 kNN-based methods that focused on instances near the decision boundary are such as Edited Nearest Neighbour (ENN) [105] and Neighbourhood Cleaning Rule (NCL) [75]. ENN selectively removes majority class instances by considering its k nearest neighbours that belong to the other class, where $k = 3$. It has to be noted that the setting of the k value in this approach significantly impacts the performance. That is, for example, a small k value can leave a lot of the overlapped majority class instances unremoved. NCL is an extension of ENN, where the k nearest neighbours of both minority class and majority class instances were considered in removing majority class instances. Results showed an improvement of NCL over a data-distribution based method proposed in [139]; however, it was outperformed by later overlap-based methods such as CCR [140] and evolutionary undersampling [141].

BLSMOTE [59] synthesises instances from borderline minority class instances and their nearest neighbours. Two techniques of the method were proposed, BLSMOTE1 and BLSMOTE2. BLSMOTE1 considers only the minority-class nearest neighbours while BLSMOTE2 includes the nearest neighbours of both classes in generating new instances. Results showed that BLSMOTE2 whose synthetic instances were generated closer to the borderline achieved higher TPR.

An undersampling method based on Tomek Link [117] was proposed in [43]. Redundant negative instances with the lowest contributions to classification were selectively removed. Similar to most of the aforementioned methods, the undersampling rate was limited by class imbalance. That is the method was applied until the balanced class distribution was achieved. This could lead to insufficient elimination when the imbalance degree is low. On the other hand, at a high imbalance degree, excessive elimination of majority class instances may occur.

In this chapter, a neighbourhood-based undersampling framework for identifying and eliminating overlapped negative instances is presented. The main contributions of this work are outlined as follows:

- Four novel kNN-based undersampling methods designed to accurately detect and optimally remove potential overlapped majority class instances are presented. Different criteria to identify overlapped instances for removal are introduced. These methods are different from existing variations of kNN in the following 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.

- Extensive experiments using extremely imbalanced and overlapped simulated and real-world datasets were carried out. Our methods proved to be capable of handling any degree of class overlap as can be seen in the results and discussion section.

- 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.

5.2 The Neighbourhood-Based Methods

The details of four neighbourhood-based (NB-based) undersampling methods are provided in this section[1]. Their common objective is to maximise the visibility of minority class instances in the overlapping region while minimising excessive elimination. The four methods are Basic Neighbourhood Search (NB-Basic), Modified Tomek Link Search (NB-Tomek), Common Nearest Neighbours Search (NB-Comm), and Recursive Search (NB-Rec). These methods vary in terms of local search and elimination criteria. NB-Basic is the first and simplest one among the methods. It is designed to remove majority class instances from the overlapping region without compromising any minority class instances. NB-Basic showed 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. NB-Tomek and NB-Comm were created to address the potential excessive elimination of majority class instances. NB-Comm was then extended to NB-Rec aiming at improving the detection of overlapped instances.

5.2.1 Basic Neighbourhood Search

NB-Basic was implemented as in Algorithm 4. The method removes any negative query that has a positive neighbour.

As can be seen in Fig. 5.1(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

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

Algorithm 4: 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
            $X \leftarrow X \cup \{x\}$;
        end
    end
    $\hat{T} \leftarrow T - X$;
    return ($\hat{T}$)
end
```

information is considerably more valuable and losing part of it is highly undesirable in some application domains.

5.2.2 Modified Tomek Link Search

Modified Tomek Link Search is proposed as an extension of NB-Basic to address potential excessive elimination of negative instances. As described in Algorithm 5, for every negative instance x with a positive neighbour y, x is removed only if it is one of the k nearest neighbours of y. In other words, when the neighbourhood between a negative query and a positive query is established in both directions, the negative query in the modified Tomek Link is eliminated (Fig. 5.1(b)).

The rationale for considering this second query is illustrated in Fig. 5.2, which shows that if q is within the k nearest neighbours of p, it does not necessarily imply that p is also within the k nearest neighbours of q.

5.2.3 Common Nearest Neighbours Search

It was observed that when a majority class query was used, there was a higher probability that NB-Tomek would miss nearby positive instances. Therefore, in this variation, we propose an alternative method, NB-Comm, to remove common negative neighbours of positive instances. As defined in Algorithm 6, two positive queries will be used

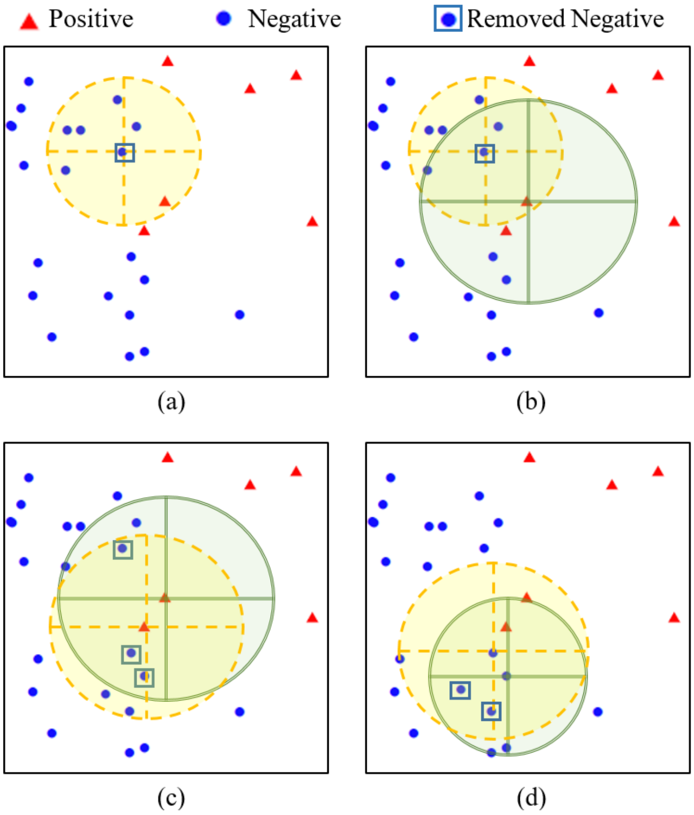


Figure 5.1: The proposed neighbourhood-based undersampling methods (a) NB-Basic (b) NB-Tomek (c) NB-Comm (d) NB-Rec

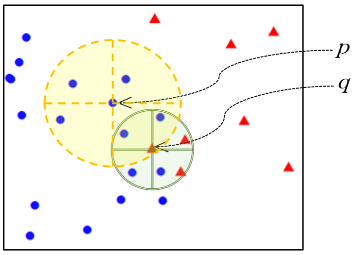


Figure 5.2: Neighbourhood is *not* established in both directions

Algorithm 5: 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
                    $X \leftarrow X \cup \{x\}$;
                end
            end
        end
    end
    $\hat{T} \leftarrow T - X$;
    return $(\hat{T})$
end
```

for considering the elimination of a negative instance. The common negative nearest neighbours of any two positive queries are identified as potential overlapped instances and removed (Fig.5.1(c)).

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

Algorithm 6: 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
            $A_y.freq \leftarrow A_y.freq + 1$;
        end
    end
    foreach $x \in A.instance$ do
        if $A_x.freq > 1$ then
            $X \leftarrow X \cup \{x\}$;
        end
    end
    $\hat{T} \leftarrow T - X$;
    return $(\hat{T})$
end
```

5.2.4 Recursive Search

NB-Rec is proposed as an extension of NB-Comm to ensure sufficient and accurate elimination of overlapped negative instances. From Algorithm 6, 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 7. 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 as depicted in Fig.5.1(d). We hypothesise that by introducing this extension, a finer-grained search criteria is provided. As a result, more overlapped negative instances will be detected and further improvement in sensitivity will be achieved.

Algorithm 7: Recursive Search Undersampling

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

5.2.5 Time Complexity Analysis

The time complexities of all NB-based methods are $O(N^2)$, which is mainly the cost of the kNN algorithm. Detailed analysis of the running time of each method is as follows. Note that in the NB-based methods, the nearest neighbour search is not necessarily applied to all instances in the datasets. However, for simplicity, it is assumed that at the beginning of each method, kNN is applied on the whole dataset, which requires $O(N^2)$. Some coefficients such as data dimension have also been dropped. In NB-Basic, after kNN is applied, checking whether each of the negative queries has any positive nearest neighbours takes additional $O(n)$, where n is the number of negative instances. Thus, the time complexity of NB-Basic is $O(N^2)$. In NB-Tomek, for each negative query, the k neighbours are checked for their k nearest neighbours. This process requires $O(nk^2)$. When a large value of k is used, e.g. $k = \sqrt{N}$, and $n \approx N$, the running time will be $O(N^2)$. Thus, the time complexity of NB-Tomek is $O(N^2)$. NB-Comm requires $O(n)$ to discover negative instances that have been found as a common nearest neighbour of a pair of positive queries. We will call this *operation-A* for the ease of later explanation of the analysis of NB-Rec. This makes $O(N^2)$, which requires by kNN, the main cost of NB-Comm. In NB-Rec, it takes $O(n(n-1))$ to discover negative instances that are common nearest neighbours of those negative instances found in *operation-A*. Thus, the time complexity of NB-Rec is $O(N^2)$.

The NB-based methods have comparable time complexities to many state-of-the-art and well-known resampling methods discussed in Chapter 3 and Chapter 4 such as methods in the SMOTE family. They will also be comparable to other methods that employ the kNN algorithm.

5.3 Experiments

The NB-based methods were evaluated on both simulated and real-world datasets. The 66 simulated datasets used in Chapter 4 covering a wide range of scenarios including extremely imbalanced and overlapped datasets were used. Another extensive experiment on 24 public real-world datasets was carried out for further evaluation. Moreover, 2 large high-dimensional datasets were used in the final experiment to verify the consistency in the performance.

5.3.1 Setup

Three sets of experiments were carried out. In Experiment I, simulated datasets were used, and in Experiment II, small to medium-sized real-world datasets were used for evaluation. In Experiment III, further evaluation was carried out using large real-world datasets with high dimensions. The datasets used in Experiment II and III also included multi-class problems. To straightforwardly apply the 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 [142] and has been shown to have good performance [143].

SVM and RF were chosen to be the learning algorithms as they are considered ones of the most-used learning methods in imbalanced classification [11]. Sensitivity, G-mean, precision, and F1-score were used for evaluation of the methods. These are common metrics for imbalanced learning [34, 35, 37, 55, 144]. This selection of classification algorithms and evaluation metrics allows the reader to compare our results with a wide range of methods in the literature.

Experimental results were compared with state-of-the-art and well-established methods for handling imbalanced datasets. These included class distribution-based methods namely, SMOTE [12] and k-means undersampling (kmUnder) [14], and class-overlap based methods including OBU [18], BLSMOTE [59] and ENN [105].

5.3.2 Datasets

In Experiment I, 66 uniformly-distributed binary-class datasets were simulated. These datasets capture wide ranges of class-overlap and imbalance degrees. The class imbalance degrees used were $1.5, 3, 12, 30, 60, 120$. 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.

Table 5.1 shows the public datasets that were used in Experiment II. These datasets were obtained from UCI Repository [129] and KEEL Repository [130]. The datasets vary in terms of imbalance degrees (*1.86-41.4*), number of features (*5-18*), and number of instances (*214-5,472*).

In Experiment III, we used the breast cancer dataset from *KDD Cup 2008*[2] and the handwritten digits dataset from the MNIST database [133]. The breast cancer dataset is 117-feature, binary-class and contains 102,294 samples with 101,671 negative and 623 positive samples, which makes $imb = 163.20$. The handwritten digits dataset is 10-class with 784 features, and contains 60,000 samples. Class 3 and class 5 were selected as the minority class to make two new datasets, MNIST_3 and MNIST_5. The minority class of the two datasets was undersampled to obtained a higher class imbalance degree. In MNIST_3, class 3 was undersampled such that $imb = 43.90$, which consists of 53,869 negative and 1,227 positive instances. In MNIST_5, class 5 was undersampled such that $imb = 20.13$, which consists of 53,869 negative and 2,711 positive instances.

In all experiments, each dataset was partitioned into 80 : 20 of training and testing sets. The testing data was only used during model evaluation for the result report. In Experiment I and II, 10-fold cross-validation was used in the training phase for the purpose of automatic parameter tuning of the classification model. Follow the methods

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

Table 5.1: 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	2.90	18
5	Vehicle0	846	199	3.25	18
6	Ecoli1	336	77	3.36	7
7	New-thyroid1	215	35	5.14	5
8	New-thyroid2	215	35	5.14	5
9	Ecoli2	336	52	5.46	7
10	Segmemt0	2308	329	6.02	19
11	Yeast3	1484	163	8.10	8
12	Ecoli3	336	35	8.60	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	14.30	7
17	Glass4	214	13	15.46	9
18	Ecoli4	336	20	15.80	7
19	Page-blocks13vs2	472	28	15.86	10
20	Abalone09-18	731	42	16.40	8
21	Glass5	214	9	22.78	9
22	Yeast4	1484	51	28.10	8
23	Ecoli0137vs26	281	7	39.14	7
24	Yeast6	1484	35	41.40	8

available in the *caret* package in R, *cost* (C) of SVM and *mtry* of RF were tuned to obtain the best models based on the overall accuracy. No cross-validation was applied to the large datasets in Experiment III as sufficient data was available.

5.3.3 Parameter Settings

For a fair comparison among the methods, no parameter tuning was performed for the resampling methods. For NB-based methods, k is an important parameter, where kNN is used to investigate the surroundings of instances. A simple rule of thumb, where k is set to equal the square root of the dataset size [145, 146], was considered. Furthermore, to take into account the class imbalance issue and promote the discovery of overlapped majority class instances, we adjusted the k value to also be proportional to the imbalance degree as can be seen in Eq. 5.1, where N is the number of instances in the dataset.

$$k = \sqrt{N} + \sqrt{imb} \tag{5.1}$$

SMOTE, in contrast, requires a small k value to ensure better distribution of synthesised instances. In experiment I, k in SMOTE was set to equal 5, following the original work [12]. However, in Experiment II, one of the real-world datasets used 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 [105], kmUnder [14], and OBU [18], the same parameter settings as stated in the original work were used.

The Radial Basis Function kernel was used for SVMs with the default setting in *caret* package in R of $\gamma = \frac{1}{f}$, where f is the number of features in the dataset. In RF, *mtree* was set to 500.

5.4 Results and Discussion

5.4.1 Experiment I: Simulations

The main objective of this experiment is to assess the impact of class imbalance and class overlap on the NB-based methods' performance across a wide range of degrees. Overall

Table 5.2: Average classification results from Experiment I

	NB-Basic	NB-Tomek	NB-Comm	NB-Rec	SMOTE	BLSMOTE	ENN	kmUnder	OBU	Baseline
sensitivity	99.86	99.59	99.64	**99.95**	98.11	98.56	75.52	98.35	97.46	67.75
G-mean	92.93	93.09	93.19	92.18	93.87	**93.78**	82.53	91.71	90.43	77.86
precision	58.83	59.67	60.12	54.59	64.21	62.83	72.80	55.30	43.00	**74.04**
F1-score	73.05	73.66	74.03	69.65	**76.41**	75.59	73.59	68.54	57.57	69.88

performance is discussed and compared against other existing methods. An experiment on 66 simulated datasets showed superior performance of the NB-based methods across different metrics. In particular, they yielded highest sensitivity, all of which were nearly 100%, while achieving competitive G-mean. These results were relatively stable across all datasets regardless of class imbalance and class overlap degrees. This is clearly illustrated in Fig. 5.3, where results of the NB-based methods are presented with solid lines, the results of the other methods are marked with dashed lines, and the shaded areas are the areas under the performance curves of the baseline (SVM).

The average performance of the methods are provided in Table 5.2, where the best result in each metric is presented in **bold**. Among the NB-based methods, NB-Rec showed the highest sensitivity of 99.95% and competitive G-mean, but was the least tolerable to information loss, resulting in the lowest precision and F1-scores. NB-Comm showed slightly better overall results than NB-Basic and NB-Tomek. A detailed discussion of these results is provided in the following subsections. Numerical results of this experiment are provided in the supplementary material[3].

NB-based methods vs class-distribution based methods

As can be see in Fig. 5.3 that NB-based methods achieved superior performance in sensitivity compared to the class-distribution based methods, namely SMOTE [12][4] and kmUnder [14]. This is evidence that the NB-based methods was better in promoting the visibility of the positive class across different class imbalance and class overlap degrees. Moreover, while the NB-based methods provided relatively stable sensitivity under different scenarios, sensitivity of the other methods tended to drop when the class overlap degree increased.

Table 5.2 shows that the NB-based methods not only produced the highest sensitivity but also showed competitive G-mean with SMOTE, and produced higher G-mean than kmUnder on average. The improvements in both G-mean and sensitivity indicate that our methods had improved the trade-offs between sensitivity and specificity, which

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

[4] In Fig. 5.3, SMOTE has similar performance in sensitivity to kmUnder (hence the line is not visible)

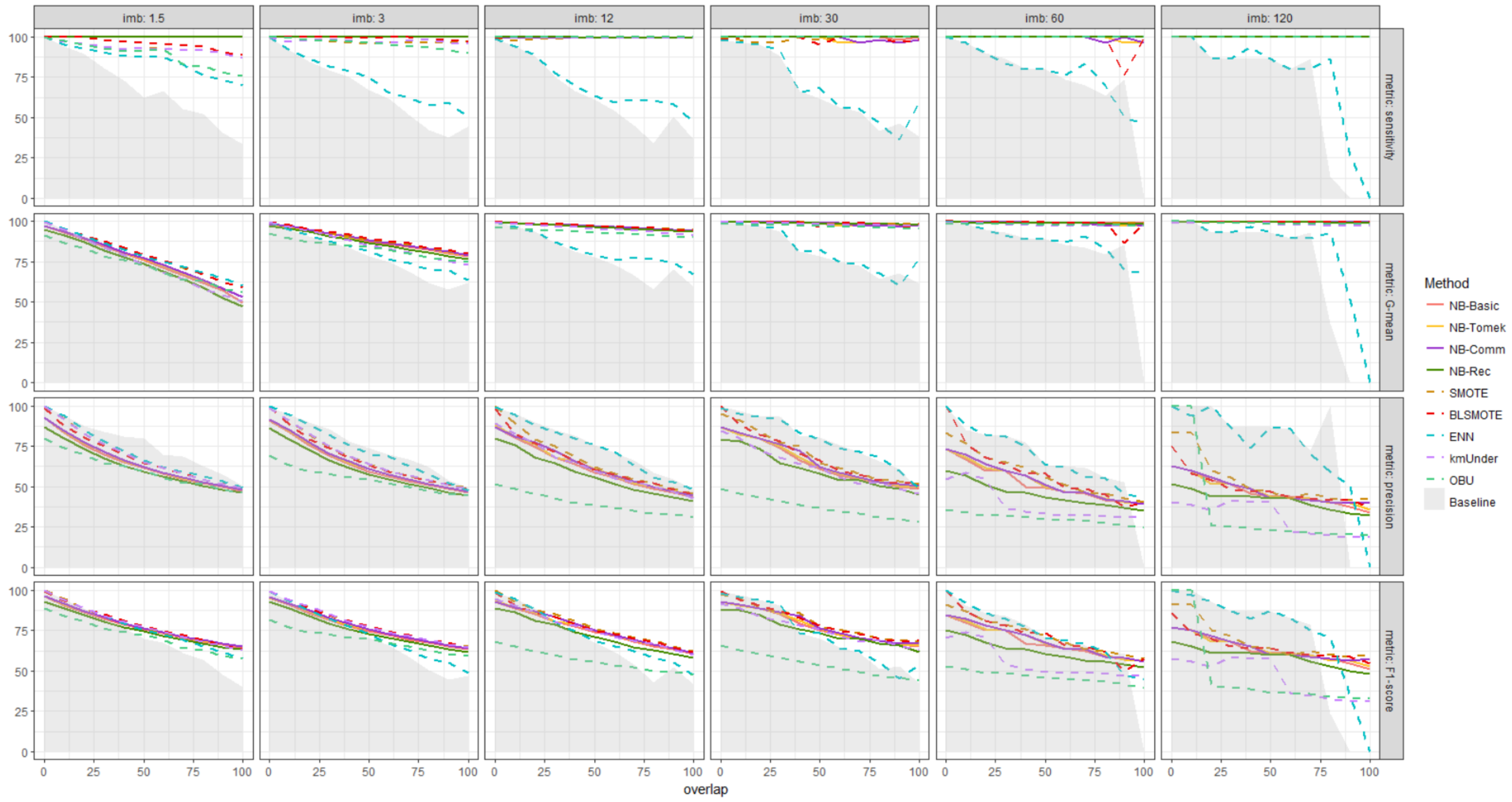


Figure 5.3: Performance of methods across different imbalance and overlap degrees

means that we have reduced both FP and FN, over state-of-the-art kmUnder across different ranges of class imbalance and class overlap degrees. It was observed that on low imbalanced datasets ($imb = 1.5$ and 3), the NB-based methods had lower precision compared to SMOTE and kmUnder; however, competitive F1-score was obtained. For datasets with higher degrees of class imbalance, our methods showed more favourable results over kmUnder in both precision and F1-score. Thus, it can be said that our methods had better performance as the degrees of class imbalance and class overlap increased. As for moderate to extreme imbalanced datasets (i.e. *imb = 12 to imb = 120*), the NB-based methods achieved comparable precision and F1-score with SMOTE in almost all datasets. Even so, it worth pointing out that our methods resulted in smaller training data than SMOTE, which could potentially reduce training time, especially in the case of large datasets.

NB-based methods VS class-overlap based methods

Fig. 5.3 shows that the NB-based methods achieved more favourable performance over other common and recent overlap-based techniques, which are BLSMOTE [59], ENN [105], and OBU [18]. All NB-based methods have competitive results in sensitivity and G-mean with OBU, but with higher precision and F1-score obtained. The improvements in precision and F1-score of our methods over OBU were clearly substantial, especially when the degrees of class imbalance and class overlap were higher. It suggests that our methods had relatively reduced both FP and FN by having more accurate detection of potential overlapped negative instances and minimisation of information loss over OBU. Table 5.2 shows that our NB-based methods provided comparable G-mean with BLSMOTE. Comparable precision and F1-score were also obtained in some cases. However, the NB-based methods showed more stable sensitivity than BLSMOTE throughout all class imbalance and class overlap degrees. This suggests that our NB-based methods had improved the positive class accuracy without sacrificing the performance in other metrics. In other words, by using our methods, lower FN could be achieved without increasing the amount of FP. Lastly, it can be said that our methods had clearly better results than ENN in all scenarios whereas ENN barely improved the performance from the baseline.

Overall results

The NB-based methods produced exceptionally high sensitivity. The highest average sensitivity of 99.95% was achieved by NB-Rec. Such high sensitivity is favourable across different imbalanced problems, especially in the medical domain. Comparable results in

G-means with SMOTE and BLSMOTE were obtained, but with lower precision and recall. It was also observed that NB-Rec produced the lowest precision and recall when compared with the other proposed NB-based methods. This suggests that maximising the visibility of positive instances may come at a high cost of FP. The NB-based methods clearly outperformed ENN in sensitivity and G-mean with comparable F1-score. More importantly, our methods outperformed state-of-the-art kmUnder and OBU in all measures, except for precision of NB-Rec that was competitive with kmUnder.

5.4.2 Experiment II: Real-world datasets

In this experiment, our methods were evaluated on real-world datasets. Tables 5.3 to 5.6 show performance of our methods against other methods on the UCI datasets using SVM, where the datasets are sorted by imbalance ratio from low to high. These tables also show the methods' ranks and average ranking based on their performance, where rank 1 means top performance and so on. Wilcoxon Signed Rank Tests were carried out to assess statistical significance of the difference in performance between the NB-based methods and other methods. Results are presented in Table 5.7, and the p values indicating a statistically significant difference between two methods at the significance level of 0.05 are highlighted in **bold**.

As can be seen in Table 5.3, the superior performance in sensitivity over other methods was achieved by the NB-based methods. This is consistent with the results obtained in Experiment I. Among the NB-based methods, NB-Rec ranked top on average sensitivity, followed by NB-basic. NB-Comm and NB-Tomek had competive ranking with OBU, and higher ranking than kmUnder, SMOTE, BLSMOTE, and ENN. Table 5.7 shows that the improvement in sensitivity achieved by NB-based methods over SMOTE, BLSMOTE and ENN was statically significant. Interestingly, both SMOTE and BLSMOTE did not improve the sensitivity and performed worse than the baseline in some cases.

The highest average ranking in G-mean was provided by NB-Comm, and the result in Table 5.7 proves that it was significantly better than BLSMOTE. The other NB-based methods had higher G-mean than SMOTE and BLSMOTE, and showed comparable G-mean with ENN, kmUnder and OBU. This is also consistent with the results on synthetic datasets.

SMOTE, BLSMOTE, and ENN outperformed our methods in precision (Table 5.5) but with significantly lower sensitivity values. Such a trade-off is not generally desirable in some specific imbalanced domains. In contrast, all our methods, outperformed sate-of-the-art kmUnder in both sensitivity and precision. Similarly, our methods outperformed OBU in precision with comparable results in sensitivity and G-mean.

Table 5.3: 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	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	97.87	6	97.87	6	97.87	6	97.87	6	**100.00**	**1**	97.87	6
Pima	98.11	2	96.23	3	92.45	5	**100.00**	**1**	43.40	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**	50.00	8	42.86	9	57.14	7	78.57	5	**85.71**	**1**	42.86	9
Vehicle1	**100.00**	**1**	**100.00**	**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	**100.00**	**1**	94.87	4	**100.00**	**1**	94.87	4	71.79	10	82.05	6	82.05	6	76.92	9	97.44	3	82.05	6
Ecoli1	**100.00**	**1**	93.33	3	93.33	3	86.67	5	53.33	10	66.67	7	66.67	7	80.00	6	**100.00**	**1**	60.00	9
New-thyroid1	71.43	5	71.43	5	**100.00**	**1**	**100.00**	**1**	57.14	7	57.14	7	57.14	7	**100.00**	**1**	**100.00**	**1**	57.14	7
New-thyroid2	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	85.71	10	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**
Ecoli2	**90.00**	**1**	**90.00**	**1**	**90.00**	**1**	**90.00**	**1**	70.00	10	80.00	8	**90.00**	**1**	**90.00**	**1**	**90.00**	**1**	80.00	8
Segmemt0	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	95.38	8	98.46	6	95.38	8	96.92	7	**100.00**	**1**	95.38	8
Yeast3	96.88	4	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	46.88	10	50.00	8	56.25	7	87.50	5	65.63	6	50.00	8
Ecoli3	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	28.57	8	14.29	9	57.14	7	**100.00**	**1**	**100.00**	**1**	14.29	9
Yeast2vs4	**80.00**	**1**	70.00	3	70.00	3	60.00	6	40.00	8	30.00	10	50.00	7	**80.00**	**1**	70.00	3	40.00	8
Vowel0	88.89	4	88.89	4	**100.00**	**1**	**100.00**	**1**	72.22	10	88.89	4	88.89	4	88.89	4	**100.00**	**1**	88.89	4
Glass2	**66.67**	**1**	**66.67**	**1**	**66.67**	**1**	**66.67**	**1**	33.33	7	33.33	7	0.00	9	**66.67**	**1**	**66.67**	**1**	0.00	9
Yeast1vs7	50.00	2	16.67	6	33.33	4	**83.33**	**1**	0.00	7	0.00	7	0.00	7	50.00	2	33.33	4	0.00	7
Glass4	50.00	3	50.00	3	50.00	3	**100.00**	**1**	50.00	3	50.00	3	50.00	3	**100.00**	**1**	50.00	3	50.00	3
Ecoli4	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	75.00	10	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**
Page-blocks13vs2	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	80.00	10	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**
Abalone09-18	50.00	4	37.50	6	50.00	4	**75.00**	**1**	12.50	7	0.00	8	0.00	8	62.50	3	**75.00**	**1**	0.00	8
Glass5	0.00	-	0.00	-	0.00	-	0.00	-	0.00	-	0.00	-	0.00	-	0.00	-	0.00	-	0.00	-
Yeast4	**80.00**	**1**	**80.00**	**1**	**80.00**	**1**	**80.00**	**1**	30.00	7	30.00	7	0.00	10	60.00	5	50.00	6	10.00	9
Ecoli0137vs26	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**
Yeast6	**85.71**	**1**	**85.71**	**1**	57.14	5	**85.71**	**1**	57.14	5	14.29	9	28.57	8	57.14	5	**85.71**	**1**	14.29	9
Average		1.74		2.22		2.17		**1.61**		6.74		7.30		5.65		3.43		2.09		6.39

Table 5.4: 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	**97.12**	**1**	**97.12**	**1**	**97.12**	**1**	95.35	9	96.66	4	96.66	4	96.66	4	96.66	4	0.00	10	96.66	4
Pima	47.50	8	52.83	7	55.24	6	0.00	10	60.02	5	61.43	3	64.60	2	**67.80**	**1**	29.43	9	61.43	3
Glass0	67.76	7	69.99	5	73.19	3	72.14	4	68.14	6	64.29	9	74.23	2	**80.34**	**1**	67.76	7	64.29	9
Vehicle1	40.00	9	46.48	8	55.92	2	51.67	4	48.93	7	51.54	5	55.91	3	**70.40**	**1**	38.02	10	51.54	5
Vehicle0	85.82	7	**89.94**	**1**	87.16	6	87.88	5	84.07	9	88.81	2	88.81	2	84.59	8	80.60	10	88.09	4
Ecoli1	91.82	2	88.71	3	87.67	4	82.45	6	70.11	10	80.85	7	80.85	7	84.95	5	**93.93**	**1**	77.46	9
New-thyroid1	82.13	5	82.13	5	95.74	2	94.28	3	75.59	7	75.59	7	75.59	7	**100.00**	**1**	89.75	4	75.59	7
New-thyroid2	95.74	4	95.74	4	95.74	4	88.19	10	**100.00**	**1**	92.58	9	98.60	2	95.74	4	92.80	8	98.60	2
Ecoli2	94.02	2	94.02	2	94.02	2	93.16	5	83.67	9	89.44	7	**94.87**	**1**	92.29	6	78.15	10	89.44	7
Segmemt0	87.58	8	92.91	7	94.00	6	87.15	9	97.54	5	**98.85**	**1**	97.67	3	97.95	2	84.64	10	97.67	3
Yeast3	90.66	3	**93.74**	**1**	91.08	2	73.85	7	67.68	10	70.04	8	74.14	6	90.11	4	78.99	5	69.90	9
Ecoli3	92.20	4	93.09	2	**94.87**	**1**	93.09	2	52.55	8	36.84	10	74.32	7	89.44	6	92.20	4	37.16	9
Yeast2vs4	86.98	2	82.29	3	81.83	5	72.23	6	62.90	9	54.77	10	70.71	7	**88.47**	**1**	82.29	3	63.25	8
Vowel0	94.28	4	94.28	4	**99.72**	**1**	98.31	3	84.98	10	94.28	4	94.28	4	94.28	4	**99.72**	**1**	94.28	4
Glass2	73.96	2	**76.24**	**1**	73.96	2	71.61	4	57.74	6	57.74	6	0.00	9	66.67	5	55.47	8	0.00	9
Yeast1vs7	57.90	3	36.78	6	51.64	4	61.83	2	0.00	7	0.00	7	0.00	7	**64.17**	**1**	51.26	5	0.00	7
Glass4	70.71	3	70.71	3	70.71	3	**82.16**	**1**	70.71	3	70.71	3	70.71	3	**82.16**	**1**	70.71	3	70.71	3
Ecoli4	99.20	7	99.20	7	**100.00**	**1**	98.40	9	**100.00**	**1**	86.60	10	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**
Page-blocks13vs2	97.70	5	97.70	5	97.70	5	89.19	10	**100.00**	**1**	89.44	9	99.43	3	97.12	8	**100.00**	**1**	99.43	3
Abalone09-18	56.35	5	52.84	6	64.50	3	64.50	3	35.10	7	0.00	8	0.00	8	66.52	2	**67.00**	**1**	0.00	8
Glass5	0.00	-	0.00	-	0.00	-	0.00	-	0.00	-	0.00	-	0.00	-	0.00	-	0.00	-	0.00	-
Yeast4	81.42	3	83.79	2	**84.29**	**1**	77.19	4	54.29	7	54.29	7	0.00	10	72.13	5	66.64	6	31.62	9
Ecoli0137vs26	98.13	8	98.13	8	**100.00**	**1**	**99.07**	**7**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	96.23	10	**100.00**	**1**	**100.00**	**1**
Yeast6	90.97	2	**91.13**	**1**	74.54	7	90.97	2	75.46	5	37.67	10	53.27	8	74.80	6	90.64	4	37.80	9
Average		4.52		4.00		**3.13**		5.43		6.00		6.39		4.65		3.78		5.30		5.78

Table 5.5: 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	91.10	9	91.77	7	**92.15**	**1**	92.06	2	92.06	2	92.06	2	92.06	2	34.99	10	92.06	2
Pima	91.33	2	83.40	3	81.13	4	**98.17**	**1**	57.77	5	55.83	7	57.42	6	50.28	9	42.23	10	55.83	7
Glass0	76.36	5	65.12	9	69.57	6	66.70	8	77.29	4	85.37	2	**88.61**	**1**	68.14	7	56.38	10	85.37	2
Vehicle1	72.39	2	68.51	4	69.37	3	**87.47**	**1**	57.96	7	66.73	5	53.64	8	41.45	9	32.38	10	66.73	5
Vehicle0	64.03	9	74.11	7	66.00	8	76.83	6	**93.44**	**1**	86.69	2	86.69	2	77.23	5	60.70	10	82.30	4
Ecoli1	72.49	6	70.28	8	69.05	9	74.12	5	66.91	10	91.00	2	91.00	2	70.81	7	83.05	4	**100.00**	**1**
New-thyroid1	75.31	8	75.31	8	76.89	7	83.44	6	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	65.49	10	**100.00**	**1**
New-thyroid2	74.01	6	73.68	7	76.54	5	70.79	9	**100.00**	**1**	**100.00**	**1**	87.50	3	70.00	10	72.97	8	87.50	3
Ecoli2	94.21	5	92.32	6	91.41	7	87.30	8	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	75.47	9	42.99	10	**100.00**	**1**
Segmemt0	52.22	8	62.07	7	65.29	6	51.49	10	98.43	3	95.57	4	**100.00**	**1**	94.09	5	51.98	9	**100.00**	**1**
Yeast3	57.27	7	60.79	5	52.58	9	56.05	8	71.79	4	**76.51**	**1**	75.33	2	60.00	6	27.82	10	73.08	3
Ecoli3	50.22	6	51.47	4	56.68	3	51.34	5	49.92	7	24.94	10	**66.59**	**1**	36.76	8	59.84	2	33.26	9
Yeast2vs4	68.85	8	74.58	6	69.04	7	54.41	9	80.21	5	**100.00**	**1**	**100.00**	**1**	80.21	4	30.33	10	**100.00**	**1**
Vowel0	**100.00**	**1**	**100.00**	**1**	96.10	9	89.65	10	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	97.28	8	**100.00**	**1**
Glass2	44.07	3	41.41	4	31.90	6	33.58	5	**100.00**	**1**	**100.00**	**1**	0.00	9	14.72	7	11.44	8	0.00	9
Yeast1vs7	17.91	2	8.88	5	15.44	4	**25.15**	**1**	0.00	7	0.00	7	0.00	7	16.54	3	4.56	6	0.00	7
Glass4	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	39.89	8	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	16.60	9	9.94	10	**100.00**	**1**
Ecoli4	82.62	8	82.35	9	**100.00**	**1**	72.31	10	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**
Page-blocks13vs2	62.09	7	61.78	8	63.57	6	33.32	10	**100.00**	**1**	**100.00**	**1**	84.73	4	52.60	9	**100.00**	**1**	84.73	4
Abalone09-18	14.85	4	14.77	5	21.62	2	19.66	3	**34.29**	**1**	0.00	8	0.00	8	11.54	6	10.22	7	0.00	8
Glass5	0.00	-	0.00	-	0.00	-	0.00	-	0.00	-	0.00	-	0.00	-	0.00	-	0.00	-	0.00	**-**
Yeast4	19.63	7	23.57	5	24.22	4	21.01	6	37.92	2	37.92	2	0.00	10	13.85	8	5.06	9	**100.00**	**1**
Ecoli0137vs26	47.93	8	46.69	9	**100.00**	**1**	66.67	7	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	25.64	10	**100.00**	**1**	**100.00**	**1**
Yeast6	42.41	6	43.23	5	35.37	8	46.56	4	79.96	2	33.27	9	49.93	3	39.93	7	18.11	10	**100.00**	**1**
Average		5.52		5.87		5.35		6.13		**3.00**		3.09		3.30		6.22		7.57		3.22

Table 5.6: 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	**94.95**	**1**	**94.95**	**1**	**94.95**	**1**	92.16	9	94.85	4	94.85	4	94.85	4	94.85	4	51.65	10	94.85	4
Pima	57.14	4	58.29	2	57.99	3	51.46	7	49.46	10	51.02	8	54.90	5	**61.31**	**1**	52.31	6	51.02	8
Glass0	61.54	6	63.16	5	64.52	4	64.86	3	60.87	8	57.14	9	69.57	2	**73.33**	**1**	61.54	6	57.14	9
Vehicle1	45.03	5	46.74	4	48.81	2	47.13	3	35.48	10	39.34	8	42.25	6	**55.38**	**1**	40.22	7	39.34	8
Vehicle0	69.64	9	77.89	5	71.56	8	74.00	7	81.16	4	**84.21**	**1**	**84.21**	**1**	76.92	6	63.33	10	82.05	3
Ecoli1	78.95	2	75.68	5	73.68	8	66.67	9	59.26	10	76.92	3	76.92	3	75.00	6	**83.33**	**1**	75.00	6
New-thyroid1	71.43	8	71.43	8	82.35	2	77.78	3	72.73	4	72.73	4	72.73	4	**100.00**	**1**	66.67	10	72.73	4
New-thyroid2	82.35	5	82.35	5	82.35	5	63.64	10	**100.00**	**1**	92.31	4	93.33	2	82.35	5	73.68	9	93.33	2
Ecoli2	90.00	2	90.00	2	90.00	2	85.71	7	82.35	8	88.89	5	**94.74**	**1**	81.82	9	48.65	10	88.89	5
Segmemt0	58.56	8	70.65	7	73.86	6	57.78	9	96.88	4	96.97	3	**97.64**	**1**	95.45	5	53.72	10	**97.64**	**1**
Yeast3	60.19	6	66.67	2	58.72	8	34.78	10	56.60	9	60.38	5	64.29	3	**70.89**	**1**	63.64	4	59.26	7
Ecoli3	60.87	5	63.64	2	**70.00**	**1**	63.64	2	36.36	8	18.18	10	61.54	4	53.85	7	60.87	5	20.00	9
Yeast2vs4	69.57	4	70.00	2	66.67	5	42.86	10	53.33	8	46.15	9	66.67	5	**80.00**	**1**	70.00	2	57.14	7
Vowel0	94.12	3	94.12	3	**97.30**	**1**	85.71	9	83.87	10	94.12	3	94.12	3	94.12	3	**97.30**	**1**	94.12	3
Glass2	33.33	4	40.00	3	33.33	4	28.57	6	**50.00**	**1**	**50.00**	**1**	0.00	9	22.22	7	15.38	8	0.00	9
Yeast1vs7	16.22	3	8.70	6	16.00	4	17.54	2	0.00	7	0.00	7	0.00	7	**25.00**	**1**	15.38	5	0.00	7
Glass4	**66.67**	**1**	**66.67**	**1**	**66.67**	**1**	23.53	9	66.67	1	**66.67**	**1**	**66.67**	**1**	23.53	9	66.67	1	**66.67**	**1**
Ecoli4	88.89	7	88.89	7	**100.00**	**1**	80.00	10	**100.00**	**1**	85.71	9	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**
Page-blocks13vs2	71.43	6	71.43	6	71.43	6	35.71	10	**100.00**	**1**	88.89	5	90.91	3	66.67	9	**100.00**	**1**	90.91	3
Abalone09-18	12.90	7	13.04	6	**22.86**	**1**	16.00	5	18.18	3	0.00	8	0.00	8	18.87	2	17.39	4	0.00	8
Glass5	0.00	-	0.00	-	0.00	-	0.00	-	0.00	-	0.00	-	0.00	-	0.00	-	0.00	-	0.00	-
Yeast4	23.88	5	30.19	4	32.00	3	17.58	9	**33.33**	**1**	**33.33**	**1**	0.00	10	22.22	6	21.28	7	18.18	8
Ecoli0137vs26	50.00	8	50.00	8	**100.00**	**1**	66.67	7	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	33.33	10	**100.00**	**1**	**100.00**	**1**
Yeast6	52.17	3	54.55	2	42.11	7	52.17	3	**66.67**	**1**	20.00	10	36.36	8	47.06	6	48.00	5	25.00	9
Average		4.87		4.17		**3.65**		6.91		5.00		5.17		4.00		4.43		5.39		5.35

Table 5.7: *p-values* of the Wilcoxon Signed Rank Tests with SVM baseline from Experiment II

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

In conclusion, NB-Comm ranked best in F1-score (Table 5.6) and G-mean. The method was also among those that provided the highest sensitivity while its average precision was moderate. This high performance across the different measures reflects a better trade-off between sensitivity and specificity of NB-Comm than those of other methods. For instance, compared to OBU, NB-Comm provided comparable ranking in sensitivity while achieving higher precision and F1-score. NB-Basic and NB-Tomek showed competitive average ranking in F1-score. They provided different trade-offs between sensitivity, and G-means and F1-score, to OBU and kmUnder. In particular, NB-Basic and OBU had higher sensitivity but lower specificity (as can be seen from lower G-mean) than NB-Tomek and kmUnder. Thus, these methods may not be compared in general as they are suitable for different problems. To consider which method is preferable, the error costs of classes must be specified. Lastly, NB-Rec, which achieved the highest sensitivity among all methods, did not perform well in F1-score. NB-Rec is thus more desirable when the classification accuracy of the positive class cannot be compromised while misclassifying negative instances is tolerable. It was interesting that the two well-established methods SMOTE and BLSMOTE ranked best in precision but showed very low ranking in G-means, F1-score and sensitivity; also, ENN showed the least improvement over the baseline in sensitivity. Thus, these well-established methods are the least suitable solutions for handling imbalanced problems.

Table 5.8 - 5.12 present the results using RF as the learning algorithm with the same experiment settings. All NB-based methods ranked top in sensitivity; however, low precision was observed in some cases. NB-Rec achieved the highest average ranking in sensitivity among all methods but with low precision. NB-Basic provided competitive sensitivity and G-mean with kmUnder and OBU, and also higher ranking in precision and F1-score. 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 on different learning algorithms.

5.4.3 Experiment III: Large and high-dimensional datasets

In this experiment, we aimed at validating the stability of the NB-based methods on large and high-dimensional real-world datasets. In this experiment we compared our methods with the top performing methods in Experiment II based on SVM with emphasis on sensitivity, namely ENN, kmUnder, and OBU. The classification results of the methods using SVM are presented in Table 5.13.

On the breast cancer dataset, all NB-based methods significantly improved sensitivity and G-mean from the baseline and outperformed ENN and OBU in both metrics. NB-Rec yielded the highest sensitivity of 86.29% and relatively high G-mean of 79.65%, which were comparable to kmUnder. As a result of trade-offs for high sensitivity, the NB-based methods suffered more from high FP as can be seen from lower precision and F1-score. However, their FPR were reasonable as evidenced by fair G-mean. The low precision and F1-score obtained were due to the nature of the dataset that is large and extremely imbalanced, which highly affects the calculation of such metrics. That is, precision and F1-score consider FP in comparison with TP. On a large and highly imbalanced dataset, FP can be far greater than TP even if specificity is high. In this case, the breast cancer data may also suffer from high class overlap since none of the methods with relatively high sensitivity could simultaneously yield high precision and F1-score, and vice versa.

On MNIST_3, the NB-based methods improved both sensitivity and G-mean from the baseline. NB-Rec achieved the highest sensitivity of 99.18% and outperformed kmUnder in all metrics. The other NB-based methods showed competitive sensitivity with significantly higher G-means, precision, and F1-scores than kmUnder. ENN and OBU did not show any improvement over the baseline.

Table 5.8: 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	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	91.49	10	95.74	6	95.74	6	95.74	6	**100.00**	**1**	95.74	6
Pima	**100.00**	**1**	96.23	3	94.34	4	**100.00**	**1**	71.70	8	73.58	7	66.04	9	75.47	6	90.57	5	62.26	10
Glass0	**100.00**	**1**	78.57	3	78.57	3	78.57	3	64.29	7	64.29	7	64.29	7	78.57	3	**100.00**	**1**	50.00	10
Vehicle1	**100.00**	**1**	**100.00**	**1**	93.02	4	**100.00**	**1**	65.12	7	62.79	8	55.81	9	74.42	6	90.70	5	51.16	10
Vehicle0	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**
Ecoli1	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	80.00	6	80.00	6	80.00	6	80.00	6	**100.00**	**1**	80.00	6
New-thyroid1	85.71	6	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	85.71	6	85.71	6	85.71	6	**100.00**	**1**	**100.00**	**1**	85.71	6
New-thyroid2	98.20	4	**99.10**	**1**	**99.10**	**1**	**99.10**	**1**	91.89	6	90.99	7	87.39	9	96.40	5	87.39	9	88.29	8
Ecoli2	80.00	2	80.00	2	80.00	2	80.00	2	80.00	2	80.00	2	80.00	2	80.00	2	**90.00**	**1**	80.00	2
Segmemt0	**100.00**	**1**	98.46	5	98.46	5	**100.00**	**1**	98.46	5	**100.00**	**1**	98.46	5	98.46	5	**100.00**	**1**	98.46	5
Yeast3	93.75	3	93.75	3	84.38	5	**100.00**	**1**	68.75	8	84.38	5	62.50	9	**100.00**	**1**	78.13	7	62.50	9
Ecoli3	**85.71**	**1**	**85.71**	**1**	71.43	7	**85.71**	**1**	**85.71**	**1**	57.14	8	57.14	8	**85.71**	**1**	**85.71**	**1**	42.86	10
Yeast2vs4	90.00	2	90.00	2	90.00	2	90.00	2	70.00	7	60.00	8	50.00	9	**100.00**	**1**	80.00	6	50.00	9
Vowel0	94.44	3	94.44	3	94.44	3	**100.00**	**1**	94.44	3	94.44	3	94.44	3	**100.00**	**1**	94.44	3	94.44	3
Glass2	**100.00**	**1**	33.33	5	0.00	6	66.67	4	0.00	6	0.00	6	0.00	6	**100.00**	**1**	**100.00**	**1**	0.00	6
Yeast1vs7	50.00	3	50.00	3	50.00	3	**100.00**	**1**	33.33	9	33.33	9	50.00	3	**100.00**	**1**	50.00	3	50.00	3
Glass4	50.00	4	50.00	4	50.00	4	**100.00**	**1**	**100.00**	**1**	50.00	4	50.00	4	50.00	4	**100.00**	**1**	50.00	4
Ecoli4	50.00	6	50.00	6	50.00	6	75.00	4	**100.00**	**1**	75.00	4	50.00	6	**100.00**	**1**	**100.00**	**1**	50.00	6
Page-blocks13vs2	80.00	7	80.00	7	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	80.00	7	**100.00**	**1**	100.00	1	80.00	7
Abalone09-18	50.00	3	50.00	3	50.00	3	**75.00**	**1**	37.50	7	37.50	7	37.50	7	50.00	3	75.00	1	37.50	7
Glass5	**100.00**	**1**	0.00	5	0.00	5	**100.00**	**1**	0.00	5	0.00	5	0.00	5	**100.00**	**1**	**100.00**	**1**	0.00	5
Yeast4	80.00	3	70.00	5	50.00	6	90.00	2	30.00	7	30.00	7	20.00	9	**100.00**	**1**	80.00	3	10.00	10
Ecoli0137vs26	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**
Yeast6	57.14	3	42.86	5	42.86	5	42.86	5	57.14	3	42.86	5	42.86	5	**100.00**	**1**	71.43	2	42.86	5
Average		2.50		3.00		3.33		**1.63**		4.92		5.17		5.92		2.50		2.42		6.21

Table 5.9: 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	0.00	10	97.29	5
Pima	48.99	8	54.62	7	57.46	6	0.00	10	**74.30**	**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	49.80	7	59.77	6	26.83	10	73.96	2	71.23	3	71.03	4	**74.81**	**1**	39.95	9	67.40	5
Vehicle0	88.48	9	90.65	7	90.22	8	76.76	10	97.25	3	96.45	5	96.85	4	94.83	6	**97.65**	**1**	**97.65**	**1**
Ecoli1	**93.93**	**1**	**93.93**	**1**	**93.93**	**1**	92.88	5	86.77	9	85.86	10	88.56	6	88.56	6	**93.93**	**1**	88.56	6
New-thyroid1	92.58	6	**100.00**	**1**	**100.00**	**1**	98.60	3	92.58	6	92.58	6	92.58	6	95.74	5	97.18	4	92.58	6
New-thyroid2	94.60	4	95.20	2	89.17	9	64.71	10	95.03	3	94.56	5	92.91	7	**95.34**	**1**	92.91	7	93.43	6
Ecoli2	85.36	10	88.64	8	89.44	2	86.19	9	89.44	2	89.44	2	89.44	2	89.44	2	**92.29**	**1**	89.44	2
Segmemt0	98.08	9	99.23	3	98.85	7	99.24	2	99.10	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	78.30	10	89.82	6	83.92	7	78.46	9
Ecoli3	80.18	6	84.52	3	80.92	5	83.67	4	**87.83**	**1**	71.71	9	74.96	8	85.36	2	79.28	7	65.47	10
Yeast2vs4	93.31	3	93.83	2	**94.35**	**1**	93.31	3	83.21	7	77.04	8	70.71	9	92.67	5	88.96	6	70.71	9
Vowel0	95.26	7	95.54	6	94.71	8	91.24	10	**97.18**	**1**	**97.18**	**1**	**97.18**	**1**	97.17	5	93.87	9	**97.18**	**1**
Glass2	**86.23**	**1**	53.91	5	0.00	6	70.41	2	0.00	6	0.00	6	0.00	6	67.94	3	64.05	4	0.00	6
Yeast1vs7	64.17	7	66.42	6	69.87	5	**90.10**	**1**	56.01	9	57.05	8	70.71	2	48.51	10	70.71	2	70.71	2
Glass4	69.82	3	69.82	3	68.92	8	93.54	2	**98.74**	**1**	69.82	3	69.82	3	68.92	8	67.08	10	69.82	3
Ecoli4	70.71	6	70.71	6	70.71	6	83.81	5	**100.00**	**1**	86.60	4	70.71	6	96.77	3	**100.00**	**1**	70.71	6
Page-blocks13vs2	87.39	8	87.39	8	97.12	4	95.94	5	**100.00**	**1**	**100.00**	**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**	0.00	5	0.00	5	**96.27**	**1**	0.00	5	0.00	5	0.00	5	93.70	4	**96.27**	**1**	0.00	5
Yeast4	83.96	3	80.84	4	68.32	5	**87.81**	**1**	54.10	7	54.48	6	44.72	8	11.83	10	84.79	2	31.62	9
Ecoli0137vs26	**99.07**	**1**	**99.07**	**1**	**99.07**	**1**	**99.07**	**1**	**99.07**	**1**	**99.07**	**1**	98.13	8	57.74	10	**99.07**	**1**	98.13	8
Yeast6	74.01	3	64.78	8	65.12	6	64.32	9	75.33	2	65.12	6	65.35	4	13.15	10	**80.62**	**1**	65.35	4
Average		5.13		4.46		4.83		4.75		**4.29**		4.83		5.21		5.25		5.13		5.50

Table 5.10: 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	34.90	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	**100.00**	**1**	89.74	4	**100.00**	**1**	72.78	5	51.15	9	**100.00**	**1**
Vehicle1	30.34	8	31.45	7	34.25	6	27.10	10	58.40	3	53.01	4	**66.73**	**1**	50.87	5	27.52	9	61.18	2
Vehicle0	58.63	9	63.30	7	62.31	8	42.81	10	85.00	3	81.51	5	83.22	4	75.32	6	**86.86**	**1**	**86.86**	**1**
Ecoli1	71.65	7	71.65	7	71.65	6	68.41	10	80.17	4	75.20	5	92.38	3	**92.38**	**1**	71.65	7	**92.38**	**1**
New-thyroid1	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	87.50	8	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**70.00**	**10**	77.78	9	**100.00**	**1**
New-thyroid2	87.50	6	87.50	6	**100.00**	**1**	70.00	10	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	77.78	9	87.50	6	**100.00**	**1**
Ecoli2	62.13	10	89.13	7	**100.00**	**1**	67.22	9	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	75.47	8	**100.00**	**1**
Segmemt0	81.41	9	**100.00**	**1**	95.57	6	91.63	8	98.48	5	98.50	4	**100.00**	**1**	95.57	7	26.51	10	**100.00**	**1**
Yeast3	60.43	7	67.06	6	71.42	5	52.90	8	73.68	4	77.46	3	80.28	2	38.98	10	49.46	9	**83.58**	**1**
Ecoli3	28.50	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	**100.00**	**1**
Yeast2vs4	75.25	9	82.02	7	90.12	3	75.25	8	87.64	5	85.88	6	**100.00**	**1**	43.81	10	89.02	4	**100.00**	**1**
Vowel0	70.76	6	73.85	5	65.31	7	37.42	10	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	64.21	8	58.54	9	**100.00**	**1**
Glass2	**25.18**	**1**	18.33	2	0.00	6	18.33	3	0.00	6	0.00	6	0.00	6	13.81	4	12.76	5	0.00	6
Yeast1vs7	16.54	9	22.91	8	59.77	4	27.09	7	28.38	6	49.77	5	**100.00**	**1**	8.38	10	**100.00**	**1**	**100.00**	**1**
Glass4	56.40	2	56.40	2	39.27	7	34.10	9	**72.12**	**1**	56.40	2	56.40	2	39.27	7	10.52	10	56.40	2
Ecoli4	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	42.78	10	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	49.92	9	**100.00**	**1**	**100.00**	**1**
Page-blocks13vs2	52.60	6	52.60	6	52.60	8	44.22	9	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	84.73	5	6.34	10	**100.00**	**1**
Abalone09-18	18.02	7	19.72	6	20.70	5	17.27	8	34.29	4	43.91	3	**100.00**	**1**	10.66	10	12.71	9	**100.00**	**1**
Glass5	37.50	2	0.00	5	0.00	5	**37.50**	**1**	0.00	5	0.00	5	0.00	5	26.47	4	37.50	2	0.00	5
Yeast4	19.32	8	27.27	5	21.13	7	18.26	9	30.37	4	50.44	3	**100.00**	**1**	3.48	10	21.92	6	**100.00**	**1**
Ecoli0137vs26	57.98	5	57.98	5	**57.98**	**1**	**57.98**	**1**	**57.98**	**1**	**57.98**	**1**	40.82	9	3.69	10	57.98	5	40.82	8
Yeast6	24.95	7	33.27	6	49.93	4	23.03	8	66.61	3	49.93	4	**74.95**	**1**	2.40	10	16.09	9	74.95	2
Average		6.17		5.13		4.67		7.88		3.17		3.38		2.13		7.04		7.00		**1.96**

Table 5.11: F1-score values and ranks with RF baseline from Experiment II

Dataset	F1-Score Value/Rank																			
	NB-Basic		NB-Tomek		NB-Comm		NB-Rec		SMOTE		BLSMOTE		ENN		kmUnder		OBU		Baseline	
Wisconsin	**98.95**	**1**	**98.95**	**1**	97.92	3	96.91	4	93.48	9	95.74	7	96.77	6	95.74	8	51.65	10	96.77	5
Pima	58.24	8	58.96	7	59.52	6	51.46	9	66.67	2	66.67	2	66.04	4	**66.67**	**1**	50.26	10	64.08	5
Glass0	65.12	10	68.75	7	70.97	5	70.97	5	**78.26**	**1**	75.00	4	**78.26**	**1**	75.86	3	68.29	8	66.67	9
Vehicle1	46.49	8	47.78	7	50.00	6	42.57	9	**61.54**	**1**	57.45	4	60.76	2	60.38	3	42.16	10	55.70	5
Vehicle0	73.58	9	77.23	7	76.47	8	59.54	10	91.76	3	89.66	5	90.70	4	85.71	6	**92.86**	**1**	**92.86**	**1**
Ecoli1	83.33	5	83.33	5	83.33	4	81.08	8	80.00	9	77.42	10	**85.71**	**1**	85.71	2	83.33	5	85.71	2
New-thyroid1	92.31	7	**100.00**	**1**	**100.00**	**1**	93.33	3	92.31	4	92.31	4	92.31	4	82.35	10	87.50	9	92.31	7
New-thyroid2	93.33	6	93.33	6	**100.00**	**1**	82.35	10	**100.00**	**1**	**100.00**	**1**	**100.00**	**1**	87.50	9	93.33	6	**100.00**	**1**
Ecoli2	69.57	10	84.21	7	88.89	3	72.73	9	88.89	3	88.89	3	88.89	3	**88.89**	**1**	81.82	8	**88.89**	**1**
Segmemt0	89.66	9	99.22	3	96.97	7	95.59	8	98.46	5	**99.24**	**1**	99.22	2	96.97	6	41.67	10	99.22	3
Yeast3	73.17	4	77.92	2	77.14	3	68.82	8	70.97	6	**80.60**	**1**	70.18	7	55.65	10	60.24	9	71.43	5
Ecoli3	42.86	9	52.17	6	58.82	4	50.00	7	63.16	2	47.06	8	**66.67**	**1**	54.55	5	41.38	10	60.00	3
Yeast2vs4	81.82	5	85.71	2	**90.00**	**1**	81.82	4	77.78	6	70.59	7	66.67	9	60.61	10	84.21	3	66.67	8
Vowel0	80.95	6	82.93	5	77.27	8	54.55	10	**97.14**	**1**	**97.14**	**1**	**97.14**	**1**	78.26	7	72.34	9	97.14	4
Glass2	**37.50**	**1**	22.22	3	0.00	6	26.67	2	0.00	6	0.00	6	0.00	6	22.22	3	20.69	5	0.00	6
Yeast1vs7	25.00	9	31.58	7	54.55	4	42.86	5	30.77	8	40.00	6	66.67	3	15.58	10	**66.67**	**1**	**66.67**	**1**
Glass4	50.00	2	50.00	2	40.00	8	44.44	7	**80.00**	**1**	50.00	2	50.00	2	40.00	8	15.38	10	50.00	2
Ecoli4	66.67	4	66.67	4	66.67	8	54.55	10	**100.00**	**1**	85.71	3	66.67	8	66.67	4	**100.00**	**1**	66.67	4
Page-blocks13vs2	61.54	7	61.54	7	66.67	6	58.82	9	**100.00**	**1**	**100.00**	**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	40.00	3	**54.55**	**1**	17.02	10	21.05	9	54.55	2
Glass5	**40.00**	**1**	0.00	5	0.00	5	**40.00**	**1**	0.00	5	0.00	5	0.00	5	28.57	4	**40.00**	**1**	0.00	5
Yeast4	30.77	5	**38.89**	**1**	29.41	8	30.00	6	30.00	6	37.50	2	33.33	4	6.62	10	34.04	3	18.18	9
Ecoli0137vs26	**66.67**	**1**	**66.67**	**1**	66.67	4	66.67	4	66.67	4	66.67	4	50.00	8	5.26	10	**66.67**	**1**	50.00	8
Yeast6	34.78	7	37.50	6	46.15	4	30.00	8	**61.54**	**1**	46.15	4	54.55	2	4.70	10	26.32	9	54.55	3
Average		5.92		4.50		4.92		6.79		**3.75**		3.92		**3.75**		6.38		6.58		4.29

Table 5.12: *p-values* of the Wilcoxon Signed Rank Tests with RF baseline from Experiment II

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

Table 5.13: Results on large and high-dimensional datasets

Dataset	Metric	NB-Basic	NB-Tomek	NB-Comm	NB-Rec	ENN	kmUnder	OBU	Baseline
Breast Cancer	sensitivity	64.52	58.87	60.48	**86.29**	40.32	**86.29**	42.74	28.23
	G-mean	78.83	76.10	76.69	79.65	63.48	**83.86**	65.35	53.12
	precision	9.66	18.02	11.81	1.95	**81.97**	2.77	73.61	81.40
	F1-score	16.81	27.60	19.76	3.81	54.05	5.36	**54.08**	41.92
MNIST_3	sensitivity	94.29	92.65	93.47	**99.18**	82.45	95.92	82.45	82.45
	G-mean	**94.82**	94.56	94.38	77.19	90.71	59.22	90.71	90.71
	precision	31.56	37.58	31.11	5.35	**90.18**	3.32	**90.18**	**90.18**
	F1-score	47.29	53.47	46.69	10.15	**86.14**	6.43	**86.14**	**86.14**
MNIST_5	sensitivity	97.42	97.42	97.42	**99.26**	91.14	95.94	90.96	90.96
	G-mean	96.26	**96.85**	95.52	70.98	95.28	93.85	95.18	95.18
	precision	49.72	56.53	43.28	9.10	**91.99**	36.78	91.98	91.98
	F1-score	65.84	71.54	59.93	16.67	**91.57**	53.17	91.47	91.47

Table 5.14: Comparative results with evolutionary and deep learning-based methods

Dataset	NB-based methods				EVINCI	CnGRSOMO*	CnGRSOMU*			
	Sensitivity	G-mean	Precision	F1-score	G-mean	F1-score	Sensitivity	G-mean	Precision	F1-score
Ecoli2	-	**89.44**	-	-	86.20	-	-	-	-	-
Ecoli1	-	-	-	**83.33**	-	78.10	-	-	-	-
Ecoli3	-	-	-	58.82	-	**64.50**	-	-	-	-
Abalone0918	75.00	76.54	17.27	27.27	-	-	**83.50**	**84.48**	**25.00**	**39.00**
Yeast4	**90.00**	**87.81**	**18.26**	**30.00**	-	-	75.00	79.93	15.00	25.00

*Estimated results from graphs.

Similarly, on MNIST_5, the NB-based methods showed significant improvements in sensitivity. NB-Rec achieved the highest sensitivity of 99.26% but had the lowest results in the other metrics. NB-Basic, NB-Tomek and NB-Comm yielded better results than kmUnder across all metrics. They also showed higher sensitivity and G-mean than ENN and OBU. Although our methods had lower precision and F1-score than ENN and OBU, it is worth point out that OBU did not improve the results from the baseline whereas ENN rarely did. The low precision and F1-score of our methods were due to unavoidable trade-offs on large and highly imbalanced datasets as discussed above. Thus, it can be said that among the methods that promoted the detection of the class of interest, NB-Tomek had the best performance followed by NB-Basic and NB-Comm.

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

5.5 Performance Comparison with Evolutionary and Deep Learning-Based Methods

In this section, the performance of the NB-based methods with RF classifiers is compared with evolutionary and deep learning-based methods, namely EVINCI [81], CnGRSOMO [62], CnGRSOMU [62] and SMOTE-CSELM [63]. Comparative results, which are based on availability of the results in the literature, are presented in Table 5.14 and Table 5.15. In the column named NB-based methods, results from the best performing method among the four variants are displayed. Note that if there are more than one available measure for the dataset, the best performing method was selected primarily based on sensitivity. The higher value between the NB-based methods and the compared method on each dataset is highlighted in bold.

Table 5.15: Comparative results with a deep learning-based method

Dataset	NB-based methods		SMOTE-CSELM	
	Sensitivity	G-mean	Sensitivity	G-mean
Wisconsin	**100.00**	**99.43**	98.74	97.99
Pima	**100.00**	48.99	80.96	**76.65**
Glass0	**100.00**	68.14	**100.00**	**82.01**
Vehicle1	**100.00**	49.80	98.00	**86.17**
Vehicle0	**100.00**	90.65	**100.00**	**99.46**
Newthyroid1	**100.00**	**100.00**	**100.00**	99.16
Newthyroid2	-	95.20	-	**99.16**
Ecoli2	80.00	89.44	**100.00**	**93.64**
Segment0	**100.00**	99.24	**100.00**	**99.67**
Yeast3	**100.00**	**94.35**	**100.00**	93.54
Ecoli3	85.71	84.52	**94.29**	**91.52**
Yeast2vs4	90.00	94.35	**100.00**	**94.73**
Vowel0	-	91.24	-	**100.00**
Glass2	**100.00**	86.23	**100.00**	**86.87**
Yeast1vs7	**100.00**	**90.10**	**100.00**	79.58
Glass4	**100.00**	93.54	**100.00**	**98.22**
Ecoli4	75.00	83.81	**100.00**	**98.40**
Abalone0918	75.00	76.54	**93.06**	**90.61**
Ecoli0137vs26	**100.00**	**99.07**	**100.00**	79.06
Yeast6	57.14	74.01	**100.00**	**89.54**

As can be seen in Table 5.14, the NB-based methods were comparable to EVINCI, CnGRSOMO and CnGRSOMU on the given datasets and metrics. Table 5.15 shows that the NB-based methods achieved the highest sensitivity on 12 out of 18 datasets and the highest G-mean on 5 out of 20 datasets. SMOTE-CSELM had the highest sensitivity and G-mean each on 15 datasets. These results suggest that the NB-based methods performed comparably to SMOTE-CSELM in terms of sensitivity but did not perform as well in G-mean. However, the NB-based methods require lower running time than SMOTE-CSELM, which needs $O(N^3)$ [63]. This will make the NB-based methods more desirable when the classification problem is extremely time-critical and mainly focus on the sensitivity.

5.6 Conclusions

In this chapter, a novel undersampling approach to handle classification of imbalanced and overlapped datasets was presented. The approach is based on neighbourhood searching to identify and eliminate potential negative instances in the overlapping region. Four different variants of the approach were designed and evaluated using simulated and real-world datasets. The four methods were compared against well-established and state-of-the-art methods. Results showed that our methods achieved the highest sensitivity with competitive G-means across all imbalance degrees on both simulated and real-world datasets. They also showed competitive performance across all degrees of class overlap on the simulated datasets. The four variants of the proposed approach provided different benefits and trade-offs as follows: 1) NB-Rec yielded exceptionally high sensitivity at all degrees of class imbalance and class overlap but showed higher FP at higher imbalance degrees. 2) NB-Basic resulted in competitive sensitivity with lower FP than the state-of-the-art methods; 3) NB-Tomek and NB-Comm showed similar trade-offs and were comparable to state-of-the-art methods in all metrics. These offer alternative potential solutions that suit different real-world problems.

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

Chapter 6

Medical Application

In this chapter, a framework for predictive diagnostics of diseases with imbalanced records is presented. Early diagnosis, especially of some life-threatening diseases such as cancers and heart, is crucial for effective treatments. Supervised machine learning has proved to be a very useful tool to serve this purpose. Historical data of patients including clinical and demographic information is used for training learning algorithms. This builds predictive models that provide initial diagnoses. However, in the medical domain, it is common to have the positive class under-represented in a dataset. In such a scenario, a typical learning algorithm tends to be biased towards the negative class, which is the majority class, and misclassify positive cases. To reduce the classification bias, we propose the usage of our overlap-based undersampling methods to improve the visibility of the minority class in the region where the two classes overlap. Results showed more successful application of our methods over others with higher prediction accuracy in positive cases and good trade-offs between sensitivity and specificity. *Part of this work was reported in the 16th International Conference on Artificial Intelligence Applications [147] and Innovations, and another part is to be published in the International Journal of Neural Systems [131].*

6.1 Overview

In the medical domain, it is important that prevention and early diagnosis are carried out to avoid further complications and achieved better treatment outcomes [148]. Hence, detecting possible existence or occurrence of diseases is of high interest in supervised learning. This is achieved by training classification models to predict patients' conditions based on the given symptoms and their personal information. However, it is common that a medical dataset has an uneven class distribution. In many situations, the class of interest rarely occurs, hence its samples are relatively limited and under-represented in the data. Traditional learning algorithms tend to be biased towards the majority class and fail to detect anomaly cases, which belong to the minority class. A number of solutions have been proposed to handle such an issue. Many of them focused on a medical dataset of a specific disease [141,149,150] while others proved their performance on several medical-related datasets [151, 152].

Learning from imbalanced medical datasets are seen in a wide range of problems. Besides classification of well-known public datasets such as breast cancer Wisconsin and Pima Indian diabetes, other types of classification tasks have also been carried out. These include classification of electrodiogram (ECG) heartbeats [153], image classification of breast cancer [141] and video classification of bowel cancer [149]. Regardless of problem types, a common objective is to achieve high prediction accuracy, especially on the positive class.

Rebalancing class distributions seems to be a typical approach to handle imbalanced medical datasets [141, 154]. However, it was shown in literature and also earlier in this work that solutions based on improving the visibility of positive samples in the overlapping region could produce significantly higher positive class accuracy [1, 18, 48].

In this chapter, an application of the overlap-based undersampling methods on medical diagnoses from imbalanced datasets is presented. One of each OBU-based and NB-based methods will be demonstrated. Since high sensitivity is preferred, BoostOBU and NB-Rec, which often achieved the highest sensitivity among the methods, were used. Datasets of various diseases were considered. These include heart disease, cancers, thyroid and some of the most common neurological disorders, namely epilepsy and Parkinson's disease (PD).

6.2 Towards Computer-Aided Diagnosis for Imbalanced Medical Records

Despite high interests in classification of medical data, the common issue of imbalanced class distributions is not often addressed [155]. This is evidenced by a review paper discussing existing methods used for medical datasets classification, where only 1 out of 71 proposed solutions take into account the class imbalanced issue [155].

To tackle class imbalance, long-established methods such as random undersampling, SMOTE [12], ENN [105] and ADASYN [74] were still used in many recent studies

[148, 156]. Although improvements in results were reported, they have been constantly outperformed by newer methods. Novel methods for handling imbalanced medical datasets have also been proposed. In [152], the authors selectively oversampled minority class instances based on their nearest neighbours. Minority class instances were defined as noise, unstable or boundary samples. Then, noisy instances were removed and only boundary instances were oversampled using linear interpolation techniques. The method showed improvement over SMOTE and a SMOTE-based method. However, it has disadvantages of high parameter dependency and the risk of losing important information in eliminating minority class instances.

In [157], a new technique for determining the final output of the classifier was developed. Unlike the traditional maximum vote approach, classes were predicted based on the highest weight that was the combination of accuracy, sensitivity, specificity and AUC. Results showed trivial improvement over the traditional method. More importantly, the improvement might not be attributed to increases in the minority class accuracy, which is highly desirable in the medical domain.

Wan et al. designed a scoring function that assigned ranking to differentiate between minority class and majority class instances [158]. Boosting was adopted to carried out automatic scoring. The method could improved sensitivity on medical datasets further than a cost-sensitive approach and other well-known ensemble-based methods. Moreover, it has the benefit of no prior costs required, which is often unknown and hard to estimate.

One of the latest techniques, Generative Adversarial Net (GAN), was employed in [151] to synthesise minority class instances. It was combined with a multilayer extreme learning machine (ELM) algorithm and showed superior performance to other techniques such as weighting and SMOTE. The method was also shown to consumed low computational time.

Rather than using a method to broadly handle datasets of multiple diseases, many studies focused on a specific disease such as cancers [141, 149, 156], polyps [159] and osteoporosis [148]. For instance, Yuan et al. proposed an ensemble-based deep learning approach for detecting bowel cancer [149]. They modified the loss function to penalise the classifier when it misclassified samples that were correctly classified in the previous iteration. However, results showed that the method was comparable to a long-established ensemble, RUSBoost, in terms of sensitivity and computational time. Other methods for classification of cancer datasets were also proposed [141, 160]. In [141], an evolutionary algorithm was used as an undersampling technique to select the most significant samples. The undersampling was then used in combination with Boosting. Results showed that

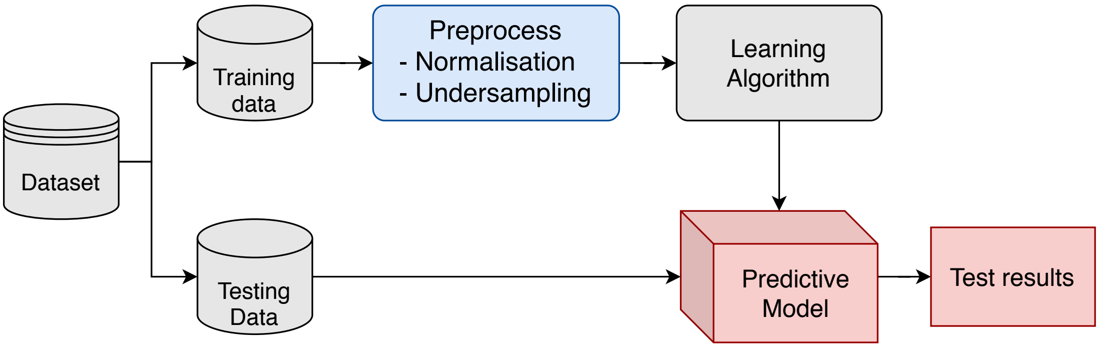


Figure 6.1: The proposed framework for classification of imbalanced medical datasets

classification of a breast cancer dataset was improved compared to other ensemble-based techniques. Similarly, in [160], a cost-sensitive ensemble integrated with a genetic algorithm was proposed to handle an imbalanced breast thermogram dataset. The method provided higher sensitivity than other existing ones. However, a common drawback of these ensemble-based solutions is high computational costs.

Electrocardiogram (ECG) of heartbeats is also of high interest and generally highly imbalanced, where most heartbeats are normal. With complicated components and morphology of ECG, deep convolutional neural networks (CNN) are often employed for classification tasks [150,153]. To enhance the performance, CNN is used in combination with many other techniques such as Borderline-SMOTE, feature selection and two-phase training [161]. The two-phase training technique introduced by Havaei et al. [161] is known as a promising training technique for imbalanced data. In the first stage, balanced data is used for training so that CNN can distinguish different classes. Then, in the second stage, the original imbalanced data is fed to fine-tune the output layer parameters.

6.3 Improving Predictive Models

The framework for improving prediction on imbalanced medical datasets is presented in Fig. 6.1. Firstly, the training data is preprocessed using normalisation and an overlap-based undersampling technique. Then, the preprocessed data is used to train a learning algorithm to build a predictive model. Finally, the model is evaluated on the testing data.

In the data preprocessing step, an overlap-based undersampling method is applied aiming at maximising the presence of minority class instances in the overlapping region. BoostOBU or NB-Rec, which were shown in the previous chapters to often achieve the highest sensitivity while having competitive G-mean with other methods, is used. Since both methods employ distance-based techniques, they are prone to noise sensitivity. To address the issue, the data is normalised before BoostOBU or NB-Rec is applied. The standard scores (z-scores) are used as the normalisation method.

6.4 Application of BoostOBU

The use of BoostOBU in the framework was demonstrated on predictive diagnostics of neurological disorders that widely affect people around the world and increase the risk of premature death – epilepsy and Parkinson's disease. We used an epileptic seizure recognition dataset and a PD dataset obtained from the UCI repository [129].

Epileptic seizure

The epileptic seizure recognition dataset contains brain activity in the form of Electroencephalogram (EEG) signals. It has $11,500$ samples, of which are $2,300$ epileptic seizure (positive) cases and $9,200$ cases with no seizures (negative) making $imb = 4$ or 80% negative instances. Each sample consists of 178 features, which are the values of EEG recorded at a different point in time.

Parkinson's disease

The data was collected by Max Little of the University of Oxford, in collaboration with the National Centre for Voice and Speech, Colorado. The dataset contains 195 speech signal samples with $imb = 3.06$, which is 147 samples with PD and 48 healthy samples. Each sample has 23 features, but only 22 were used in the experiment as the patient ID was excluded. Note that on this dataset, even though the positive class (PD) is the majority class, all resampling methods were not modified and were applied based on the minority and majority class concept.

The same settings as detailed in Chapter 4, where BoostOBU with SVM as the learning algorithm was proven successful, were used. Each dataset was partitioned into a training set and a testing set at 80:20, where the testing data was only used to evaluate the model for the result report. During model training, 10-fold cross-validation was employed to automatically select the C parameter of SVM for the best overall accuracy. Results are ported in terms of sensitivity, specificity, G-mean and F1-score, which are common evaluation metrics used in the epilepsy and PD-related literature [162, 163].

Table 6.1: Results on epilepsy and Parkinson's disease predictions

Dataset	Metric/μ_{th}	Baseline	SMOTE	BLSMOTE	kmUnder	SMOTE-ENN	SMTBagging	RusBoost	BoostOBU
Epilepsy	sensitivity	92.83	97.83	96.52	93.04	97.83	95.87	98.04	**98.26**
	specificity	98.10	97.23	97.45	95.63	97.28	**98.15**	97.12	92.88
	G-mean	95.43	97.53	96.98	94.33	97.55	97.00	**97.58**	95.53
	F1-score	92.62	93.65	93.38	**98.21**	93.75	94.33	93.57	86.67
	μ_{th}	-	-	-	-	**-**	-	-	0.499985
Parkinson's	sensitivity	96.55	96.55	**100.00**	89.66	93.10	75.86	96.55	**100.00**
	specificity	55.56	77.78	77.78	88.89	44.44	**100.00**	88.89	**100.00**
	G-mean	73.24	86.66	88.19	89.27	64.33	87.10	92.64	**100.00**
	F1-score	91.80	94.92	96.67	92.86	88.52	86.27	96.55	**100.00**
	μ_{th}	-		-	-	-	-	-	0.266147

The performance of BoostOBU was compared against SMOTE [12], BLSMOTE [59], k-means undersampling (kmUnder) [14], SMOTE-ENN [132], SMOTEBagging [68], and RUSBoost [67]. The parameter setting discussed in Chapter 4 was followed.

6.4.1 Results and Discussions

The proposed framework with BoostOBU showed favourable classification performance on the epileptic seizure and PD datasets. Highest sensitivity among the methods was achieved on both datasets. Detailed results are shown in Table 6.1, where the highest value in each evaluation metric is highlighted in **bold**.

Table 6.1 shows that BoostOBU provided the highest detection rate of epileptic seizures of 98.26%. Even though it obtained the lowest accuracy on the healthy cases, which is indicated by low specificity and F1-score, its G-mean was comparable to the other methods. This suggests a comparable trade-off between sensitivity and specificity. Moreover, it can be seen that all methods performed relatively well. This could be partly due to the availability of sufficient samples of both classes and a low imbalance degree even though there may be a high degree of class overlap as evidenced by μ_{th} near 0.5 of BoostOBU.

On the PD dataset, BoostOBU achieved 100% accuracy on both PD and healthy test cases, which clearly outperformed all other methods. The low μ_{th} indicates that the dataset tends to have small class overlap, and thus relatively few majority class samples might have been removed by BoostOBU. Among the methods, which include both class distribution-based and class overlap-based methods, BoostOBU is the only method whose sampling amount depend on class overlap, not the class imbalance degree. This may have enabled more accurate and necessary removal of overlapped majority class instances over the other methods, which led to the highest performance of BoostOBU.

Table 6.2: Datasets for the Application of NB-Rec

dataset	instances	features	imb	%neg
Wisconsin	683	9	1.86	65.00
Thoracic	470	17	5.71	85.11
Cleveland	173	13	12.31	92.49
Thyroid	7200	21	12.48	92.58
Breast cancer	102294	117	163.20	99.39

6.5 Application of NB-Rec

Five well-known medical datasets – Wisconsin, Thoracic, Cleveland, Thyroid and Breast cancer, were experimented. The first four were obtained from the UCI repository [129]. The Breast cancer dataset was given as a challenge in the KDD Cup 2008[1]. In all datasets, the positive class is the minority class. We cleaned the datasets so that there were no missing values. Their general information are presented in Table 6.2 in ascending order of imbalance ratio.

Wisconsin breast cancer

The Wisconsin breast cancer dataset, widely-known as Wisconsin, was collected at the University of Wisconsin Hospitals, USA during 1989-1991. The class labels are diagnoses of malignant (positive) or benign (negative) breast mass. Other given information is cells characteristics.

Thoracic surgery

The data was collected from patients who underwent major lung resections for primary lung cancer at Wroclaw Thoracic Surgery Centre, Poland during 2007-2011. The prediction labels are one-year survival period, which are died (positive) and survived (negative). Model training and prediction will be based on patients' personal information, conditions, behavior and symptoms.

Cleveland heart disease

The dataset consists of databases obtained from patients in different regions: Cleveland, Long Beach, Hungary and Switzerland. Patients with the presence of heart disease (positive) are to be distinguished from those with absence (negative).

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

Thyroid

The records were provided by the Garavan Institute of Sydney, Australia. The objective is to determine whether a patient referred to the clinic is hypothyroid. The original dataset contains 3 classes: normal, hyperfunction and subnormal function. In this experiment, we identified both hyperfunction and subnormal function as hypothyroid (positive). The normal cases (negative) occupies over 92 % of the dataset.

Breast cancer

The dataset is composed of features computed from X-ray images of breasts for early detection of breast cancer. Each sample is labelled with malignant (positive) or benign (negative). This dataset is very large and extremely imbalanced with positive instances of less than 1%.

All datasets were partitioned into training and testing data at 70:30, where the testing data was only used during model evaluation for the result report. RF was chosen as the learning algorithm as it is one of the most-frequently used classifiers for imbalanced datasets [11]. Also, it showed promising results on sensitivity with a better trade-off between sensitivity and specificity than other algorithms [148]. This is also evidenced by the result presented in Experiment II of Chapter 5. The default settings of RF in *caret* package in *R* including $mtree = 500$ were used. Results are provided in terms of sensitivity, specificity, G-mean and F1-score, which allow broad comparison with the literature. The performance of NB-Rec was compared against well-established and state-of-the-art algorithms. These were SMOTE [12], BLSMOTE [59], DBSMOTE [39] and k-means undersampling [14]. The parameters of these methods were set as in the original works. Except for KDD's breast cancer, where sufficient data was available, 10-fold cross-validation was used in the training phase for the purpose of model selection with the best *mtry* setting in RF.

6.5.1 Results and Discussions

Experimental results showed that the proposed framework with NB-Rec was effective in handling classification of the medical datasets. NB-Rec showed better results than the well-established and state-of-the-art methods by achieving the highest sensitivity and the highest G-mean on most datasets. Across all datasets, sensitivity and G-mean were significantly improved over the baseline (RF with no resampling). These results are presented in Table 6.3 - 6.7.

Table 6.3: Results on Wisconsin

method	sensitivity	specificity	G-mean	F1-score
baseline	94.37	**96.97**	95.66	**94.37**
NB-Rec	**98.59**	93.18	**95.85**	93.33
SMOTE	94.37^a	96.97^a	95.66^a	$\mathbf{94.37}^a$
BLSMOTE	94.37^a	96.97^a	95.66^a	$\mathbf{94.37}^a$
DBSMOTE	94.37^a	96.97^a	95.66^a	$\mathbf{94.37}^a$
kmUnder	95.77^b	95.45^b	95.61^b	93.79^b

[a] No changes in the results after applying the method
[b] Results obtained with modified parameter setting

Table 6.4: Results on Thoracic

method	sensitivity	specificity	G-mean	F1-score
baseline	0.00	**99.17**	0.00	0.00
NB-Rec	**95.24**	5.83	23.57	**25.97**
SMOTE	9.52	89.17	29.14	11.11
BLSMOTE	9.52	87.50	28.87	10.53
DBSMOTE	9.52	97.50	30.47	15.38
kmUnder	80.95	20.83	**41.07**	25.56

Table 6.3 shows the results on Wisconsin breast cancer dataset. The NB-Rec method provided the highest sensitivity of 98.59% and the highest G-mean of 95.85%. These were achieved with high specificity and F1-score. It should be noted that the other methods failed to work on this dataset. In particular, the SMOTE-based methods, i.e., SMOTE, BLSMOTE and DBSMOTE, had no effects on the classification results. This could have been because insufficient positive samples were synthesised, which was due to their objective to rebalance data. As a result, the presence of the positive class, especially around the boundary regions, could not be improved. As opposed, NB-Rec does not factor the imbalance ratio and the removal only depends on the amount of class overlap. Lastly, kmUnder could not be carried out using the k value proposed in the original work since there were fewer distinct samples than k. Thus, we replaced it with $k = N_{minority}/2$. However, it did not give better results than NB-Rec.

As shown in Table 6.4, NB-Rec achieved the best sensitivity and F1-score on Thoracic surgery dataset. It is worth pointing out that this dataset is very hard to classify. This can be seen from the baseline results that none of the positive test cases were correctly identified. Moreover, none of the methods could produce high sensitivity and high specificity at the same time. These high trade-offs between the accuracy of the two classes indicates that the dataset is likely to suffer from severe class overlap. Due to such a trade-off, the NB-Rec method had the lowest specificity but achieve very high sensitivity of 95.24% compared to 9.52% of the SMOTE-based methods and 80.95% of kmUnder.

Table 6.5: Results on Cleveland

method	sensitivity	specificity	G-mean	F1-score
baseline	33.33	**100.00**	57.74	50.00
NB-Rec	**100.00**	93.75	96.82	66.67
SMOTE	**100.00**	97.92	98.95	85.71
BLSMOTE	**100.00**	91.67	95.74	60.00
DBSMOTE	**100.00**	**100.00**	**100.00**	**100.00**
kmUnder	**100.00**	39.58	62.92	17.14

Table 6.6: Results on Thyroid

method	sensitivity	specificity	G-mean	F1-score
baseline	98.74	99.75	99.24	**97.82**
NB-Rec	**100.00**	99.20	**99.60**	95.21
SMOTE	98.74[a]	99.75[a]	99.24[a]	**97.82**[a]
BLSMOTE	98.11	98.15	98.13	88.64
DBSMOTE	98.74[a]	99.75[a]	99.24[a]	**97.82**[a]
kmUnder	0.00	**100.00**	0.00	0.00

[a] No changes in the results after applying the method

From Table 6.5, NB-Rec perfectly classified the positive test cases on the Cleveland heart disease dataset. Its specificity and G-mean were high and comparable to SMOTE, BLSMOTE and DBSMOTE. Due to the high class imbalance nature of the dataset, F1-score of NB-Rec was much lower than those of SMOTE and DBSMOTE even though their specificity values were not far different. This is because F1-score considers TP and FP. Thus, in a highly class imbalanced situation, F1-score can be strongly affected by high FP, which could be misleading when considering the metric alone. Compared to kmUnder, NB-Rec provided a substantially better trade-off between sensitivity and specificity. This could be attributed to less information loss of the NB-Rec method.

As can be seen from Table 6.6, the NB-Rec method provided the highest sensitivity of 100% as well as the highest G-mean of 99.60% on the Thyroid dataset. This is evidence of the best trade-off between sensitivity and specificity among the methods. NB-Rec also yielded high specificity and F1-score, which were competitive with the other methods except kmUnder, which completely failed to handle the dataset. SMOTE and DBSMOTE led to no changes in the classification results whereas BLSMOTE resulted in lower sensitivity compared to the baseline.

Finally, results on the large and extremely imbalanced dataset of breast cancer are presented in Table 6.7. NB-Rec achieved the second highest sensitivity, which was lower than kmUnder but significantly higher than the other methods. However, essentially higher specificity, G-mean and F1-score of NB-Rec indicate that the method had a better trade-off than kmUnder. NB-Rec showed high specificity and the highest G-mean

Table 6.7: Results on Breast cancer

method	sensitivity	specificity	G-mean	F1-score
baseline	29.57	**99.98**	54.37	44.72
NB-Rec	74.73	93.49	**83.59**	12.03
SMOTE	45.16	99.75	67.12	**48.55**
BLSMOTE	33.33	99.89	57.70	44.13
DBSMOTE	36.02	99.84	59.97	44.37
kmUnder	**93.01**	40.27	61.20	1.86

of 83.59%. Low F1-score of NB-Rec was due to the bias caused by high class imbalance as discussed above.

6.6 Conclusions

It has been shown that some of the most promising undersampling methods presented in this thesis, namely BoostOBU and NB-Rec, were successfully applied to the medical problems. The predictive diagnostics of diseases including life-threatening and highly-affected neurological diseases with imbalanced records were demonstrated. Consistent results of both methods with previous experiments were achieved. BoostOBU and NB-Rec provided high sensitivity and often with favourable trade-offs with specificity. On nearly all datasets, their sensitivity was the highest among many methods while comparable G-mean was obtained. This can be attributed to the following advantages. First, the resampling amount of our undersampling methods is independent of class imbalance and based on the amount of class overlap. Second, BoostOBU and NB-Rec specifically addresses the problem of class overlap, which often causes errors in classification. Furthermore, both methods employ adaptive parameters and do not need any parameter settings. These enable generalisation of the framework across any medical datasets. Above all, these results suggest a successful application of our overlap-based methods in an important real-world domain. It should be noted that in this work, we have assumed the highest sensitivity was preferred even with specificity being among the lowest as compared to other methods. This would be suitable in a medical-related problem where misidentifying positive cases comes at an unacceptably high cost. However, in a more compromised situation, other overlap-based undersampling methods presented in this thesis, which offer various trade-offs between sensitivity and specificity, could be considered. Moreover, to allow wider applicability on real-world medical problems, a framework for multi-class datasets should be developed. Other application domains may also be explored in the future.

Chapter 7

Conclusions and Future Work

This chapter summarises the main findings resulting from this body of work.

In this thesis, it has been shown that:

- Existing methods for handling classification of imbalanced datasets focus mainly on skewed class distributions and borderline instances. Some aim to rebalance class distributions while some others address the issue of instances overlapping near the class boundary.

- Class overlap highly affects classification of imbalanced datasets, and it also influences the impact of class imbalance. That is, the effect of class imbalance depends on the presence and the amount of class overlap. However, relatively few solutions deal with instances in the entire overlapping region.

- Class overlap often affects the minority class more than the majority class. Thus, by accurately identifying and removing majority class instances from the overlapping region, high sensitivity and good trade-offs between higher sensitivity and lower specificity can be achieved. This approach proved to significantly improve classification performance on imbalanced datasets and outperformed some state-of-the-art class distribution-based methods.

7.1 Summary

This thesis has provided a critical review of literature on classification of imbalanced datasets including well-established and state-of-the-art solutions. For an in-depth discussion, solutions were categorised into class distribution-based focus and class overlap-based focus to allow comparisons. Other methods employing emerging techniques were also discussed. Moreover, a comprehensive study on the impact of class overlap on imbalanced dataset classification was carried out. This was presented through an extensive experiment at the full scale of class overlap and an extreme range of class imbalance. Results clearly showed that class overlap had a higher impact on the learning algorithm's performance than class imbalance.

The importance of the class overlap problem was highlighted by findings from the literature as well as the experimental results. This motivated us to develop new solutions to address the issue of class imbalance in imbalanced datasets. Unlike most existing methods, we aimed at removing the presence of class overlap by means of undersampling. This was achieved by eliminating majority class instances from such a region considering that the majority class was less affected by class overlap than the minority class. Following this idea, several methods were designed with the challenging task of identifying overlapped instances.

The new approaches presented in this thesis are OBU [18] along with its extensions – AdaOBU and BoostOBU [131], and the novel NB-based methods [1]. The OBU-based methods search for overlapped negative instances based on global clustering on the dataset whereas the NB-based methods perform the search locally using kNN. OBU employs a soft clustering algorithm to identify instances with uncertainty in membership degrees, which indicate that the instances are likely to be in the overlapping region. Experimental results showed that significant improvement in sensitivity with relatively small trade-offs with specificity was achieved using OBU. However, the universal setting of the elimination threshold caused some variations in the results. AdaOBU was then introduced to address this issue. AdaOBU extends OBU with an adaptive elimination threshold, which was proved to enable the amount of eliminated majority class instances to be proportional to the class overlap degree. Results suggest that this helped reduce the problem of insufficient or excessive elimination. Moreover, the adaptive threshold replaces the need for fine tuning and enables generalisation of the method across various datasets. BoostOBU is a hybrid method developed to increase the accuracy in detecting overlapped instances. It incorporates an oversampling method to emphasise the minority class boundary in order to enhance the performance of the clustering algorithm. The more accurate removal of overlapped instances of BoostOBU is evidenced by the statistically significant improvements in all metrics over OBU and AdaOBU. Furthermore, the method showed better performance than other well-established and state-of-the-art methods including those using hybrid and ensemble techniques.

The NB-based approach employs a neighbourhood searching algorithm and has four variants with different criteria to determine instances for elimination. All of the four variants achieved the highest sensitivity and competitive G-mean when comparing against other well-known methods. Results also showed clearly that the four methods provided different benefits and trade-offs. This offers users with choices of potential solutions that suit different needs in real-world problems.

Finally, a successful application of the overlap-based methods in the medical domain was demonstrated. A framework for predictive diagnostics of diseases with imbalanced records [131, 147] was presented. Life-threatening and highly-affected neurological diseases such as cancer, heart disease, epilepsy and Parkinson's disease were considered. The highest detection of positive test cases and favourable G-mean were often achieved. Moreover, the benefit of adaptive parameters and no parameter tuning required by the methods enable the generalisation of the framework across any medical datasets.

The more effective performance of the new overlap-based approaches presented in this thesis over other common resampling techniques can be attributed to several advantages as follows. First, their undersampling rates are independent of class imbalance and are proportional to the overlap degree. Second, elimination of instances outside the overlapping region is limited and kept as small as possible. These result in reduced information loss of the majority class while attempting to maximise the visibility of the minority class. Significantly higher sensitivity with relatively lower specificity as a trade-off will be achieved accordingly. Different trade-offs offered by these methods provide more alternatives to real-world users in selecting the best fit solution to the problem.

In conclusion, the achievements on the objectives set out in Chapter 1 are as follows:

- This work has investigated and critically reviewed literature on imbalanced dataset classification and solutions.
- This work has assessed and objectively evaluated the impact of class imbalance and class overlap on a learning algorithm's performance. An experimental framework to assess the scale of impact was created. This included developing a method to measure the degree of class overlap and designing an experiment to compare the two factors.
- In this work, two novel approaches consisting of seven methods to improve the classification of imbalanced datasets have been created. The methods aim at minimising the presence of class overlap while at the same time maximising the visibility of the minority class. Techniques to identify and remove majority class instances in the overlapping region were designed and developed.
- The developed methods have been evaluated across extensive class imbalance and

class overlap degrees using a wide range of data including simulated, real-world and large datasets. The evaluation was also carried out against well-established and state-of-the-art techniques. Results proved competitive performance of our methods with existing ones. Significant improvement in classification especially in terms of sensitivity was achieved.

7.2 Limitations and Future Work

The novel methods presented in this thesis proved to offer competitive solutions for handling class imbalance and class overlap with state-of-the-art methods. However, this work can be further improved and extended to overcome some of the limitations as follows.

- In the OBU-based methods, namely OBU, AdaOBU and BoostOBU, it is assumed that the positive and negative classes could be roughly represented by two distinct clusters. With this assumption, the problem of small disjuncts or within-cluster variation has not been considered. To address this, a method to find the optimal number of clusters may be utilised. One of the most commonly-used methods is the elbow method, in which the sums of squared distances at various numbers of clusters are calculated and graphed. The optimal number of clusters is determined from when adding another cluster does not further improve the result. The elbow method has been used largely with k-means clustering [164,165], which has virtually identical objective functions as the soft clustering algorithm used in OBU. Thus, the elbow method can be used with k-means to find the optimal cluster number prior to performing soft clustering. Alternatively, hierarchical clustering or other less time-consuming adaptive techniques [166,167] may be employed.

- The self-adaptive elimination threshold of AdaOBU and BoostOBU may be improved by also factoring in other data characteristics such as class density and local data density, which could be obtained using such techniques proposed in [136]. Alternatively, a density-based k-means algorithm [168–170] may be modified to serve the purpose. Moreover, since the performance of BoostOBU could be highly dependant on how BLSMOTE performs, other oversampling methods that could provide better outcomes may be explored. Another limitation of the OBU-based methods is the dependency on FCM, which is a distance-based technique. The well-known curse of dimensionality due to the use of a distance-based algorithm could affect the performance of the method on high-dimensional datasets. This issue may be addressed with other improved soft clustering algorithms that showed less dependency on similarity measure such as ones proposed in [137,138]. The technique used in [171], which utilises kernel fuzzy C-means (KFCM) and a local density adaptive diffusion maps (LDM), is also interesting. While the Euclidean distance was replaced with kernel methods in KFCM, LDM could provide reliable similarity description and dimensionality reduction. This would address the sensitivity issue in both data dimensionality and noise of FCM. Alternatively, simply projecting data onto a lower-dimensional space using a technique such as Principle Components Analysis may be considered.

- In the NB-based methods, Euclidean distance is used in determining neighbouring instances. It was shown that the methods were more robust on simulated datasets, whose positive and negative classes were uniformly distributed. Thus, possible improvement to the NB-based methods is incorporating data density into the neighbourhood searching criteria. This can be achived using a modified kNN algorithm that considers both distance-based and density-based affinity measures [172] or an adaptive kNN algorithm based on local density and distribution proposed in [60]. Alternatively, an adaptive k based on relative local density, where a higher k is used when the surrounding minority class density is lower than that of the majority class, may be designed. This will also potentially help diminish the effect of under-representation of minority class instances in the overlapping region.

- This research work is limited to binary-class datasets. Extending the methods to handle multi-class datasets where the minority classes are positive subclasses will make them generalised across a wider range of problems. The OBU-based methods can be extended as ensemble-based methods using binarisation techniques [142]. This will be done by first obtaining binary-class subsets of the original multi-class datasets using the one-vs-all scheme. Then, the OBU-based methods can be straightly applied to each subset of the training data. Weak learners as a result of several training sets would then be used to produce the final outcome based on a common technique such as majority voting. Alternatively, modifications of the OBU-based methods to handle imbalanced multi-class datasets are also achievable. Since FCM is applicable to multi-class problems, the remaining task is to re-design the elimination criteria of the methods. This will involve a modification to consider several membership degrees instead of two degrees of each instance. For example, removing uncertain negative instances whose $max(\mu_{pos_1}, \mu_{pos_2}, ..., \mu_{pos_{i-1}}) - max(\mu_{neg_1}, \mu_{neg_2}, ..., \mu_{neg_{j-1}}) \geq \mu_{th}$, where i and j are the number of positive and negative classes, respectively, could be an initial idea for further investigation of the problem. The elimination threshold, μ_{th}, needs to be empirically set to ensure proper elimination; otherwise, its adaptive form can be modified to suit multi-class problems accordingly. On the other hand, the NB-based methods can potentially be applied to handle multi-class datasets. This is because in the neighbourhood search and the elimination criteria, an instance will be considered as a positive or negative instance regardless of the subclass it belongs to. Performance evaluation of the methods against other existing ones on multi-class problems need to be carried out. However, when not all positive subclasses are of equal importance, their costs may be taken into account so that proper removal of positive instances belonging to a less important subclass is allowed.

- Finally, to expand knowledge in the context of the class overlap problem in imbalanced data classification, the use of emerging techniques is encouraged. These are such as deep learning algorithms and evolutionary algorithms, which are self-learning and capable of providing optimal results. The use of such algorithms have been widely proposed to address the class imbalance problem [87, 89, 91] whereas class overlap was rarely discussed.

Bibliography

[1] Vuttipittayamongkol P, Elyan E. Neighbourhood-based Undersampling Approach for Handling Imbalanced and Overlapped Data. Information Sciences. 2020;509:47–70.

[2] Erfani SM, Rajasegarar S, Karunasekera S, Leckie C. High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. Pattern Recognition. 2016;58:121–134.

[3] Santucci V, Milani A, Caraffini F. An Optimisation-Driven Prediction Method for Automated Diagnosis and Prognosis. Mathematics. 2019;7(11):1051.

[4] Qi CR, Su H, Niessner M, Dai A, Yan M, Guibas LJ. Volumetric and Multi-View CNNs for Object Classification on 3D Data. In: IEEE Conf. Comput. Vision Pattern Recognit. IEEE; 2016. .

[5] Lin W, Wu Z, Lin L, Wen A, Li J. An ensemble random forest algorithm for insurance big data analysis. Ieee access. 2017;5:16568–16575.

[6] López V, Fernández A, García S, Palade V, Herrera F. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Information Sciences. 2013;250:113–141.

[7] García V, Mollineda RA, Sánchez JS. On the k-NN performance in a challenging scenario of imbalance and overlapping. Pattern Analysis and Applications. 2008;11(3-4):269–280.

[8] Sun H, Wang S. Measuring the component overlapping in the Gaussian mixture model. Data mining and knowledge discovery. 2011;23(3):479–502.

[9] Lee HK, Kim SB. An Overlap-Sensitive Margin Classifier for Imbalanced and Overlapping Data. Expert Systems with Applications. 2018;.

[10] Xiong H, Wu J, Liu L. Classification with ClassOverlapping: A Systematic Study. In: Proceedings of the 1st International Conference on E-Business Intelligence (ICEBI2010). Atlantis Press; 2010. .

[11] Haixiang G, Yijing L, Shang J, Mingyun G, Yuanyue H, Bing G. Learning from class-imbalanced data: Review of methods and applications. Expert Systems with Applications. 2017;73:220–239.

[12] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research. 2002;16:321–357.

[13] Das S, Datta S, Chaudhuri BB. Handling data irregularities in classification: Foundations, trends, and future challenges. Pattern Recognition. 2018;.

[14] Lin WC, Tsai CF, Hu YH, Jhang JS. Clustering-based undersampling in class-imbalanced data. Information Sciences. 2017;409:17–26.

[15] Batista GE, Prati RC, Monard MC. Balancing strategies and class overlapping. In: International Symposium on Intelligent Data Analysis. Springer; 2005. p. 24–35.

[16] Visa S, Ralescu A. Learning imbalanced and overlapping classes using fuzzy sets. In: Proceedings of the ICML. vol. 3; 2003. .

[17] Stefanowski J. Overlapping, rare examples and class decomposition in learning classifiers from imbalanced data. In: Emerging paradigms in machine learning. Springer; 2013. p. 277–306.

[18] Vuttipittayamongkol P, Elyan E, Petrovksi A, Jayne C. Overlap-based undersampling for improving imbalanced data classification. In: International Conference on Intelligent Data Engineering and Automated Learning. Springer; 2018. p. 689–697.

[19] Cover T, Hart P. Nearest neighbor pattern classification. IEEE transactions on information theory. 1967;13(1):21–27.

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

[21] Prati RC, Batista GE, Monard MC. Class imbalances versus class overlapping: an analysis of a learning system behavior. In: Mexican international conference on artificial intelligence. Springer; 2004. p. 312–321.

[22] Tax DM, Duin RP. Support vector data description. Machine learning. 2004;54(1):45–66.

[23] Lee JS. AUC4. 5: AUC-based C4. 5 decision tree algorithm for imbalanced data classification. IEEE Access. 2019;7:106034–106042.

[24] Kotu V, Deshpande B. Chapter 4 - Classification. In: Kotu V, Deshpande B, editors. Data Science (Second Edition). second edition ed. Morgan Kaufmann; 2019. p. 65 – 163.

[25] Kim K, Hong Js. A hybrid decision tree algorithm for mixed numeric and categorical data in regression analysis. Pattern Recognition Letters. 2017;98:39–45.

[26] Breiman L. Random forests. Machine learning. 2001;45(1):5–32.

[27] Breiman L. Bagging predictors. Machine learning. 1996;24(2):123–140.

[28] Fawagreh K, Gaber MM, Elyan E. Random forests: from early developments to recent advancements. Systems Science & Control Engineering: An Open Access Journal. 2014;2(1):602–609.

[29] Elyan E, Gaber MM. A fine-grained random forests using class decomposition: an application to medical diagnosis. Neural computing and applications. 2016;27(8):2279–2288.

[30] Vapnik V. Support vector estimation of functions. Statistical Learning Theory. 1998;p. 375–570.

[31] Nisbet R, Miner G, Yale K. Chapter 8 - Advanced Algorithms for Data Mining. In: Nisbet R, Miner G, Yale K, editors. Handbook of Statistical Analysis and Data Mining Applications (Second Edition). second edition ed. Boston: Academic Press; 2018. p. 149 – 167.

[32] Abu Alfeilat HA, Hassanat AB, Lasassmeh O, Tarawneh AS, Alhasanat MB, Eyal Salman HS, et al. Effects of distance measure choice on K-Nearest neighbor classifier performance: A review. Big data. 2019;7(4):221–248.

[33] Maillo J, Ramírez S, Triguero I, Herrera F. kNN-IS: An Iterative Spark-based design of the k-Nearest Neighbors classifier for big data. Knowledge-Based Systems. 2017;117:3–15.

[34] Bekkar M, Djemaa HK, Alitouche TA. Evaluation measures for models assessment over imbalanced data sets. J Inf Eng Appl. 2013;3(10).

[35] Jeni LA, Cohn JF, De La Torre F. Facing Imbalanced Data–Recommendations for the Use of Performance Metrics. In: 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction. IEEE; 2013. p. 245–251.

[36] Hossin M, Sulaiman M. A review on evaluation metrics for data classification evaluations. International Journal of Data Mining & Knowledge Management Process. 2015;5(2):1.

[37] Gu Q, Zhu L, Cai Z. Evaluation measures of the classification performance of imbalanced data sets. In: International symposium on intelligence computation and applications. Springer; 2009. p. 461–471.

[38] Rivera WA, Xanthopoulos P. A priori synthetic over-sampling methods for increasing classification sensitivity in imbalanced data sets. Expert Systems with Applications. 2016;66:124–135.

[39] Bunkhumpornpat C, Sinapiromsaran K, Lursinsap C. DBSMOTE: density-based synthetic minority over-sampling technique. Applied Intelligence. 2012;36(3):664–684.

[40] Brodersen KH, Ong CS, Stephan KE, Buhmann JM. The balanced accuracy and its posterior distribution. In: 2010 20th International Conference on Pattern Recognition. IEEE; 2010. p. 3121–3124.

[41] Amin A, Anwar S, Adnan A, Nawaz M, Howard N, Qadir J, et al. Comparing oversampling techniques to handle the class imbalance problem: a customer churn prediction case study. IEEE Access. 2016;4:7940–7957.

[42] Sarafianos N, Xu X, Kakadiaris IA. Deep imbalanced attribute classification using visual attention aggregation. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. p. 680–697.

[43] Devi D, Purkayastha B, et al. Redundancy-driven modified Tomek-link based undersampling: A solution to class imbalance. Pattern Recognition Letters. 2017;93:3–12.

[44] Huang C, Li Y, Change Loy C, Tang X. Learning deep representation for imbalanced classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 5375–5384.

[45] Collell G, Prelec D, Patil KR. A simple plug-in bagging ensemble based on threshold-moving for classifying binary and multiclass imbalanced data. Neurocomputing. 2018;275:330–340.

[46] Luque A, Carrasco A, Martín A, de las Heras A. The impact of class imbalance in classification performance metrics based on the binary confusion matrix. Pattern Recognition. 2019;p. 6829.

[47] Barandela R, Valdovinos RM, Sánchez JS. New applications of ensembles of classifiers. Pattern Analysis & Applications. 2003;6(3):245–256.

[48] Bunkhumpornpat C, Sinapiromsaran K. DBMUTE: density-based majority under-sampling technique. Knowledge and Information Systems. 2017;50(3):827–850.

[49] Lobo JM, Jiménez-Valverde A, Real R. AUC: a misleading measure of the performance of predictive distribution models. Global ecology and Biogeography. 2008;17(2):145–151.

[50] Huang J, Ling CX. Using AUC and accuracy in evaluating learning algorithms. IEEE Transactions on knowledge and Data Engineering. 2005;17(3):299–310.

[51] Adams NM, Hand DJ. Comparing classifiers when the misallocation costs are uncertain. Pattern Recognition. 1999;32(7):1139–1147.

[52] Yen SJ, Lee YS. Cluster-based under-sampling approaches for imbalanced data distributions. Expert Systems with Applications. 2009;36(3):5718–5727.

[53] Ng WW, Hu J, Yeung DS, Yin S, Roli F. Diversified sensitivity-based undersampling for imbalance classification problems. IEEE transactions on cybernetics. 2015;45(11):2402–2412.

[54] Ofek N, Rokach L, Stern R, Shabtai A. Fast-CBUS: A fast clustering-based undersampling method for addressing the class imbalance problem. Neurocomputing. 2017;243:88–102.

[55] Branco P, Torgo L, Ribeiro RP. A survey of predictive modeling on imbalanced domains. ACM Computing Surveys (CSUR). 2016;49(2):31.

[56] Dal Pozzolo A, Caelen O, Bontempi G. When is undersampling effective in unbalanced classification tasks? In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer; 2015. p. 200–215.

[57] Douzas G, Bacao F, Last F. Improving Imbalanced Learning Through a Heuristic Oversampling Method Based on K-Means and SMOTE. Information Sciences. 2018;.

[58] Bunkhumpornpat C, Sinapiromsaran K, Lursinsap C. Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In: Pacific-Asia conference on knowledge discovery and data mining. Springer; 2009. p. 475–482.

[59] Han H, Wang WY, Mao BH. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In: International Conference on Intelligent Computing. Springer; 2005. p. 878–887.

[60] Mullick SS, Datta S, Das S. Adaptive Learning-Based k-Nearest Neighbor Classifiers With Resilience to Class Imbalance. IEEE transactions on neural networks and learning systems. 2018;29(11):5713–5725.

[61] de Morais RF, Vasconcelos GC. Boosting the performance of over-sampling algorithms through under-sampling the minority class. Neurocomputing. 2019;343:3–18.

[62] Chetchotsak D, Pattanapairoj S, Arnonkijpanich B. Integrating new data balancing technique with committee networks for imbalanced data: GRSOM approach. Cognitive neurodynamics. 2015;9(6):627–638.

[63] Raghuwanshi BS, Shukla S. SMOTE based class-specific extreme learning machine for imbalanced learning. Knowledge-Based Systems. 2020;187:104814.

[64] Raghuwanshi BS, Shukla S. Underbagging based reduced kernelized weighted extreme learning machine for class imbalance learning. Engineering Applications of Artificial Intelligence. 2018;74:252–270.

[65] Raghuwanshi BS, Shukla S. Class imbalance learning using UnderBagging based kernelized extreme learning machine. Neurocomputing. 2019;329:172–187.

[66] Wallace BC, Small K, Brodley CE, Trikalinos TA. Class imbalance, redux. In: 2011 IEEE 11th international conference on data mining. IEEE; 2011. p. 754–763.

[67] Seiffert C, Khoshgoftaar TM, Van Hulse J, Napolitano A. RUSBoost: A hybrid approach to alleviating class imbalance. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans. 2010;40(1):185–197.

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

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

[70] Chawla NV, Lazarevic A, Hall LO, Bowyer KW. SMOTEBoost: Improving prediction of the minority class in boosting. In: European Conference on Principles of Data Mining and Knowledge Discovery. Springer; 2003. p. 107–119.

[71] Tahir MA, Kittler J, Yan F. Inverse random under sampling for class imbalance problem and its application to multi-label classification. Pattern Recognition. 2012;45(10):3738–3750.

[72] Barua S, Islam MM, Yao X, Murase K. MWMOTE–majority weighted minority oversampling technique for imbalanced data set learning. IEEE Transactions on Knowledge and Data Engineering. 2014;26(2):405–425.

[73] Nekooeimehr I, Lai-Yuen SK. Adaptive semi-unsupervised weighted oversampling (A-SUWO) for imbalanced datasets. Expert Systems with Applications. 2016;46:405–416.

[74] He H, Bai Y, Garcia EA, Li S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on. IEEE; 2008. p. 1322–1328.

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

[76] Zhang X, Li Y. A positive-biased nearest neighbour algorithm for imbalanced classification. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer; 2013. p. 293–304.

[77] Sáez JA, Luengo J, Stefanowski J, Herrera F. SMOTE–IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. Information Sciences. 2015;291:184–203.

[78] Ertekin Ş. Adaptive oversampling for imbalanced data classification. In: Information Sciences and Systems 2013. Springer; 2013. p. 261–269.

[79] Jian C, Gao J, Ao Y. A new sampling method for classifying imbalanced data based on support vector machine ensemble. Neurocomputing. 2016;193:115–122.

[80] Nanni L, Fantozzi C, Lazzarini N. Coupling different methods for overcoming the class imbalance problem. Neurocomputing. 2015;158:48–61.

[81] Fernandes ER, de Carvalho AC. Evolutionary inversion of class distribution in overlapping areas for multi-class imbalanced learning. Information Sciences. 2019;494:141–154.

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

[83] Beyan C, Fisher R. Classifying imbalanced data sets using similarity based hierarchical decomposition. Pattern Recognition. 2015;48(5):1653–1672.

[84] DAddabbo A, Maglietta R. Parallel selective sampling method for imbalanced and large data classification. Pattern Recognition Letters. 2015;62:61–67.

[85] Díez-Pastor JF, Rodríguez JJ, García-Osorio C, Kuncheva LI. Random balance: ensembles of variable priors classifiers for imbalanced data. Knowledge-Based Systems. 2015;85:96–111.

[86] García S, Herrera F. Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy. Evolutionary computation. 2009;17(3):275–306.

[87] Galar M, Fernández A, Barrenechea E, Herrera F. EUSBoost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling. Pattern Recognition. 2013;46(12):3460–3471.

[88] García S, Derrac J, Triguero I, Carmona CJ, Herrera F. Evolutionary-based selection of generalized instances for imbalanced classification. Knowledge-Based Systems. 2012;25(1):3–12.

[89] Vluymans S, Triguero I, Cornelis C, Saeys Y. EPRENNID: An evolutionary prototype reduction based ensemble for nearest neighbor classification of imbalanced data. Neurocomputing. 2016;216:596–610.

[90] Douzas G, Bacao F. Effective data generation for imbalanced learning using conditional generative adversarial networks. Expert Systems with applications. 2018;91:464–471.

[91] Ali-Gombe A, Elyan E. MFC-GAN: Class-imbalanced dataset classification using Multiple Fake Class Generative Adversarial Network. Neurocomputing. 2019;361:212–221.

[92] Wang S, Liu W, Wu J, Cao L, Meng Q, Kennedy PJ. Training deep neural networks on imbalanced data sets. In: 2016 International Joint Conference on Neural Networks (IJCNN). IEEE; 2016. p. 4368–4374.

[93] Khan SH, Hayat M, Bennamoun M, Sohel FA, Togneri R. Cost-sensitive learning of deep feature representations from imbalanced data. IEEE transactions on neural networks and learning systems. 2018;29(8):3573–3587.

[94] Chung YA, Lin HT, Yang SW. Cost-aware pre-training for multiclass cost-sensitive deep learning. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI16, AAAI Press; 2015. p. 1411–1417.

[95] Weiss GM, Provost F. Learning when training data are costly: The effect of class distribution on tree induction. Journal of artificial intelligence research. 2003;19:315–354.

[96] Sun J, Lang J, Fujita H, Li H. Imbalanced enterprise credit evaluation with DTE-SBD: decision tree ensemble based on SMOTE and bagging with differentiated sampling rates. Information Sciences. 2018;425:76–91.

[97] Ijaz M, Alfian G, Syafrudin M, Rhee J. Hybrid Prediction Model for Type 2 Diabetes and Hypertension Using DBSCAN-Based Outlier Detection, Synthetic Minority Over Sampling Technique (SMOTE), and Random Forest. Applied Sciences. 2018;8(8):1325.

[98] Wang S, Wang D, Li J, Huang T, Cai YD. Identification and analysis of the cleavage site in a signal peptide using SMOTE, dagging, and feature selection methods. Molecular omics. 2018;14(1):64–73.

[99] Douzas G, Bacao F. Geometric SMOTE: Effective oversampling for imbalanced learning through a geometric extension of SMOTE. arXiv preprint arXiv:170907377. 2017;.

[100] Ester M, Kriegel HP, Sander J, Xu X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: Proc. of 2nd International Conference on Knowledge Discovery and Data Mining; 1996. p. 226–231.

[101] Koziarski M, Krawczyk B, Woźniak M. Radial-Based oversampling for noisy imbalanced data classification. Neurocomputing. 2019;343:19–33.

[102] Raghuwanshi BS, Shukla S. Class-specific kernelized extreme learning machine for binary class imbalance learning. Applied Soft Computing. 2018;73:1026–1038.

[103] Raghuwanshi BS, Shukla S. Class-specific extreme learning machine for handling binary class imbalance problem. Neural Networks. 2018;105:206–217.

[104] Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences. 1997;55(1):119–139.

[105] Wilson DL. Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man, and Cybernetics. 1972;(3):408–421.

[106] Hoi CH, Chan CH, Huang K, Lyu MR, King I. Biased support vector machine for relevance feedback in image retrieval. In: 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541). vol. 4. IEEE; 2004. p. 3189–3194.

[107] Lin CF, Wang SD. Fuzzy support vector machines. IEEE transactions on neural networks. 2002;13(2):464–471.

[108] Wang S, Yao X. Relationships between diversity of classification ensembles and single-class performance measures. IEEE Transactions on Knowledge and Data Engineering. 2013;25(1):206–219.

[109] Zhou ZH, Liu XY. Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Transactions on Knowledge & Data Engineering. 2006;(1):63–77.

[110] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial nets. In: Advances in neural information processing systems; 2014. p. 2672–2680.

[111] Mirza M, Osindero S. Conditional generative adversarial nets. arXiv preprint arXiv:14111784. 2014;.

[112] Shaikhina T, Khovanova NA. Handling limited datasets with neural networks in medical applications: A small-data approach. Artificial intelligence in medicine. 2017;75:51–63.

[113] Feng S, Zhou H, Dong H. Using deep neural network with small dataset to predict material defects. Materials & Design. 2019;162:300–310.

[114] Odena A, Olah C, Shlens J. Conditional image synthesis with auxiliary classifier gans. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org; 2017. p. 2642–2651.

[115] Ali-Gombe A, Elyan E, Savoye Y, Jayne C. Few-shot Classifier GAN. In: 2018 International Joint Conference on Neural Networks (IJCNN). IEEE; 2018. p. 1–8.

[116] Hart P. The condensed nearest neighbor rule (Corresp.). IEEE transactions on information theory. 1968;14(3):515–516.

[117] Tomek I. Two modifications of CNN. IEEE Trans Systems, Man and Cybernetics. 1976;6:769–772.

[118] Zhang N, Paluri M, Ranzato M, Darrell T, Bourdev L. Panda: Pose aligned networks for deep attribute modeling. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2014. p. 1637–1644.

[119] Schroff F, Kalenichenko D, Philbin J. Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. p. 815–823.

[120] Liu Z, Luo P, Wang X, Tang X. Deep learning face attributes in the wild. In: Proceedings of the IEEE international conference on computer vision; 2015. p. 3730–3738.

[121] Girshick R. Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision; 2015. p. 1440–1448.

[122] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. p. 1–9.

[123] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 770–778.

[124] Liu XY, Wu J, Zhou ZH. Exploratory undersampling for class-imbalance learning. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). 2009;39(2):539–550.

[125] Bezdek JC, Ehrlich R, Full W. FCM: The fuzzy c-means clustering algorithm. Computers & Geosciences. 1984;10(2-3):191–203.

[126] Ghosh S, Dubey SK. Comparative analysis of k-means and fuzzy c-means algorithms. International Journal of Advanced Computer Science and Applications. 2013;4(4).

[127] Liu H, Wu J, Liu T, Tao D, Fu Y. Spectral ensemble clustering via weighted k-means: Theoretical and practical evidence. IEEE transactions on knowledge and data engineering. 2017;29(5):1129–1143.

[128] Elyan E, Gaber MM. A genetic algorithm approach to optimising random forests applied to class engineered data. Information sciences. 2017;384:220–234.

[129] Dua D, Graff C. UCI Machine Learning Repository; 2017. http://archive.ics.uci.edu/ml.

[130] Alcalá-Fdez J, Fernández A, Luengo J, Derrac J, García S, Sánchez L, et al. Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. Journal of Multiple-Valued Logic & Soft Computing. 2011;17.

[131] Vuttipittayamongkol P, Elyan E. Improved Overlap-based Undersampling for Imbalanced Dataset Classification with Application to Epilepsy and Parkinsons Disease. International journal of neural systems. 2020;30(08):2050043.

[132] Batista GE, Prati RC, Monard MC. A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD explorations newsletter. 2004;6(1):20–29.

[133] LeCun Y, Cortes C. MNIST handwritten digit database. 2010;http://yann.lecun.com/exdb/mnist.

[134] Galar M, Fernandez A, Barrenechea E, Bustince H, Herrera F. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews). 2012;42(4):463–484.

[135] Napierala K, Stefanowski J. Types of minority class examples and their influence on learning classifiers from imbalanced data. J Intell Inf Syst. 2016;46(3):563–597.

[136] Ahmadlou M, Adeli H. Enhanced probabilistic neural network with local decision circles: A robust classifier. Integr Comput-Aid Eng. 2010;17(3):197–210.

[137] Zhou J, Chen L, Chen CP, Zhang Y, Li HX. Fuzzy clustering with the entropy of attribute weights. Neurocomputing. 2016;198:125–134.

[138] Xia H, Zhuang J, Yu D. Novel soft subspace clustering with multi-objective evolutionary approach for high-dimensional data. Pattern Recognit. 2013;46(9):2562–2575.

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

[140] Koziarski M, Wożniak M. CCR: A combined cleaning and resampling algorithm for imbalanced data classification. International Journal of Applied Mathematics and Computer Science. 2017;27(4):727–736.

[141] Krawczyk B, Galar M, Jeleń Ł, Herrera F. Evolutionary undersampling boosting for imbalanced classification of breast cancer malignancy. Applied Soft Computing. 2016;38:714–726.

[142] Galar M, Fernández A, Barrenechea E, Bustince H, Herrera F. 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. 2011;44(8):1761–1776.

[143] Rifkin R, Klautau A. In defense of one-vs-all classification. Journal of machine learning research. 2004;5(Jan):101–141.

[144] He H, Garcia EA. Learning from imbalanced data. IEEE Transactions on Knowledge & Data Engineering. 2008;(9):1263–1284.

[145] Hassanat AB, Abbadi MA, Altarawneh GA, Alhasanat AA. Solving the problem of the K parameter in the KNN classifier using an ensemble learning approach. International Journal of Computer Science and Information Security. 2014;12(8):33–39.

[146] Chaudhari P, Agarwal H, Bhateja V. Data augmentation for cancer classification in oncogenomics: an improved KNN based approach. Evolutionary Intelligence. 2019;p. 1–10.

[147] Vuttipittayamongkol P, Elyan E. Overlap-Based Undersampling Method for Classification of Imbalanced Medical Datasets. In: Artificial Intelligence Applications and Innovations. AIAI 2020. IFIP Advances in Information and Communication Technology. vol. 584. Springer; 2020. p. 358–369.

[148] Bach M, Werner A, Żywiec J, Pluskiewicz W. The study of under-and over-sampling methods utility in analysis of highly imbalanced data on osteoporosis. Information Sciences. 2017;384:174–190.

[149] Yuan X, Xie L, Abouelenien M. A regularized ensemble framework of deep learning for cancer detection from multi-class, imbalanced training data. Pattern Recognition. 2018;77:160–172.

[150] Acharya UR, Oh SL, Hagiwara Y, Tan JH, Adam M, Gertych A, et al. A deep convolutional neural network model to classify heartbeats. Computers in biology and medicine. 2017;89:389–396.

[151] Zhang L, Yang H, Jiang Z. Imbalanced biomedical data classification using self-adaptive multilayer ELM combined with dynamic GAN. Biomedical engineering online. 2018;17(1):181.

[152] Han W, Huang Z, Li S, Jia Y. Distribution-sensitive unbalanced data oversampling method for medical diagnosis. Journal of medical Systems. 2019;43(2):39.

[153] Jiang J, Zhang H, Pi D, Dai C. A novel multi-module neural network system for imbalanced heartbeats classification. Expert Systems with Applications: X. 2019;1:100003.

[154] Sun C, Cui H, Zhou W, Nie W, Wang X, Yuan Q. Epileptic Seizure Detection with EEG Textural Features and Imbalanced Classification Based on EasyEnsemble Learning. International journal of neural systems. 2019;p. 1950021–1950021.

[155] Kalantari A, Kamsin A, Shamshirband S, Gani A, Alinejad-Rokny H, Chronopoulos AT. Computational intelligence approaches for classification of medical data: State-of-the-art, future challenges and research directions. Neurocomputing. 2018;276:2–22.

[156] Fotouhi S, Asadi S, Kattan MW. A comprehensive data level analysis for cancer diagnosis on imbalanced data. Journal of biomedical informatics. 2019;.

[157] Shilaskar S, Ghatol A. Diagnosis system for imbalanced multi-minority medical dataset. Soft Computing. 2019;23(13):4789–4799.

[158] Wan X, Liu J, Cheung WK, Tong T. Learning to improve medical decision making from imbalanced data without a priori cost. BMC medical informatics and decision making. 2014;14(1):111.

[159] Bae SH, Yoon KJ. Polyp detection via imbalanced learning and discriminative feature learning. IEEE transactions on medical imaging. 2015;34(11):2379–2393.

[160] Krawczyk B, Schaefer G, Woźniak M. A hybrid cost-sensitive ensemble for imbalanced breast thermogram classification. Artificial intelligence in medicine. 2015;65(3):219–227.

[161] Havaei M, Davy A, Warde-Farley D, Biard A, Courville A, Bengio Y, et al. Brain tumor segmentation with deep neural networks. Medical image analysis. 2017;35:18–31.

[162] Acharya UR, Oh SL, Hagiwara Y, Tan JH, Adeli H. Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals. Computers in biology and medicine. 2018;100:270–278.

[163] Pereira CR, Pereira DR, Weber SA, Hook C, de Albuquerque VHC, Papa JP. A survey on computer-assisted Parkinson's disease diagnosis. Artificial intelligence in medicine. 2019;95:48–63.

[164] Bholowalia P, Kumar A. EBK-means: A clustering technique based on elbow method and k-means in WSN. International Journal of Computer Applications. 2014;105(9).

[165] Marutho D, Handaka SH, Wijaya E, et al. The determination of cluster number at k-mean using elbow method and purity evaluation on headline news. In: 2018 International Seminar on Application for Technology of Information and Communication. IEEE; 2018. p. 533–538.

[166] Wang XD, Chen RC, Yan F, Zeng ZQ, Hong CQ. Fast adaptive K-means subspace clustering for high-dimensional data. IEEE Access. 2019;7:42639–42651.

[167] Vincent O, Makinde A, Salako O, Oluwafemi O. A self-adaptive k-means classifier for business incentive in a fashion design environment. Applied computing and informatics. 2018;14(1):88–97.

[168] Kumar KM, Reddy ARM. An efficient k-means clustering filtering algorithm using density based initial cluster centers. Information Sciences. 2017;418:286–301.

[169] Bai L, Cheng X, Liang J, Shen H, Guo Y. Fast density clustering strategies based on the k-means algorithm. Pattern Recognition. 2017;71:375–386.

[170] Zhang G, Zhang C, Zhang H. Improved K-means algorithm based on density Canopy. Knowledge-Based Systems. 2018;145:289–297.

[171] Jia B, Yu B, Wu Q, Yang X, Wei C, Law R, et al. Hybrid local diffusion maps and improved cuckoo search algorithm for multiclass dataset analysis. Neurocomputing. 2016;189:106–116.

[172] Kang M, Ramaswami GK, Hodkiewicz M, Cripps E, Kim JM, Pecht M. A sequential K-nearest neighbor classification approach for data-driven fault diagnosis using distance-and density-based affinity measures. In: International Conference on Data Mining and Big Data. Springer; 2016. p. 253–261.

Appendix A

Supplementary Results

A.1 Full results for Experiment II in the Evaluation of AdaOBU and BoostOBU

Table A.1: Sensitivity results for the full 66 datasets

Dataset	Sensitivity Value/Rank																			
	Baseline		SMOTE		BLSMOTE		kmUnder		SMOTE-ENN		SMTBagging		RUSBoost		OBU		AdaOBU		BoostOBU	
Glass1	25.00	9	37.50	7	62.50	4	62.50	4	12.50	10	37.50	7	62.50	4	75.00	2	87.50	1	75.00	2
Ecoli0vs1	0.00	9	50.00	6	50.00	6	66.67	5	0.00	9	83.33	3	83.33	4	100.00	1	100.00	1	33.33	8
Wisconsin	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1
Pima	100.00	1	100.00	1	0.00	9	100.00	1	100.00	1	100.00	1	0.00	9	100.00	1	100.00	1	100.00	1
Iris0	80.00	8	86.67	5	80.00	8	100.00	1	66.67	10	86.67	5	86.67	3	100.00	1	86.67	3	86.67	5
Glass0	80.00	8	90.00	3	60.00	10	90.00	3	80.00	8	90.00	3	90.00	3	90.00	3	100.00	1	100.00	1
Yeast1	42.86	9	71.43	6	42.86	9	100.00	1	71.43	6	71.43	6	85.71	4	100.00	1	85.71	4	100.00	1
Haberman	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1
Vehicle2	71.43	7	71.43	7	85.71	4	78.57	5	50.00	10	71.43	7	78.57	5	100.00	1	100.00	1	100.00	1
Vehicle1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	90.00	8	100.00	1	100.00	1	90.00	8	90.00	8
Vehicle3	0.00	7	0.00	7	33.33	3	33.33	6	33.33	3	33.33	3	0.00	7	100.00	1	66.67	2	0.00	7
Glass0123vs456	100.00	1	100.00	1	100.00	1	100.00	1	0.00	9	100.00	1	100.00	1	0.00	9	100.00	1	100.00	1
Vehicle0	66.67	8	60.00	9	80.00	5	73.33	7	60.00	9	80.00	5	86.67	4	100.00	1	100.00	1	100.00	1
Ecoli1	0.00	10	66.67	4	66.67	4	66.67	2	33.33	8	33.33	8	66.67	2	66.67	4	100.00	1	66.67	4
Newthyroid1	100.00	1	100.00	1	100.00	1	100.00	1	50.00	10	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1
Newthyroid2	0.00	2	0.00	2	0.00	2	100.00	1	0.00	2	0.00	2	0.00	2	0.00	2	0.00	2	0.00	2
Ecoli2	80.00	4	100.00	1	80.00	4	60.00	9	60.00	9	80.00	4	100.00	1	80.00	4	100.00	1	80.00	4
Segment0	18.75	10	50.00	6	43.75	8	56.25	4	31.25	9	56.25	4	50.00	6	75.00	2	68.75	3	81.25	1
Glass6	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1
Yeast3	100.00	1	100.00	1	14.29	10	100.00	1	57.14	9	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1
Ecoli3	57.14	7	57.14	7	57.14	7	100.00	1	100.00	1	57.14	7	100.00	1	100.00	1	100.00	1	100.00	1
Pageblocks0	81.98	10	92.79	5	95.50	3	91.89	6	93.69	4	91.89	7	90.09	8	87.39	9	100.00	1	100.00	1
Yeast2vs4	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1
Ecoli067vs35	50.00	4	50.00	4	50.00	4	50.00	4	50.00	4	50.00	4	75.00	1	75.00	1	75.00	1	50.00	4
Glass015vs2	0.00	9	33.33	4	33.33	4	66.67	1	0.00	9	33.33	4	33.33	4	66.67	1	66.67	1	33.33	4
Yeast02579vs368	73.68	9	78.95	6	73.68	9	78.95	6	84.21	1	84.21	1	84.21	1	84.21	1	78.95	6	84.21	1
Ecoli046vs5	100.00	1	100.00	1	75.00	7	100.00	1	100.00	1	75.00	7	100.00	1	100.00	1	75.00	7	75.00	7
Ecoli0267vs35	50.00	1	50.00	1	50.00	1	50.00	1	50.00	1	50.00	1	50.00	1	50.00	1	50.00	1	50.00	1
Glass04vs5	100.00	1	100.00	1	0.00	10	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1
Ecoli0346vs5	75.00	1	75.00	1	25.00	10	75.00	1	75.00	1	75.00	1	75.00	1	75.00	1	75.00	1	75.00	1
Yeast05679vs4	58.49	9	73.58	6	81.13	3	73.58	5	47.17	10	60.38	8	71.70	7	88.68	1	77.36	4	84.91	2
Vowel0	89.23	9	96.92	6	89.23	9	100.00	1	98.46	5	95.38	8	95.38	7	100.00	1	100.00	1	100.00	1
Ecoli067vs5	75.00	3	75.00	3	75.00	3	75.00	3	75.00	3	75.00	3	75.00	3	100.00	1	100.00	1	75.00	3
Ecoli0147vs2356	60.00	2	60.00	2	60.00	2	60.00	2	60.00	2	60.00	2	60.00	2	60.00	2	80.00	1	60.00	2
Led7digit02456789vs1	71.43	7	85.71	2	85.71	2	71.43	7	71.43	7	85.71	2	100.00	1	85.71	2	85.71	2	71.43	7
Ecoli01vs5	50.00	1	50.00	1	25.00	10	50.00	1	50.00	1	50.00	1	50.00	1	50.00	1	50.00	1	50.00	1
Glass0146vs2	33.33	8	66.67	6	100.00	1	100.00	1	33.33	8	100.00	1	66.67	6	100.00	1	100.00	1	33.33	8
Glass2	0.00	5	0.00	5	0.00	5	100.00	1	100.00	1	0.00	5	100.00	1	100.00	1	0.00	5	0.00	5
Cleveland0vs4	50.00	2	0.00	6	0.00	6	100.00	1	0.00	6	0.00	6	50.00	2	50.00	2	50.00	2	0.00	6
Ecoli0146vs5	100.00	1	100.00	1	75.00	10	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1
Shuttle0vs4	92.31	8	94.87	6	92.31	8	97.44	5	71.79	10	94.87	6	100.00	1	100.00	1	100.00	1	100.00	1
Yeast1vs7	41.86	9	88.37	5	81.40	8	95.35	2	25.58	10	88.37	5	95.35	2	95.35	1	93.02	4	88.37	5
Glass4	95.35	4	95.35	4	81.40	9	95.35	7	65.12	10	95.35	4	95.35	7	100.00	1	100.00	1	100.00	1
Ecoli4	38.10	9	71.43	7	76.19	6	85.71	3	21.43	10	69.05	8	78.57	5	85.71	2	90.48	1	83.33	4
Pageblocks13vs2	100.00	1	100.00	1	100.00	1	100.00	1	72.22	10	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1
Abalone0918	97.87	3	97.87	3	97.87	3	100.00	1	97.87	3	97.87	3	97.87	9	100.00	1	97.87	9	97.87	3
Dermatology6	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	50.00	10	100.00	1	100.00	1	100.00	1
Glass016vs5	60.00	8	90.00	1	50.00	9	70.00	7	50.00	9	90.00	1	90.00	1	80.00	4	80.00	4	80.00	4
Shuttle2vs4	41.18	10	74.12	6	78.82	4	71.76	7	51.76	9	71.76	8	75.29	5	82.35	3	85.88	2	87.06	1
Yeast1458vs7	66.67	6	66.67	6	33.33	9	100.00	1	0.00	10	83.33	3	83.33	5	83.33	3	100.00	1	66.67	6
Glass5	0.00	9	33.33	8	83.33	2	66.67	4	0.00	9	66.67	5	100.00	1	50.00	7	83.33	3	66.67	5
Yeast2vs8	0.00	9	33.33	1	33.33	1	33.33	6	0.00	9	33.33	1	16.67	7	33.33	1	16.67	7	33.33	1
Yeast4	70.00	9	90.00	2	100.00	1	90.00	2	40.00	10	80.00	7	90.00	2	80.00	7	90.00	2	90.00	2
Winequalityred4	0.00	10	40.00	4	40.00	4	90.00	1	40.00	4	30.00	8	20.00	9	50.00	2	50.00	2	40.00	4
Yeast1289vs7	75.00	3	75.00	3	0.00	8	50.00	6	0.00	8	75.00	3	100.00	1	0.00	8	100.00	1	25.00	7
Winequalityred8vs6	0.00	10	33.33	8	100.00	1	100.00	1	66.67	7	33.33	8	100.00	1	66.67	4	66.67	4	66.67	4
Ecoli0137vs26	30.00	9	60.00	4	80.00	1	80.00	1	30.00	9	60.00	4	60.00	4	80.00	1	60.00	4	50.00	8
Abalone21vs8	0.00	8	50.00	5	0.00	8	100.00	1	50.00	5	0.00	8	50.00	5	100.00	1	100.00	1	100.00	1
Yeast6	62.50	9	100.00	1	87.50	6	75.00	7	50.00	10	100.00	1	100.00	1	75.00	7	100.00	1	100.00	1
Winequalitywhite3vs7	25.00	5	0.00	7	0.00	7	100.00	1	0.00	7	0.00	7	100.00	1	75.00	3	75.00	3	25.00	5
Winequalityred8vs67	0.00	6	0.00	6	0.00	6	100.00	1	0.00	6	0.00	6	33.33	5	100.00	1	100.00	1	66.67	4
Abalone19vs10111213	0.00	10	16.67	4	16.67	4	33.33	3	16.67	8	16.67	4	16.67	8	50.00	1	50.00	1	16.67	4
Winequalitywhite39vs5	0.00	7	0.00	7	20.00	6	100.00	1	0.00	7	0.00	7	60.00	2	40.00	3	40.00	3	40.00	3
Shuttle2vs5	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1
Winequalityred3vs5	0.00	10	50.00	2	50.00	2	100.00	1	50.00	2	50.00	2	50.00	2	50.00	2	50.00	2	50.00	2
Abalone19	71.43	8	85.71	3	85.71	3	71.43	9	42.86	10	85.71	3	100.00	1	85.71	3	100.00	1	85.71	3

Table A.2: Specificity results for the full 66 datasets

Dataset	Specificity Value/Rank																			
	Baseline		SMOTE		BLSMOTE		kmUnder		SMOTE-ENN		SMTBagging		RUSBoost		OBU		AdaOBU		BoostOBU	
Glass1	99.27	1	91.24	4	86.13	6	84.67	7	98.54	2	91.97	3	81.75	8	67.88	9	61.31	10	87.59	5
Ecoli0vs1	100.00	1	95.29	3	94.20	4	82.97	7	99.28	2	88.65	6	82.85	8	60.63	9	53.02	10	92.03	5
Wisconsin	100.00	1	100.00	1	92.86	10	100.00	1	100.00	1	100.00	1	92.86	9	100.00	1	100.00	1	100.00	1
Pima	100.00	1	98.15	6	96.30	8	92.59	10	100.00	1	96.30	8	96.30	7	100.00	1	100.00	1	100.00	1
Iris0	98.04	1	96.08	2	94.12	4	86.27	10	90.20	7	96.08	2	92.16	5	86.27	9	92.16	5	90.20	7
Glass0	100.00	1	100.00	1	92.86	7	98.21	5	100.00	1	100.00	1	96.43	6	57.14	9	55.36	10	67.86	8
Yeast1	98.33	1	93.33	3	90.00	5	75.00	10	96.67	2	93.33	3	88.33	7	76.67	9	86.67	8	88.33	6
Haberman	100.00	1	98.41	6	98.41	6	100.00	1	100.00	1	98.41	6	96.83	10	100.00	1	100.00	1	98.41	6
Vehicle2	82.14	2	78.57	4	64.29	7	75.00	6	89.29	1	78.57	4	78.57	3	53.57	8	39.29	9	39.29	10
Vehicle1	100.00	1	96.88	6	100.00	1	93.75	7	90.63	9	100.00	1	100.00	1	93.75	7	100.00	1	59.38	10
Vehicle3	100.00	1	94.29	5	94.29	5	57.14	8	97.14	3	94.29	5	100.00	1	31.43	9	28.57	10	97.14	3
Glass0123vs456	97.14	8	100.00	1	100.00	1	91.43	10	100.00	1	100.00	1	100.00	1	100.00	1	97.14	9	100.00	1
Vehicle0	88.89	1	81.48	2	70.37	5	66.67	6	77.78	3	77.78	3	59.26	7	25.93	10	51.85	8	51.85	9
Ecoli1	100.00	1	97.44	3	97.44	3	53.85	8	100.00	1	97.44	3	74.36	6	43.59	9	38.46	10	61.54	7
Newthyroid1	100.00	1	100.00	1	100.00	1	95.00	6	100.00	1	95.00	6	82.50	9	60.00	10	95.00	6	100.00	1
Newthyroid2	100.00	1	100.00	1	100.00	1	53.66	10	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1
Ecoli2	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	94.59	10	100.00	1	100.00	1	100.00	1
Segment0	100.00	1	57.78	6	55.56	8	66.67	3	71.11	2	62.22	4	55.56	7	53.33	9	57.78	5	33.33	10
Glass6	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1
Yeast3	97.22	8	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	88.89	10	94.44	9	100.00	1
Ecoli3	100.00	1	100.00	1	100.00	1	97.22	7	100.00	1	100.00	1	100.00	1	88.89	8	80.56	9	69.44	10
Pageblocks0	98.17	1	94.20	7	91.65	8	95.21	4	96.54	3	94.60	5	94.40	6	98.07	2	23.01	10	36.56	9
Yeast2vs4	100.00	1	100.00	1	100.00	1	94.32	8	100.00	1	100.00	1	87.50	9	97.73	7	78.41	10	100.00	1
Ecoli067vs35	100.00	1	100.00	1	97.50	6	100.00	1	100.00	1	100.00	1	87.50	8	80.00	9	77.50	10	97.50	6
Glass015vs2	100.00	1	96.77	4	80.65	6	54.84	8	100.00	1	100.00	1	90.32	5	16.13	10	29.03	9	64.52	7
Yeast02579vs368	99.45	1	95.58	3	83.98	8	95.03	5	95.58	3	96.13	2	92.27	7	64.64	10	93.92	6	66.85	9
Ecoli046vs5	100.00	1	97.22	5	97.22	5	97.22	5	97.22	5	97.22	5	94.44	10	100.00	1	100.00	1	100.00	1
Ecoli0267vs35	100.00	1	97.50	5	95.00	8	95.00	8	97.50	5	97.50	5	95.00	8	100.00	1	100.00	1	100.00	1
Glass04vs5	100.00	1	100.00	1	100.00	1	93.75	9	100.00	1	100.00	1	100.00	1	81.25	10	100.00	1	100.00	1
Ecoli0346vs5	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	94.59	10	100.00	1	100.00	1	100.00	1
Yeast05679vs4	84.00	1	71.00	5	63.00	7	73.00	4	79.00	2	74.00	3	68.00	6	14.00	10	33.00	8	32.00	9
Vowel0	96.96	5	99.75	2	96.96	5	99.24	4	100.00	1	99.49	3	94.94	7	64.30	9	61.77	10	81.52	8
Ecoli067vs5	100.00	1	100.00	1	97.50	8	100.00	1	100.00	1	100.00	1	100.00	1	95.00	9	95.00	9	100.00	1
Ecoli0147vs2356	100.00	1	96.72	4	83.61	9	93.44	8	96.72	4	96.72	4	98.36	2	96.72	4	77.05	10	98.36	2
Led7digit02456789vs1	100.00	1	96.30	3	100.00	1	92.59	7	96.30	3	95.06	5	91.36	8	88.89	9	87.65	10	95.06	5
Ecoli01vs5	100.00	1	100.00	1	97.73	9	100.00	1	100.00	1	100.00	1	95.45	10	100.00	1	100.00	1	100.00	1
Glass0146vs2	94.59	4	97.30	1	94.59	4	70.27	8	97.30	1	78.38	7	89.19	6	35.14	10	37.84	9	97.30	1
Glass2	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1
Cleveland0vs4	100.00	1	100.00	1	100.00	1	65.63	10	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1
Ecoli0146vs5	100.00	1	98.08	5	96.15	9	92.31	10	98.08	5	98.08	5	98.08	5	100.00	1	100.00	1	100.00	1
Shuttle0vs4	97.67	2	96.12	5	96.12	5	93.80	9	97.67	2	96.90	4	94.57	8	96.12	5	56.59	10	100.00	1
Yeast1vs7	93.60	1	72.00	6	70.40	7	76.00	4	91.20	2	78.40	3	68.00	8	21.60	10	24.00	9	72.80	5
Glass4	97.60	3	96.80	5	96.00	6	98.40	2	99.20	1	97.60	3	96.00	6	37.60	10	39.20	9	43.20	8
Ecoli4	94.44	1	76.19	5	76.19	5	74.60	7	94.44	1	79.37	3	76.98	4	22.22	10	27.78	9	32.54	8
Pageblocks13vs2	98.88	5	99.44	4	98.88	5	100.00	1	100.00	1	100.00	1	93.30	7	91.62	9	82.68	10	92.18	8
Abalone0918	92.05	4	90.91	9	92.05	4	94.32	2	95.45	1	93.18	3	92.05	7	0.00	10	92.05	7	92.05	4
Dermatology6	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	98.51	10	100.00	1	100.00	1	100.00	1
Glass016vs5	97.89	1	88.42	4	84.21	8	86.32	7	94.74	2	89.47	3	78.95	10	88.42	4	83.16	9	88.42	4
Shuttle2vs4	93.36	1	72.99	5	61.61	7	70.62	6	85.31	2	74.88	3	72.99	4	56.87	8	56.87	9	51.18	10
Yeast1458vs7	100.00	1	84.71	4	78.82	6	64.71	10	96.47	2	84.71	4	91.76	3	65.88	9	78.82	7	75.29	8
Glass5	100.00	1	84.15	7	87.43	5	68.85	10	97.81	2	87.98	4	78.69	8	85.25	6	73.77	9	91.80	3
Yeast2vs8	100.00	1	80.30	7	80.30	7	75.76	10	98.48	2	83.33	5	90.91	4	83.33	5	78.79	9	93.18	3
Yeast4	98.91	1	95.65	4	95.65	4	93.48	7	98.91	1	95.65	4	92.39	8	88.04	9	86.96	10	97.83	3
Winequalityred4	100.00	1	88.35	4	87.06	5	30.42	10	89.00	3	90.94	2	86.73	6	56.63	9	61.17	8	78.96	7
Yeast1289vs7	100.00	1	91.30	8	94.57	7	72.83	9	97.83	5	95.65	6	100.00	1	100.00	1	71.74	10	98.91	4
Winequalityred8vs6	100.00	1	92.13	3	94.49	2	38.58	10	91.34	5	92.13	3	79.53	7	70.87	9	78.74	8	89.76	6
Ecoli0137vs26	99.65	1	93.36	5	88.46	7	82.87	10	98.25	2	94.06	3	89.51	6	84.62	9	86.36	8	94.06	3
Abalone21vs8	100.00	1	98.23	4	99.12	3	91.15	7	98.23	6	98.23	4	100.00	1	50.44	10	53.98	9	87.61	8
Yeast6	99.31	2	98.61	4	97.92	5	94.10	9	99.31	2	97.92	5	94.79	8	95.14	7	90.97	10	99.65	1
Winequalitywhite3vs7	100.00	1	96.02	4	93.75	6	5.11	10	96.59	3	96.59	2	69.32	9	75.00	8	94.89	5	92.61	7
Winequalityred8vs67	100.00	1	93.41	5	94.61	4	38.92	10	93.41	6	96.41	3	78.44	7	73.05	8	67.07	9	97.01	2
Abalone19vs10111213	100.00	1	89.94	3	83.33	7	66.04	10	92.14	2	87.42	5	89.62	4	66.35	8	66.35	8	84.91	6
Winequalitywhite39vs5	99.31	1	92.78	5	90.72	6	11.68	10	93.47	3	97.59	2	89.00	7	81.44	8	80.76	9	93.13	4
Shuttle2vs5	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1
Winequalityred3vs5	100.00	1	96.32	5	97.79	3	50.00	10	96.32	4	96.32	5	88.24	7	63.97	8	58.82	9	100.00	1
Abalone19	100.00	1	94.46	5	88.24	8	94.12	6	100.00	1	91.35	7	84.43	10	95.50	4	86.51	9	95.85	3

Table A.3: G-mean results for the full 66 datasets

Dataset	G-mean Value/Rank																			
	Baseline		SMOTE		BLSMOTE		kmUnder		SMOTE-ENN		SMTBagging		RUSBoost		OBU		AdaOBU		BoostOBU	
Glass1	49.82	9	58.49	8	73.37	2	72.75	4	35.10	10	58.73	7	71.48	5	71.35	6	73.25	3	81.05	1
Ecoli0vs1	0.00	9	69.03	6	68.63	7	74.37	4	0.00	9	85.95	1	83.09	2	77.86	3	72.81	5	55.39	8
Wisconsin	100.00	1	100.00	1	96.36	10	100.00	1	100.00	1	100.00	1	96.36	9	100.00	1	100.00	1	100.00	1
Pima	100.00	1	99.07	6	0.00	9	96.23	8	100.00	1	98.13	7	0.00	9	100.00	1	100.00	1	100.00	1
Iris0	88.56	7	91.25	3	86.77	9	92.88	2	77.54	10	91.25	3	89.37	5	92.88	1	89.37	5	88.41	8
Glass0	89.44	5	94.87	1	74.64	8	94.02	3	89.44	5	94.87	1	93.16	4	71.71	10	74.40	9	82.38	7
Yeast1	64.92	9	81.65	7	62.11	10	86.60	4	83.09	6	81.65	7	87.01	3	87.56	2	86.19	5	93.99	1
Haberman	100.00	1	99.20	6	99.20	6	100.00	1	100.00	1	99.20	6	98.40	10	100.00	1	100.00	1	99.20	6
Vehicle2	76.60	3	74.91	4	74.23	6	76.76	2	66.82	8	74.91	4	78.57	1	73.19	7	62.68	9	62.68	10
Vehicle1	100.00	1	98.43	4	100.00	1	96.82	5	95.20	7	94.87	8	100.00	1	96.82	5	94.87	8	73.10	10
Vehicle3	0.00	7	0.00	7	56.06	2	43.64	6	56.90	1	56.06	2	0.00	7	56.06	2	43.64	5	0.00	7
Glass0123vs456	98.56	6	100.00	1	100.00	1	95.62	8	0.00	9	100.00	1	100.00	1	0.00	9	98.56	7	100.00	1
Vehicle0	76.98	2	69.92	8	75.03	3	69.92	7	68.31	9	78.88	1	71.66	6	50.92	10	72.01	4	72.01	5
Ecoli1	0.00	10	80.60	1	80.60	1	59.91	6	57.74	7	56.99	8	70.41	3	53.91	9	62.02	5	64.05	4
Newthyroid1	100.00	1	100.00	1	100.00	1	97.47	5	70.71	10	97.47	5	90.83	8	77.46	9	97.47	5	100.00	1
Newthyroid2	0.00	2	0.00	2	0.00	2	73.25	1	0.00	2	0.00	2	0.00	2	0.00	2	0.00	2	0.00	2
Ecoli2	89.44	4	100.00	1	89.44	4	77.46	9	77.46	9	89.44	4	97.26	3	89.44	4	100.00	1	89.44	4
Segment0	43.30	10	53.75	5	49.30	8	61.24	3	47.14	9	59.16	4	52.70	6	63.25	1	63.03	2	52.04	7
Glass6	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1
Yeast3	98.60	6	100.00	1	37.80	10	100.00	1	75.59	9	100.00	1	100.00	1	94.28	8	97.18	7	100.00	1
Ecoli3	75.59	7	75.59	7	75.59	7	98.60	3	100.00	1	75.59	7	100.00	1	94.28	4	89.75	5	83.33	6
Pageblocks0	89.71	8	93.49	4	93.55	2	93.54	3	95.11	1	93.24	5	92.22	7	92.57	6	47.97	10	60.46	9
Yeast2vs4	100.00	1	100.00	1	100.00	1	97.12	8	100.00	1	100.00	1	93.54	9	98.86	7	88.55	10	100.00	1
Ecoli067vs35	70.71	4	70.71	4	69.82	9	70.71	4	70.71	4	70.71	4	81.01	1	77.46	2	76.24	3	69.82	9
Glass015vs2	0.00	9	56.80	3	51.85	5	60.46	1	0.00	9	57.74	2	54.87	4	32.79	8	43.99	7	46.37	6
Yeast02579vs368	85.60	7	86.87	4	78.66	8	86.62	5	89.72	2	89.97	1	88.15	3	73.78	10	86.11	6	75.03	9
Ecoli046vs5	100.00	1	98.60	3	85.39	9	98.60	3	98.60	3	85.39	9	97.18	6	100.00	1	86.60	7	86.60	7
Ecoli0267vs35	70.71	1	69.82	5	68.92	8	68.92	8	69.82	5	69.82	5	68.92	8	70.71	1	70.71	1	70.71	1
Glass04vs5	100.00	1	100.00	1	0.00	10	96.82	8	100.00	1	100.00	1	100.00	1	90.14	9	100.00	1	100.00	1
Ecoli0346vs5	86.60	1	86.60	1	50.00	10	86.60	1	86.60	1	86.60	1	84.23	9	86.60	1	86.60	1	86.60	1
Yeast05679vs4	70.09	4	72.28	2	71.49	3	73.29	1	61.04	7	66.84	6	69.82	5	35.24	10	50.53	9	52.12	8
Vowel0	93.02	6	98.32	3	93.02	6	99.62	1	99.23	2	97.42	4	95.16	5	80.19	9	78.60	10	90.29	8
Ecoli067vs5	86.60	3	86.60	3	85.51	10	86.60	3	86.60	3	86.60	3	86.60	3	97.47	1	97.47	1	86.60	3
Ecoli0147vs2356	77.46	2	76.18	5	70.83	10	74.88	9	76.18	5	76.18	5	76.82	3	76.18	5	78.51	1	76.82	3
Led7digit02456789vs1	84.52	7	90.85	3	92.58	2	81.33	10	82.94	8	90.27	4	95.58	1	87.29	5	86.68	6	82.40	9
Ecoli01vs5	70.71	1	70.71	1	49.43	10	70.71	1	70.71	1	70.71	1	69.08	9	70.71	1	70.71	1	70.71	1
Glass0146vs2	56.15	10	80.54	4	97.26	1	83.83	3	56.95	8	88.53	2	77.11	5	59.27	7	61.51	6	56.95	8
Glass2	0.00	5	0.00	5	0.00	5	100.00	1	100.00	1	0.00	5	100.00	1	100.00	1	0.00	5	0.00	5
Cleveland0vs4	70.71	2	0.00	6	0.00	6	81.01	1	0.00	6	0.00	6	70.71	2	70.71	2	70.71	2	0.00	6
Ecoli0146vs5	100.00	1	99.03	5	84.92	10	96.08	9	99.03	5	99.03	5	99.03	5	100.00	1	100.00	1	100.00	1
Shuttle0vs4	94.95	7	95.50	6	94.20	8	95.60	5	83.74	9	95.88	4	97.25	3	98.04	2	75.23	10	100.00	1
Yeast1vs7	62.60	7	79.77	5	75.70	6	85.13	1	48.30	8	83.24	2	80.52	3	45.38	10	47.25	9	80.21	4
Glass4	96.47	2	96.07	4	88.40	6	96.86	1	80.37	7	96.47	2	95.67	5	61.32	10	62.61	9	65.73	8
Ecoli4	59.98	6	73.77	5	76.19	3	79.97	1	44.99	9	74.03	4	77.77	2	43.64	10	50.13	8	52.07	7
Pageblocks13vs2	99.44	4	99.72	3	99.44	4	100.00	1	84.98	10	100.00	1	96.59	6	95.72	8	90.93	9	96.01	7
Abalone0918	94.91	4	94.33	9	94.91	4	97.12	1	96.66	2	95.50	3	94.91	7	0.00	10	94.91	7	94.91	4
Dermatology6	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	70.18	10	100.00	1	100.00	1	100.00	1
Glass016vs5	76.64	8	89.21	2	64.89	10	77.73	7	68.82	9	89.74	1	84.29	3	84.11	4	81.56	6	84.11	4
Shuttle2vs4	62.00	10	73.55	2	69.69	6	71.19	4	66.45	9	73.31	3	74.13	1	68.44	7	69.89	5	66.75	8
Yeast1458vs7	81.65	4	75.15	6	51.26	9	80.44	5	0.00	10	84.02	3	87.45	2	74.10	7	88.78	1	70.85	8
Glass5	0.00	9	52.96	8	85.36	2	67.75	6	0.00	9	76.58	5	88.71	1	65.29	7	78.41	3	78.23	4
Yeast2vs8	0.00	9	51.74	4	51.74	4	50.25	6	0.00	9	52.70	2	38.92	7	52.70	2	36.24	8	55.73	1
Yeast4	83.21	9	92.78	3	97.80	1	91.72	4	62.90	10	87.48	7	91.19	5	83.93	8	88.47	6	93.83	2
Winequalityred4	0.00	10	59.45	2	59.01	3	52.32	7	59.66	1	52.23	8	41.65	9	53.21	6	55.30	5	56.20	4
Yeast1289vs7	86.60	2	82.75	5	0.00	8	60.34	6	0.00	8	84.70	3	100.00	1	0.00	8	84.70	4	49.73	7
Winequalityred8vs6	0.00	10	55.42	8	97.21	1	62.11	7	78.03	3	55.42	8	89.18	2	68.73	6	72.45	5	77.36	4
Ecoli0137vs26	54.68	9	74.84	5	84.12	1	81.42	3	54.29	10	75.12	4	73.28	6	82.28	2	71.98	7	68.58	8
Abalone21vs8	0.00	8	70.08	6	0.00	8	95.47	1	70.08	7	0.00	8	70.71	5	71.02	4	73.47	3	93.60	2
Yeast6	78.78	9	99.30	2	92.56	6	84.01	8	70.46	10	98.95	3	97.36	4	84.47	7	95.38	5	99.83	1
Winequalitywhite3vs7	50.00	4	0.00	7	0.00	7	22.61	6	0.00	7	0.00	7	83.26	2	75.00	3	84.36	1	48.12	5
Winequalityred8vs67	0.00	6	0.00	6	0.00	6	62.39	4	0.00	6	0.00	6	51.13	5	85.47	1	81.89	2	80.42	3
Abalone19vs10111213	0.00	10	38.72	5	37.27	9	46.92	3	39.19	4	38.17	7	38.65	6	57.60	1	57.60	1	37.62	8
Winequalitywhite39vs5	0.00	7	0.00	7	42.60	5	34.18	6	0.00	7	0.00	7	73.08	1	57.08	3	56.84	4	61.03	2
Shuttle2vs5	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1
Winequalityred3vs5	0.00	10	69.40	5	69.93	3	70.71	1	69.40	4	69.40	5	66.42	7	56.56	8	54.23	9	70.71	1
Abalone19	84.52	8	89.98	5	86.97	7	81.99	9	65.47	10	88.49	6	91.89	2	90.48	4	93.01	1	90.64	3

Table A.4: F1-score results for the full 66 datasets

Dataset	F1-score Value/Rank																			
	Baseline		SMOTE		BLSMOTE		kmUnder		SMOTE-ENN		SMTBagging		RUSBoost		OBU		AdaOBU		BoostOBU	
Glass1	36.36	2	26.09	7	31.25	3	29.41	4	18.18	10	27.27	5	26.32	6	20.69	8	20.59	9	38.71	1
Ecoli0vs1	0.00	9	12.50	1	10.53	2	5.30	6	0.00	9	9.52	3	6.54	4	3.55	7	2.99	8	5.41	5
Wisconsin	100.00	1	100.00	1	93.75	9	100.00	1	100.00	1	100.00	1	93.75	9	100.00	1	100.00	1	100.00	1
Pima	100.00	1	66.67	6	0.00	9	33.33	8	100.00	1	50.00	7	0.00	9	100.00	1	100.00	1	100.00	1
Iris0	85.71	3	86.67	1	80.00	8	81.08	6	66.67	10	86.67	1	81.25	4	81.08	7	81.25	4	78.79	9
Glass0	88.89	4	94.74	1	60.00	7	90.00	3	88.89	4	94.74	1	85.71	6	41.86	10	44.44	9	52.63	8
Yeast1	54.55	7	62.50	3	37.50	10	48.28	9	71.43	1	62.50	3	60.00	5	50.00	8	57.14	6	66.67	2
Haberman	100.00	1	88.89	6	88.89	6	100.00	1	100.00	1	88.89	6	80.00	10	100.00	1	100.00	1	88.89	6
Vehicle2	68.97	2	66.67	5	66.67	5	68.75	3	58.33	10	66.67	5	70.97	1	68.29	4	62.22	9	62.22	8
Vehicle1	100.00	1	95.24	4	100.00	1	90.91	8	86.96	9	94.74	5	100.00	1	90.91	7	94.74	6	56.25	10
Vehicle3	0.00	7	0.00	7	33.33	2	10.53	6	40.00	1	33.33	2	0.00	7	20.00	4	13.33	5	0.00	7
Glass0123vs456	66.67	7	100.00	1	100.00	1	40.00	8	0.00	9	100.00	1	100.00	1	0.00	9	66.67	6	100.00	1
Vehicle0	71.43	2	62.07	8	68.57	5	62.86	7	60.00	9	72.73	1	66.67	6	60.00	9	69.77	3	69.77	4
Ecoli1	0.00	10	66.67	1	66.67	1	17.39	8	50.00	3	40.00	4	26.67	5	14.81	9	20.00	6	20.00	6
Newthyroid1	100.00	1	100.00	1	100.00	1	66.67	5	66.67	7	66.67	7	36.36	9	20.00	10	66.67	5	100.00	1
Newthyroid2	0.00	2	0.00	2	0.00	2	9.52	1	0.00	2	0.00	2	0.00	2	0.00	2	0.00	2	0.00	2
Ecoli2	88.89	3	100.00	1	88.89	3	75.00	9	75.00	9	88.89	3	83.33	8	88.89	3	100.00	1	88.89	3
Segment0	31.58	9	37.21	6	32.56	8	45.00	3	29.41	10	42.86	5	36.36	7	48.98	1	47.83	2	44.07	4
Glass6	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1
Yeast3	93.33	6	100.00	1	25.00	10	100.00	1	72.73	9	100.00	1	100.00	1	77.78	8	87.50	7	100.00	1
Ecoli3	72.73	5	72.73	5	72.73	5	93.33	3	100.00	1	72.73	5	100.00	1	77.78	4	66.67	9	56.00	10
Pageblocks0	82.73	3	76.01	6	70.90	8	78.46	4	83.53	2	76.69	5	75.19	7	85.46	1	22.70	10	26.27	9
Yeast2vs4	100.00	1	100.00	1	100.00	1	66.67	8	100.00	1	100.00	1	47.62	9	83.33	7	34.48	10	100.00	1
Ecoli067vs35	66.67	1	66.67	1	57.14	6	66.67	1	66.67	1	66.67	1	50.00	8	40.00	9	37.50	10	57.14	6
Glass015vs2	0.00	9	40.00	2	20.00	5	21.05	4	0.00	9	50.00	1	28.57	3	12.90	8	14.81	6	13.33	7
Yeast02579vs368	82.35	1	71.43	4	45.16	8	69.77	5	74.42	3	76.19	2	65.31	7	32.32	10	66.67	6	33.68	9
Ecoli046vs5	100.00	1	88.89	3	75.00	9	88.89	3	88.89	3	75.00	9	80.00	8	100.00	1	85.71	6	85.71	6
Ecoli0267vs35	66.67	1	57.14	5	50.00	8	50.00	8	57.14	5	57.14	5	50.00	8	66.67	1	66.67	1	66.67	1
Glass04vs5	100.00	1	100.00	1	0.00	10	66.67	8	100.00	1	100.00	1	100.00	1	40.00	9	100.00	1	100.00	1
Ecoli0346vs5	85.71	1	85.71	1	40.00	10	85.71	1	85.71	1	85.71	1	66.67	9	85.71	1	85.71	1	85.71	1
Yeast05679vs4	62.00	4	64.46	3	64.66	2	65.55	1	50.51	10	57.66	6	61.79	5	50.54	9	50.93	8	54.22	7
Vowel0	85.93	5	97.67	3	85.93	5	97.74	2	99.22	1	96.12	4	84.35	7	47.97	9	46.26	10	64.04	8
Ecoli067vs5	85.71	1	85.71	1	75.00	10	85.71	1	85.71	1	85.71	1	85.71	1	80.00	8	80.00	8	85.71	1
Ecoli0147vs2356	75.00	1	60.00	4	33.33	10	50.00	8	60.00	4	60.00	4	66.67	2	60.00	4	34.78	9	66.67	2
Led7digit02456789vs1	83.33	2	75.00	3	92.31	1	55.56	8	66.67	5	70.59	4	66.67	5	54.55	9	52.17	10	62.50	7
Ecoli01vs5	66.67	1	66.67	1	33.33	10	66.67	1	66.67	1	66.67	1	50.00	9	66.67	1	66.67	1	66.67	1
Glass0146vs2	33.33	8	66.67	2	75.00	1	35.29	7	40.00	5	42.86	4	44.44	3	20.00	10	20.69	9	40.00	5
Glass2	0.00	5	0.00	5	0.00	5	100.00	1	100.00	1	0.00	5	100.00	1	100.00	1	0.00	5	0.00	5
Cleveland0vs4	66.67	1	0.00	6	0.00	6	26.67	5	0.00	6	0.00	6	66.67	1	66.67	1	66.67	1	0.00	6
Ecoli0146vs5	100.00	1	88.89	5	66.67	9	66.67	9	88.89	5	88.89	5	88.89	5	100.00	1	100.00	1	100.00	1
Shuttle0vs4	92.31	4	91.36	6	90.00	7	89.41	8	80.00	9	92.50	3	91.76	5	93.98	2	58.21	10	100.00	1
Yeast1vs7	52.17	7	65.52	5	60.87	6	71.93	1	33.85	10	70.37	2	66.13	3	45.05	8	44.94	9	66.09	4
Glass4	94.25	2	93.18	4	84.34	6	95.35	1	77.78	7	94.25	2	92.13	5	52.44	10	53.09	9	54.78	8
Ecoli4	49.23	6	58.82	5	61.54	3	65.45	1	31.03	10	59.79	4	63.46	2	40.91	9	44.44	7	43.21	8
Pageblocks13vs2	94.74	4	97.30	3	94.74	4	100.00	1	83.87	6	100.00	1	75.00	7	70.59	9	53.73	10	72.00	8
Abalone0918	92.00	3	91.09	8	92.00	3	66.67	9	94.85	1	92.93	2	92.00	3	51.65	10	92.00	3	92.00	3
Dermatology6	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	57.14	10	100.00	1	100.00	1	100.00	1
Glass016vs5	66.67	1	60.00	3	33.33	10	46.67	8	50.00	6	62.07	2	46.15	9	55.17	4	47.06	7	55.17	4
Shuttle2vs4	52.24	10	61.46	2	57.51	6	58.65	4	55.00	9	61.31	3	62.14	1	56.91	7	58.63	5	56.49	8
Yeast1458vs7	80.00	1	34.78	5	15.38	9	28.57	6	0.00	10	41.67	3	55.56	2	25.00	8	40.00	4	25.81	7
Glass5	0.00	9	10.81	8	29.41	2	11.94	7	0.00	9	25.00	3	23.53	4	16.67	6	16.95	5	32.00	1
Yeast2vs8	0.00	9	11.76	4	11.76	4	10.00	7	0.00	9	13.33	2	10.53	6	13.33	2	5.71	8	23.53	1
Yeast4	77.78	4	78.26	3	83.33	2	72.00	6	53.33	10	72.73	5	69.23	7	55.17	9	58.06	8	85.71	1
Winequalityred4	0.00	10	16.00	2	14.81	3	7.69	6	16.67	1	14.63	4	7.55	7	6.71	9	7.41	8	10.13	5
Yeast1289vs7	85.71	2	40.00	4	0.00	8	12.90	7	0.00	8	54.55	3	100.00	1	0.00	8	23.53	6	33.33	5
Winequalityred8vs6	0.00	10	14.29	5	46.15	1	7.14	9	25.00	2	14.29	5	18.75	4	9.52	8	12.50	7	22.22	3
Ecoli0137vs26	42.86	1	34.29	3	31.37	5	23.88	9	33.33	4	36.36	2	26.09	7	25.81	8	21.82	10	31.25	6
Abalone21vs8	0.00	8	40.00	2	0.00	8	28.57	4	40.00	2	0.00	8	66.67	1	6.67	7	7.14	6	22.22	5
Yeast6	66.67	4	80.00	2	66.67	4	38.71	9	57.14	6	72.73	3	51.61	7	42.86	8	38.10	10	94.12	1
Winequalitywhite3vs7	40.00	1	0.00	7	0.00	7	4.57	6	0.00	7	0.00	7	12.90	3	11.76	4	37.50	2	11.11	5
Winequalityred8vs67	0.00	6	0.00	6	0.00	6	5.56	4	0.00	6	0.00	6	5.00	5	11.76	2	9.84	3	40.00	1
Abalone19vs10111213	0.00	10	5.13	4	3.33	9	3.45	8	6.25	1	4.26	6	5.00	5	5.17	2	5.17	2	3.64	7
Winequalitywhite39vs5	0.00	7	0.00	7	6.06	5	3.75	6	0.00	7	0.00	7	15.00	1	6.56	3	6.35	4	14.81	2
Shuttle2vs5	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1	100.00	1
Winequalityred3vs5	0.00	10	25.00	3	33.33	2	5.56	7	25.00	3	25.00	3	10.53	6	3.85	8	3.39	9	66.67	1
Abalone19	83.33	1	41.38	5	25.53	9	34.48	6	60.00	2	31.58	7	23.73	10	46.15	4	26.42	8	48.00	3